

# pandas

---

主讲老师：杨 洋

# 1 pandas快速入门2



## pandas快速入门2

1.1使用分组聚合进行组内计算

1.2创建透视表与交叉表

## 1.1.1 使用groupby方法拆分数据

### groupby方法的参数及其说明

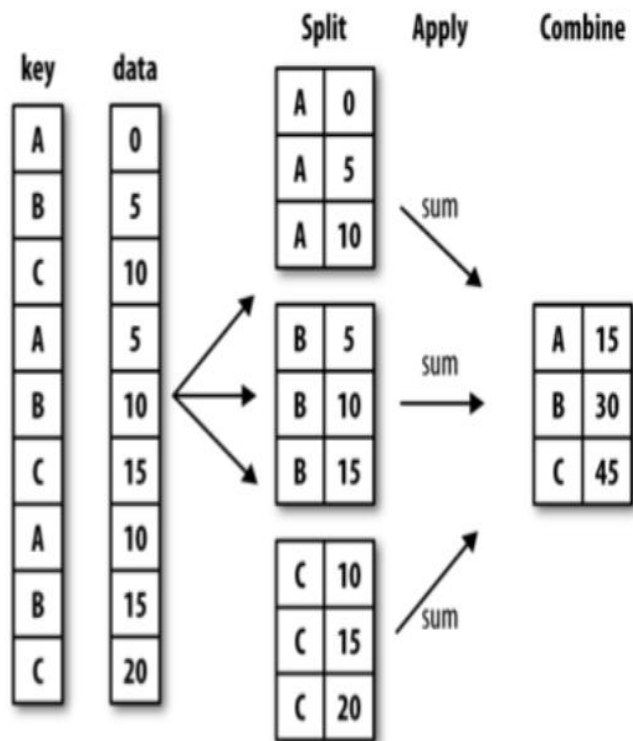
- 该方法提供的是分组聚合步骤中的拆分功能，能根据索引或字段对数据进行分组。其常用参数与使用格式如下。

*DataFrame.groupby(by=None, axis=0, level=None, as\_index=True, sort=True, group\_keys=True, squeeze=False, \*\*kwargs)*

参数名称	说明
by	接收list, string, mapping或generator。用于确定进行分组的依据。无默认。
axis	接收int。表示操作的轴向，默认对列进行操作。默认为0。
level	接收int或者索引名。代表标签所在级别。默认为None。
as_index	接收boolearn。表示聚合后的聚合标签是否以DataFrame索引形式输出。默认为True。
sort	接收boolearn。表示是否对分组依据分组标签进行排序。默认为True。
group_keys	接收boolearn。表示是否显示分组标签的名称。默认为True。
squeeze	接收boolearn。表示是否在允许的情况下对返回数据进行降维。默认为False。

## 1.1.1 使用groupby方法拆分数据

groupby方法的参数及其说明——by参数的特别说明



- 如果传入的是一个数组则对其进行计算并分组。
- 如果传入的是一个字典或者Series 则字典或者Series的值用来做分组依据。
- 如果传入一个NumPy数组则数据的元素作为分组依据。
- 如果传入的是列名，字符串或者字符串列表，则使用这些字符串所代表的字段作为分组依据。
- 打开group.py练习

### 1.1.1 使用groupby方法拆分数据

#### GroupBy对象常用的描述性统计方法

- 用groupby方法分组后的结果并不能直接查看，而是被存在内存中，输出的是内存地址。实际上分组后的数据对象GroupBy类似Series与DataFrame，是pandas提供的一种对象。GroupBy对象常用的描述性统计方法如下。

方法名称	说明	方法名称	说明
count	计算分组的数目，包括缺失值。	cumcount	对每个分组中组员的进行标记，0至n-1。
head	返回每组的前n个值。	size	返回每组的大小。
max	返回每组最大值。	min	返回每组最小值。
mean	返回每组的均值。	std	返回每组的标准差。
median	返回每组的中位数。	sum	返回每组的和。

# 1.1 使用分组聚合进行组内计算

## 1.1.1 使用groupby方法拆分数据

1. 在group.py文件中给定以下数组,

```
states = np.array(['Ohio', 'California', 'California', 'Ohio', 'Ohio'])
```

```
years = np.array([2005, 2005, 2006, 2005, 2006])
```

按照states, years传入数组的方式对data1列数据分组后进行平均值计算

2. 创建以下数据, 并按照 mapping = {'a':'red', 'b':'red', 'c':'blue', 'd':'blue', 'e':'red'} 进行分组求和计算

提示: 计算时需要指定轴

	key1	key2	data1	data2
0	a	one	1.071446	-1.340262
1	a	two	1.125852	1.727092
2	b	one	0.342136	0.202296
3	b	two	-0.488568	-0.188095
4	a	one	1.031421	0.019024

	a	b	c	d	e
Joe	1.550826	0.103725	-0.475930	-0.519439	0.163914
Steve	0.595992	-0.845790	1.111299	-1.279929	-1.540709
Wes	0.202888	1.031555	-1.659072	1.468362	0.828954
Jim	0.440928	0.785941	-0.536841	0.167377	0.936588
Travis	-1.668387	0.289039	0.824654	-0.468110	-1.294456

## 1.1.2 使用agg方法聚合数据

agg和aggregate函数参数及其说明

- agg, aggregate方法都支持对每个分组应用某函数，包括Python内置函数或自定义函数。同时这两个方法能够也能够直接对DataFrame进行函数应用操作。
- 在正常使用过程中，agg函数和aggregate函数对DataFrame对象操作时功能几乎完全相同，因此只需要掌握其中一个函数即可。它们的参数说明如下表。

*DataFrame.agg(func, axis=0, \*args, \*\*kwargs)*

*DataFrame.aggregate(func, axis=0, \*args, \*\*kwargs)*

参数名称	说明
func	接收list、dict、function。表示应用于每行 / 每列的函数。无默认。
axis	接收0或1。代表操作的轴向。默认为0。



## 1.1.2 使用agg方法聚合数据

agg方法求统计量

- 可以使用agg方法一次求出当前数据中所有菜品销量和售价的总和与均值，如 `detail[['counts','amounts']].agg([np.sum,np.mean])`。
- 对于某个字段希望只做求均值操作，而对另一个字段则希望只做求和操作，可以使用字典的方式，将两个字段名分别作为key，然后将NumPy库的求和与求均值的函数分别作为value，如 `detail.agg({'counts':np.sum,'amounts':np.mean})`。
- 在某些时候还希望求出某个字段的多个统计量，某些字段则只需要一个统计量，此时只需要将字典对应key的value变为列表，列表元素为多个目标的统计量即可，如 `detail.agg({'counts':np.sum,'amounts':[np.mean,np.sum]})`

## 1.1 使用分组聚合进行组内计算

### 案例

在age.py文件中进行操作，按照A列分组后聚合后对B列求最小值和最大值，C列求和

### 1.1.3 使用apply方法聚合数据

- apply方法类似agg方法能够将函数应用于每一列。不同之处在于apply方法相比agg方法传入的函数只能够作用于整个DataFrame或者Series，而无法像agg一样能够对不同字段，应用不同函数获取不同结果。
- 使用apply方法对GroupBy对象进行聚合操作其方法和agg方法也相同，只是使用agg方法能够实现对不同的字段进行应用不同的函数，而apply则不行。

*DataFrame.apply(func, axis=0, broadcast=False, raw=False, reduce=None, args=(), \*\*kws)*

参数名称	说明
func	接收functions。表示应用于每行 / 列的函数。无默认。
axis	接收0或1。代表操作的轴向。默认为0。
broadcast	接收boolearn。表示是否进行广播。默认为False。
raw	接收boolearn。表示是否直接将ndarray对象传递给函数。默认为False。
reduce	接收boolearn或者None。表示返回值的格式。默认None。

# 1.1 使用分组聚合进行组内计算

## 练习

在apply.py文件中进行操作

计算各列数据总和并作为新列添加到末尾,

计算各行数据总和并作为新行添加到末尾

	A	B	C	D	E	Col_sum
0	0.610764	-1.096924	0.856870	0.017334	-0.945760	-0.557716
1	0.689502	2.188957	-1.565911	-1.339480	1.120758	1.093826
2	0.425448	0.589814	0.531356	-1.579106	1.856440	1.823953
3	-1.470899	0.740575	-1.276242	1.471284	-0.658182	-1.193465
Row_sum	0.254815	2.422421	-1.453927	-1.429967	1.373256	1.166598

### 1.2.1 使用pivot\_table函数创建透视表

pivot\_table函数常用参数及其说明

- 利用pivot\_table函数可以实现透视表，pivot\_table()函数的常用参数及其使用格式如下。

*pandas.pivot\_table(data, values=None, index=None, columns=None, aggfunc='mean', fill\_value=None, margins=False, dropna=True, margins\_name='All')*

参数名称	说明
data	接收DataFrame。表示创建表的数据。无默认。
values	接收字符串。用于指定想要聚合的数据字段名，默认使用全部数据。默认为None。
index	接收string或list。表示行分组键。默认为None。
columns	接收string或list。表示列分组键。默认为None。
aggfunc	接收functions。表示聚合函数。默认为mean。
margins	接收boolean。表示汇总（Total）功能的开关，设为True后结果集中会出现名为“ALL”的行和列。默认为True。
dropna	接收boolean。表示是否删掉全为NaN的列。默认为False。

## 1.2.1 使用pivot\_table函数创建透视表

pivot\_table函数主要的参数调节

- 在不特殊指定聚合函数aggfunc时，会默认使用numpy.mean进行聚合运算，numpy.mean会自动过滤掉非数值类型数据。可以通过指定aggfunc参数修改聚合函数。
- 和groupby方法分组的时候相同，pivot\_table函数在创建透视表的时候分组键index可以有多个。
- 通过设置columns参数可以指定列分组。
- 当全部数据列数很多时，若只想要显示某列，可以通过指定values参数来实现。
- 当某些数据不存在时，会自动填充NaN，因此可以指定fill\_value参数，表示当存在缺失值时，以指定数值进行填充。
- 可以更改margins参数，查看汇总数据。

## ●案例：查看每一部电影不同性别的平均评分

	Gender	F	M
Title			
\$1,000,000 Duck (1971)		3.375000	2.761905
'Night Mother (1986)		3.388889	3.352941
'Til There Was You (1997)		2.675676	2.733333
'burbs, The (1989)		2.793478	2.962085
...And Justice for All (1979)		3.828571	3.689024
1-900 (1994)		2.000000	3.000000
10 Things I Hate About You (1999)		3.646552	3.311966
101 Dalmatians (1961)		3.791444	3.500000
101 Dalmatians (1996)		3.240000	2.911215
12 Angry Men (1957)		4.184397	4.328421
13th Warrior, The (1999)		3.112000	3.168000

## ●问题

对分布在三个表的数据进行分析同时进行分析很难，那必须将所有数据都合并到一个表中进行分析。采用什么方法呢？

下面，用pandas的merge函数将ratings跟users合并到一起，然后再将movies也合并进去。

```
data = pd.merge(pd.merge(ratings,users),movies)
```

pandas会根据列名的重叠情况推断出哪些列是合并（或连接）键



## ●案例： 查看每一部电影不同性别的平均评分

使用pivot\_table函数创建透视表-pivot\_table函数常用参数及其说明

- 利用pivot\_table函数可以实现透视表， pivot\_table()函数的常用参数及其使用格式如下。

```
data_gender = data.pivot_table(values='Rating', index='Title', columns='Gender', aggfunc='mean')
```

data	接收DataFrame。表示创建表的数据。无默认。
values	接收字符串。用于指定想要聚合的数据字段名，默认使用全部数据。默认为None。
index	接收string或list。表示行分组键。默认为None。
columns	接收string或list。表示列分组键。默认为None。
aggfunc	接收functions。表示聚合函数。默认为mean。
margins	接收boolean。表示汇总（Total）功能的开关，设为True后结果集中会出现名为“ALL”的行和列。默认为True。
dropna	接收boolean。表示是否删掉全为NaN的列。默认为False。

### ● 1.3.2 查看每一部电影不同性别的平均评分

#### ● pivot\_table函数主要的参数调节

- #index表示透视表的行
- #columns表示透视表的列
- #aggfunc表示对分析对象进行的分析，一般默认为求平均值，可以指定
- #margins表示添加每行每列求和的值，默认不添加。

## 1.2.2 使用crosstab函数创建交叉表

### crosstab函数

- 交叉表是一种特殊的透视表，主要用于计算分组频率。利用pandas提供的crosstab函数可以制作交叉表，crosstab函数的常用参数和使用格式如下。
- 由于交叉表是透视表的一种，其参数基本保持一致，不同之处在于crosstab函数中的index, columns, values填入的都是对应的从Dataframe中取出的某一行。

*pandas.crosstab(index, columns, values=None, rownames=None, colnames=None, aggfunc=None, margins=False, dropna=True, normalize=False)*

## 1.2 创建透视表与交叉表

### 1.2.2 使用crosstab函数创建交叉表

crosstab的常用参数及其说明

参数名称	说明
<b>index</b>	接收string或list。表示行索引键。无默认。
<b>columns</b>	接收string或list。表示列索引键。无默认。
<b>values</b>	接收array。表示聚合数据。默认为None。
<b>aggfunc</b>	接收function。表示聚合函数。默认为None。
<b>rownames</b>	表示行分组键名。无默认。
<b>colnames</b>	表示列分组键名。无默认。
<b>dropna</b>	接收boolearn。表示是否删掉全为NaN的。默认为False。
<b>margins</b>	接收boolearn。默认为True。汇总（Total）功能的开关，设为True后结果集中会出现名为“ALL”的行和列。
<b>normalize</b>	接收boolearn。表示是否对值进行标准化。默认为False。

完成以下练习：

数据如下图所示 打开crosstab.py文件练习

我们想要直观的看到此样本数据中，按照性别分组后统计他们用手习惯的次数，如右图所示，

请使用pivot\_table，crosstab分别完成

	Sample	Gender	Handedness
0	1	Female	Right-handed
1	2	Male	Left-handed
2	3	Female	Right-handed
3	4	Male	Right-handed
4	5	Male	Left-handed
5	6	Male	Right-handed
6	7	Female	Right-handed
7	8	Female	Left-handed
8	9	Male	Right-handed
9	10	Female	Right-handed

Handedness	Left-handed	Right-handed	All
Gender			
Female	1	4	5
Male	2	3	5
All	3	7	10