

**pandas数据处理的常用操作**

[](https://www.zhihu.com/people/lswbjtu)

[**数据科学汪**](https://www.zhihu.com/people/lswbjtu)

**引言**

众所周知，python的pandas库对于数据框的处理具有极其简化的强大功能，官方document中涉及的功能(function)更是有好几百个，要想做好数据处理的工作，那么你必须熟练掌握pandas中的各种操作，做到得心应手，拿来即用。本文针对我最近在处理数据的时候，掌握到的一些关于pandas的行和列操作的使用方法，我决定将它大概总结出来，分享给大家，也希望能帮助大家提升对数据处理的操作。本文的代码环境是Python3+Anaconda，推荐使用

**内容**

* 查看数据框
* 列切片
* 按列排序
* 合并数据框
* 分组Groupby
* 设置结果显示限制
* 读取写入excel

**一、查看数据框的常用操作**

**1、查看dataframe的前n行或尾n行**

df**=**pandas**.**Dataframe(columns**=**[],index**=**[],data**=**[]) *##创建一个Dataframe*

df**.**head(n**=**10) *## 显示前n行的数据*

df**.**tail(n**=**10) *## 显示尾n行的数据*

**2、查看索引名，列名，values**

df**.**index *##返回df的行索引值，是一个array*

df**.**columns *##返回df的列名，是一个array*

df**.**values *##返回df的所有值，是一个2D array*

**3、查看dataframe的一些统计特性**

df**.**describe('all') *##统计每列的min, max,mean,std,quantile*

df**.**dtypes *##返回每列数据的类型*

df**.**T *##转置数据框*

df**.**sort\_index(axis**=**1,ascending**=False**) *##按索引排序所有列，升序或者降序*

**二、常用的列切片操作**

**1、取一列的所有数据**

df['column\_name'] *##利用列名直接取某一列*

df[df**.**columns[index]] *##适合于你不知道列名，但是知道它在第几列*

**2、根据位置取行列数据df.loc**

df**.**loc[index] *##根据行的位置，取特定行数据（列全取）*

df**.**loc[[index],['a','b']] *##取index行的，ab两列数据*

df**.**loc[[index],'a':'b'] *##取index行的，列名为'a' 到 列名为 'b'的所有列*

*##总之，列不能通过索引来取数*

**3、根据索引位置来取数df.iloc**

df**.**iloc[0:10,0:10] *##切片后面的值取不到，即col\_index=10,row\_indx=10取不到*

df**.**iloc[[0,5,10],[1,8,10]] *##可按照需求，选择特定的行和列*

*##总之iloc之内的数据都是数字，不能是行名列名*

**4、根据条件，逻辑值索引取数**

df[df**.**A**>**0] *##取出A列中大于0的数，*

df[df['A']**.**isin(['one','two'])] *##取出A列中包含'one','two'的数据,这个功能很强大，##可以帮助我们filter出符合条件的数据。*

**5、给列赋值**

df['A']**=**np**.**array([1]**\***len(df)) *##用数组给某列赋值*

df**.**loc[:,['a','c']]**=**[] *##根据位置赋值*

*##知道如何取数，就能轻松给数据框赋值啦。*

**三、按列排序**

df**.**sort\_values(by**=**'column\_Name',ascending**=True**) *##按某列升序排序*

df**.**sort\_index(axis**=**1,ascending**=True**) *##索引排序*

**四、合并操作**

**1、将数据框的行或列合并（concat）**

pd**.**concat([df1[:],df2[:],**...**],axis**=**0) *##按列拼接数据，要求列数和列名一样*

pd**.**concat([df1,df2,**...**],axis**=**1）*##按行拼接数据，行数和行索引相同*

*##如果数据结构不一样，可以选择join="inner","outer",..sql中的操作*

**2、append将一行或多行数据添加**

df**.**append(df1[:],ignore\_index**=True**) *##将会重新设定index*

**3、将多个dataframe整合在一起Merge**

df**.**merge(df1,on**=**['column\_name',**...**],how**=**inner) *##内联表，根据主键来拼接*

how**=**"inner","left","right","outer"分别表示内连接，左连接，右连接，外连接。

*##具体如何连接，大家去温习一下sql中的表连接操作吧.*

**五、分组操作Groupby**

**1、Groupby for splitting 把数据分成已有的几种类别**

grouped**=**df**.**groupby(key) *##将某个主键按照类别分组，默认是列主键*

grouped**=**df**.**groupby(key,axis**=**1) *##按照某个key分组，行操作*

grouped**=**df**.**groupby([key1,key2,**...**]) *##可以依次group多个key。*

grouped**.**groups *##返回分组的结果*

grouped**.**get\_group('a') *## 选择其中一个分组的类别，查看该类别的数据*

**2、Groupby for aggregation 分组聚合**

grouped**.**aggregate(np**.**sum) *##分组求和，常见操作*

grouped**.**size() *##分组统计数量*

grouped**.**describe() *##分组查看描述统计结果*

**3、Groupby for applying 分组求各种函数**

grouped**.**agg([np**.**sum,np**.**std,np**.**mean]) *##同时求和，均值方差。*

grouped**.**apply(**lambda** x: function(x)) *##可以接上apply函数，进行自定义操作*

**4、Groupby for filtering 分组过滤数据**

grouped**.**filter(**lambda** x : len(x)**>**2,dropna**=True**) *##类似这种filter操作*

*##根据自己需求，都能够相应地实现。*

**六、设置结果显示限制**

很多情况下，我们查看数据操作的结果时，很多想要的行或列数据由于输出限制，而无法看到，这时，我们可以重新设置显示结果的限制条件，来获得更多想查看的数据结果。

pd**.**set\_option("display.height",200) *##设置显示结果的高度*

pd**.**set\_option("display.max\_seq\_items",200) *##设置序列显示的最大个数*

pd**.**set\_option("display.max\_columns",120) *##设置数据框显示的列数*

pd**.**set\_option("display.max\_rows",50) *##设置数据框显示的行数*

**七、数据框写入excel或者csv**

很多情况下，你的分析汇总结果需要保存到excel文件中，发给你的上司。因此这项技能你必须学会。基本的函数使用如下：

*##先是读取数据*

pd**.**read\_csv(file\_path,header**=**,sep**=**',',index\_col**=**,**...**) *##常用的取数据函数*

pd**.**read\_excel(file\_path,sheetName**=**'',startcol**=**,startrow**=**,header**=**,**...**)

*##保存数据相对规范的话是如下代码：*

writer**=**pd**.**ExcelWriter('excel name.xlsx') *##新建一个excel*

df**.**to\_excel(writer,sheetName**=**'dfName',startcol**=**,startrow**=**,**...**)

writer**.**save()

至此，相信大家基本已经了解了一些常用的pandas操作，下次我会讲一些更加深入的东西，不局限于pandas，包括scikit-learn，machine learning，deep learning等知识，如果能对大家有一点点的帮助，我就很满足啦～喜欢的话可以关注一下喔！

编辑于 2019-03-13 02:13