

欢迎继续我们的学习。

在这一课的内容中，我们将学习Pandas中的其它常用操作。

首先打开Anaconda和Jupyter Notebook创建一个新的项目，命名为Operations

先让我们创建一个Dataframe:

```
In [1]: import numpy as np
import pandas as pd

In [2]: df = pd.DataFrame({'col1':[1,2,3,4],
                           'col2':[444,555,666,444],
                           'col3':['abc','def','ghi','xyz']})
df.head()

Out[2]:
```

	col1	col2	col3
0	1	444	abc
1	2	555	def
2	3	666	ghi
3	4	444	xyz

接下来让我们看看如何在dataframe中找到特定的元素。

```
In [3]: df['col2'].unique()

Out[3]: array([444, 555, 666])
```

在以上代码中，我们使用unique函数找到某列中所有不重复的数据。

```
In [13]: df['col2'].nunique()

Out[13]: 3
```

在以上代码中，我们使用nunique函数找到某列中不重复数据的数量。

```
In [14]: df['col2'].value_counts()

Out[14]: 444    2
         555    1
         666    1
         Name: col2, dtype: int64
```

在以上代码中，我们使用value\_counts获取某列中每个元素出现的次数。

```
In [15]: df[df['col1']>2]
```

```
Out[15]:
```

	col1	col2	col3
2	3	666	ghi
3	4	444	xyz

在以上代码中，我们以某列的元素大于2为选择条件的子矩阵。

```
In [16]: def times2(x):  
         return x*2
```

```
In [18]: df['col1'].apply(times2)
```

```
Out[18]: 0    2  
         1    4  
         2    6  
         3    8  
         Name: col1, dtype: int64
```

在以上代码中，我们定义了一个自己的函数，并将其应用于矩阵的某列。

```
In [19]: df['col2'].apply(lambda x: x*2)
```

```
Out[19]: 0    888  
         1   1110  
         2   1332  
         3    888  
         Name: col2, dtype: int64
```

在以上代码中，我们使用lambda表达式将某列元素的数值乘以2。

```
In [20]: df.drop('col1',axis = 1)
```

```
Out[20]:
```

	col2	col3
0	444	abc
1	555	def
2	666	ghi
3	444	xyz

在以上代码中，使用drop函数删除了矩阵的特定列。

```
In [21]: df.columns
```

```
Out[21]: Index(['col1', 'col2', 'col3'], dtype='object')
```

```
In [22]: df.index
```

```
Out[22]: RangeIndex(start=0, stop=4, step=1)
```

在以上代码中，我们分别使用columns和index查看了矩阵的相关属性。  
接下来我们看看如何给矩阵排序。

```
In [23]: df.sort_values('col2')
```

```
Out[23]:
```

	col1	col2	col3
0	1	444	abc
3	4	444	xyz
1	2	555	def
2	3	666	ghi

在以上代码中，我们按照col2中的元素大小给矩阵排序。

```
In [24]: df.isnull()
```

```
Out[24]:
```

	col1	col2	col3
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False

在以上代码中，我们使用isnull函数判断矩阵中的元素是否为空。

最后再来看一下pivot\_table的使用：

```
In [25]: data = {'A': ['foo', 'foo', 'foo', 'bar', 'bar', 'bar'],  
                'B': ['one', 'one', 'two', 'two', 'one', 'one'],  
                'C': ['x', 'y', 'x', 'y', 'x', 'y'],  
                'D': [1, 3, 2, 5, 4, 1]}  
df = pd.DataFrame(data)
```

```
In [26]: df
```

```
Out[26]:
```

	A	B	C	D
0	foo	one	x	1
1	foo	one	y	3
2	foo	two	x	2
3	bar	two	y	5
4	bar	one	x	4
5	bar	one	y	1

```
In [27]: df.pivot_table(values= 'D', index = ['A', 'B'], columns= ['C'])
```

```
Out[27]:
```

		C	x	y
A	B			
bar	one	4.0	1.0	
	two	NaN	5.0	
foo	one	1.0	3.0	
	two	2.0	NaN	

好了，本课的内容就到这里了，我们下一课再见~

讨论群-笨猫学编程QQ群：

375143733

知乎专栏:

<https://zhuanlan.zhihu.com/kidscoding>

新浪博客:

<http://blog.sina.com.cn/eseedo>

Github:

<https://github.com/eseedo>

个人网站:

<http://icode.ai/>