23、利用pandas处理数据（1）

1、利用编程进行数据分析与挖掘的优势

使用计算机语言编程，可以更加灵活、深入地进行数据分析和挖掘。选用Python语言编程进行数据处理，可以调Python的扩展模块，常用的扩展模块有numpy、scipy、pandas和matplotlib等。

numpy是科学计算的基础模块，提供了数组、矩阵运算等基础函数；

scipy主要用于高等数学、信号处理、图像处理、统计等方面；

pandas模块主要用于数据处理和分析，能方便的操作大型数据集；

matplotlib模块主要用于数据图表可视化操作；

2、Series数据结构

Series是pandas定义的一种一维数据结构，包括一个数组的数据和与数据相关联的索引。除此之外，它还提供了强大的数据处理功能。使用Series数据结构，许多问题会变的很简单。

1）创建Series有很多种办法，主要有：

a）用列表创建，不指定索引

```
import pandas as pd
s1=pd.Series([2,6,9,4,3,7,8])
print(s1)
```

```
0    2
1    6
2    9
3    4
4    3
5    7
6    8
dtype: int64
```

b)用列表创建，并指定索引

```
import pandas as pd
s1=pd.Series([2,6,9,4,3,7,8],index=['a','b','c','d','e','f','g'])
print(s1)
```

```
a    2
b    6
c    9
d    4
e    3
f    7
g    8
dtype: int64
```

c)用字典创建，自动加上索引

```
import pandas as pd
dic={'a': 2, 'b': 6, 'c': 9, 'd': 4, 'e': 3, 'f': 7, 'g': 8}
s1=pd.Series(dic)
print(s1)
```

```
a    2
b    6
c    9
d    4
e    3
f    7
g    8
dtype: int64
```

2）Series两大属性：

a）values属性：

```
s1.values
```

```
array([2, 6, 9, 4, 3, 7, 8])
```

b）index属性：

```
s1.index
```

```
Index(['A', 'B', 'C', 'D', 'E', 'F', 'G'], dtype='object')
```

3、利用Series统计评分

某次比赛中共有7位评委为选手打分，评委及打分信息存储在文本文件scores.txt中。

1）打开文件，并读取数据到字符串：

```
with open("scores.txt") as f:
    s=f.read()
s
```

```
'评委A:8.2\n评委B:7.9\n评委C:7.7\n评委D:9.1\n评委E:8.5\n评委F:8.3\n评委G:8.4'
```

可以看到字符串中，各评委评分之间用\n分隔，评委与分数之间用:分隔

2）将数据保存到Series对象中：

```
import pandas as pd
slist=s.split("\n")
pw=[i.split(":")[0] for i in sl]
fs=[float(i.split(":")[1]) for i in sl]
s1=pd.Series(fs,index=pw)
print(s1)
```

```
评委A    8.2
评委B    7.9
评委C    7.7
评委D    9.1
评委E    8.5
评委F    8.3
评委G    8.4
dtype: float64
```

sl-->slist

还有哪些方法可以保存数据到s1?

3）计算最高分和最低分，同时输出是哪位评委的给分：

```
print("最高分为:",s1.max(),"给分者:",s1.idxmax())
print("最低分为:",s1.min(),"给分者:",s1.idxmin())
```

```
最高分为: 9.1 给分者: 评委D
最低分为: 7.7 给分者: 评委C
```

4）按照评分从高到低排序：

```
s1=s1.sort_values(ascending=False)
s1
```

```
评委D    9.1
评委E    8.5
评委G    8.4
评委F    8.3
评委A    8.2
评委B    7.9
评委C    7.7
dtype: float64
```

5）得到去掉最高分和最低分后的平均得分：

```
print("去掉最高分和最低分后的平均得分为: ",s1[1:-1].mean())
```

```
去掉最高分和最低分后的平均得分为:  8.26
```

注意，Series可以用类似列表的切片方法

还有哪些方法可以实现该功能?

6）输出评分超过8分的评委名单：

```
print("超过8分的评委有: ","、".join(s1[s1>8].index))
```

```
超过8分的评委有:  评委D、评委E、评委G、评委F、评委A
```

可见，学会熟练利用Series对象来处理数据会非常方便。