Python 数据清洗

pandas如何复制筛选出的一些行，形成一个新的DataFrame？

关注问题 写回答 邀请回答 好问题 添加评论 分享 ···

7 个回答 默认排序

测试开发实战
http://t.cn/EUKiwsY python 测试

8 人赞同了该回答

参考下 数据分析工具PANDAS技巧-如何过滤数据

选摘了一小部分，详细的参见上述原文

在本文中，我们将介绍在Python中过滤pandas数据帧的各种方法。 数据过滤是最常见的数据操作操作之一。 它类似于SQL中的WHERE子句，或者必须在MS Excel中使用过滤器根据某些条件选择特定行。 就速度而言，python执行过滤和聚合更佳。 它有很棒的库：pandas。 Pandas是在numpy包之上构建的，它是用C语言编写的，这是一种低级语言。 因此，使用pandas包进行数据操作是处理大型数据集的快速而智能的方法。

▲ 赞同 4 ▼ 2 条评论 分享 收藏 喜欢 收起

继续浏览内容

知乎 发现更大的世界 打开

Chrome 继续

- 选择在2019年1月1日之后开立帐户的所有活跃客户
- 提取过去6个月内进行超过3笔交易的所有客户的详细信息
- 获取在组织中工作超过3年且在过去两年中获得最高评级的员工的信息
- 分析投诉数据并确定在过去1年内提交超过5个投诉的客户
- 提取人均收入超过40K美元的地铁城市的详细信息

导入数据

我们将使用包含2013年从纽约出发的航班详情的数据集。该数据集有32735行和16列。下载 https://itbooks.pipipan.com/fs/18113597-393403297。

表头如下：

```
['year', 'month', 'day', 'dep_time', 'dep_delay', 'arr_time', 'arr_delay', 'carrier',
```

导入数据

```
import pandas as pd
df = pd.read_csv("nycflights.csv")
```

使用列值过滤

选择JetBlue Airways航班详细信息，其中包含2个字母的运营商代码B6 ，起源于JFK机场。

- 方法1：DataFrame方式

```
>>> newdf = df[(df.origin == "JFK") & (df.carrier == "B6")]
>>> newdf.head()
    year  month  day  dep_time  dep_delay  arr_time  arr_delay carrier tailnum  flight
7   2013      8   13      1920       85.0      2032       71.0      B6  N284JB    1407
10  2013      6   17       940        5.0      1050       -4.0      B6  N351JB      20
14  2013     10   21      1217       -4.0      1322       -6.0      B6  N192JB      34
23  2013      7    7      2310      105.0       201      127.0      B6  N506JB      97
35  2013      4   12       840       20.0      1240       28.0      B6  N655JB     403
```

这部分代码(df.origin == "JFK") & (df.carrier == "B6")返回True / False。 条件匹配时为真，条件不匹配时为假。 稍后它在df内传递并返回与True对应的所有行。 它返回4166行。

- 方法2：查询函数

在pandas包中，有多种方法可以执行过滤。 上面的代码也可以像下面显示的代码一样编写。 此方法更优雅，更易读，每次指定列（变量）时都不需要提及数据框名称。

```
>>> newdf = df.query('origin == "JFK" & carrier == "B6"')
```

- 方法3：loc函数

loc是位置术语的缩写。 所有这三种方法都返回相同的输出。 这只是一种不同的过滤行的方法。

```
>>> newdf = df.loc[(df.origin == "JFK") & (df.carrier == "B6")]
```

按行和列位置过滤Pandas数据帧

假设您想按位置选择特定的行（假设从第二行到第五行）。 我们可以使用df.iloc[ ]函数。 python中的索引从零开始。 df.iloc [0：5，]指第一至第五行（此处不包括终点第6行）。 df.iloc [0：5，]相当于df.iloc [：5，]

```
df.iloc[:5,] #First 5 rows
df.iloc[1:5,] #Second to Fifth row
df.iloc[5,0] #Sixth row and 1st column
df.iloc[1:5,0] #Second to Fifth row, first column
df.iloc[1:5,:5] #Second to Fifth row, first 5 columns
df.iloc[2:7,1:3] #Third to Seventh row, 2nd and 3rd column
```

loc根据索引标签考虑行。 而iloc根据索引中的位置考虑行，因此它只需要整数。 让我们创建一个示例数据进行说明

```
>>> x
  col1
9     1
8     3
7     5
6     7
0     9
1    11
2    13
3    15
4    17
5    19
>>> x.iloc[0:5]
  col1
9     1
8     3
7     5
6     7
0     9
>>> x.loc[0:5]
  col1
0     9
1    11
2    13
3    15
4    17
5    19
```

编辑于 2019-08-21

▲ 赞同 8 ▼ 添加评论 分享 收藏 喜欢 收起

梦境之末
金融IT

16 人赞同了该回答

如果是知道要对某列值进行筛选
pd=pd[pd['列名字'].isin([行列表])]
可以用布尔运算符通过多列的值筛选行
pd=pd[pd['列名字'].isin([行列表]) & pd['列名字'].isin([行列表])]

编辑于 2016-03-25

▲ 赞同 16 ▼ 4 条评论 分享 收藏 喜欢

张幼薇
经济学爱好者

14 人赞同了该回答

假设你的大表是data, 子表是sub1, 而且你筛选条件有字段，也有数值，而且就保留那么几个关键的其他列C,D

那么：
sub1=data.loc [(data['列A ']=='筛选的数值') & data['列B '].str.contains('筛选的关键字') ，['C','D']]
然后随便你对sub1进行进一步处理，比如清蒸啦，红烧啦，油闷啦。。。

发布于 2016-12-06

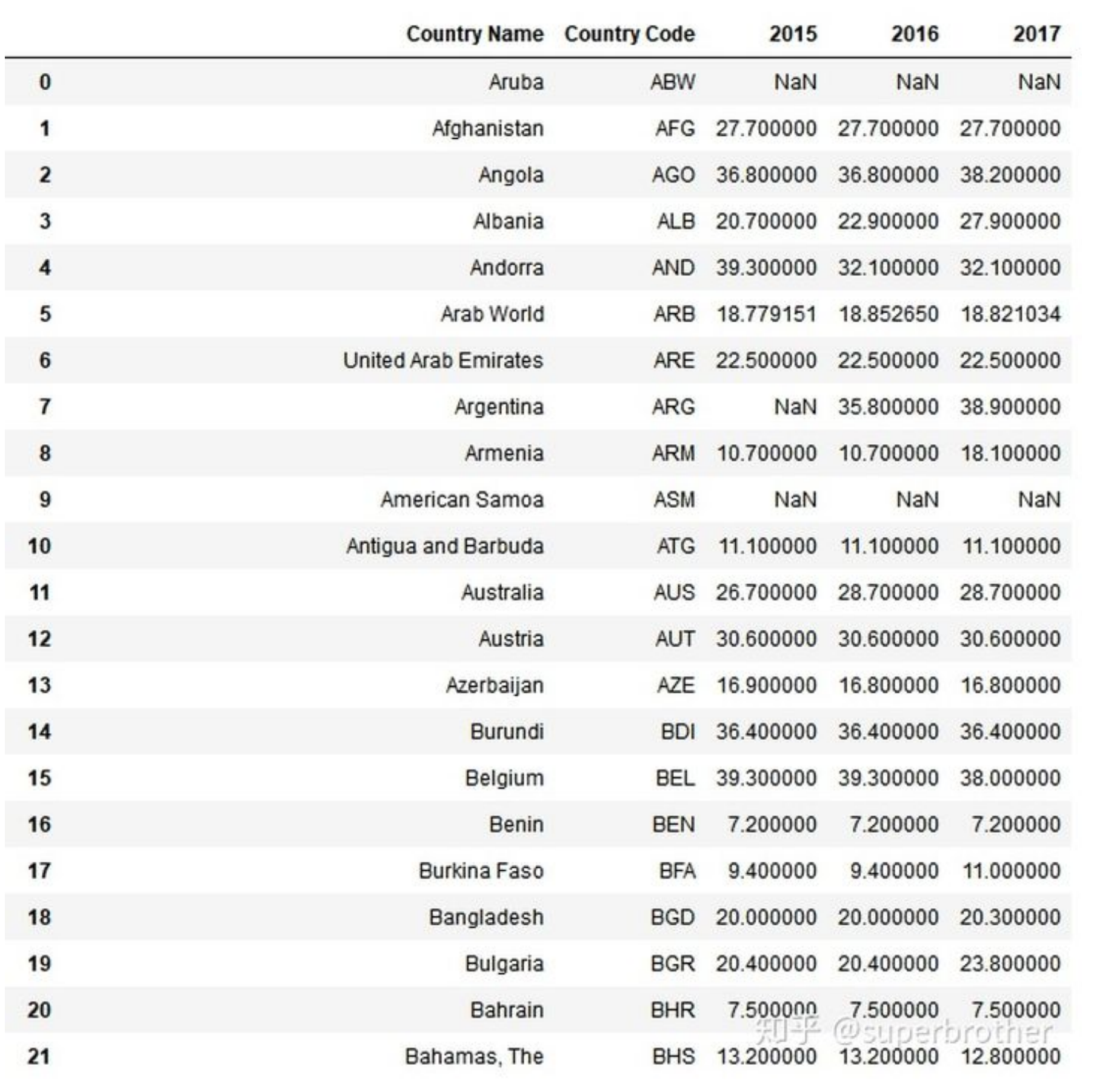
▲ 赞同 14 ▼ 1 条评论 分享 收藏 喜欢

superbrother
清华大学 土木工程博士在读

4 人赞同了该回答

实际上，这个**问题的本质还是DataFrame的索引**

下面来举个栗子，现在有一个叫做all_data的DataFrame，是这样的



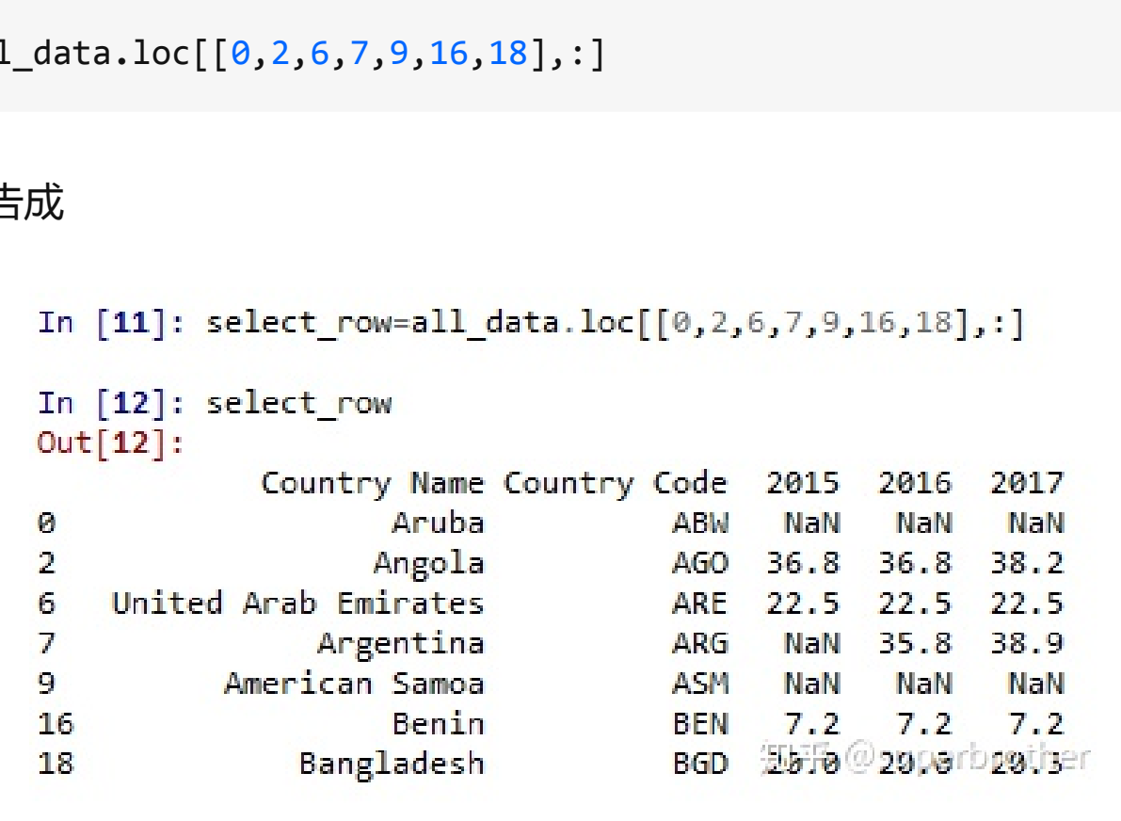
	Country Name	Country Code	2015	2016	2017
0	Aruba	ABW	NaN	NaN	NaN
1	Afghanistan	AFG	27.700000	27.700000	27.700000
2	Angola	AGO	36.800000	36.800000	38.200000
3	Albania	ALB	20.700000	22.900000	27.900000
4	Andorra	AND	39.300000	32.100000	32.100000
5	Arab World	ARB	18.779151	18.852650	18.821034
6	United Arab Emirates	ARE	22.500000	22.500000	22.500000
7	Argentina	ARG	NaN	35.800000	38.900000
8	Armenia	ARM	10.700000	10.700000	18.100000
9	American Samoa	ASM	NaN	NaN	NaN
10	Antigua and Barbuda	ATG	11.100000	11.100000	11.100000
11	Australia	AUS	26.700000	28.700000	28.700000
12	Austria	AUT	30.600000	30.600000	30.600000
13	Azerbaijan	AZE	16.900000	16.800000	16.800000
14	Burundi	BDI	36.400000	36.400000	36.400000
15	Belgium	BEL	39.300000	39.300000	38.000000
16	Benin	BEN	7.200000	7.200000	7.200000
17	Burkina Faso	BFA	9.400000	9.400000	11.000000
18	Bangladesh	BGD	20.000000	20.000000	20.300000
19	Bulgaria	BGR	20.400000	20.400000	23.800000
20	Bahrain	BHR	7.500000	7.500000	7.500000
21	Bahamas, The	BHS	13.200000	13.200000	12.800000

all_data

现在要选择其中Index为0,2,6,7,9,16,18的行组成新的DataFrame，利用loc完成

```
select_row=all_data.loc[[0,2,6,7,9,16,18],:]
```

可以看到，大功告成



```
In [11]: select_row=all_data.loc[[0,2,6,7,9,16,18],:]

In [12]: select_row
Out[12]:
             Country Name Country Code  2015  2016  2017
0                   Aruba          ABW   NaN   NaN   NaN
2                  Angola          AGO  36.8  36.8  38.2
6    United Arab Emirates          ARE  22.5  22.5  22.5
7               Argentina          ARG   NaN  35.8  38.9
9          American Samoa          ASM   NaN   NaN   NaN
16                  Benin          BEN   7.2   7.2   7.2
18             Bangladesh          BGD  20.0  20.0  20.3
```

select_row

顺便说一下：loc与iloc的区别：

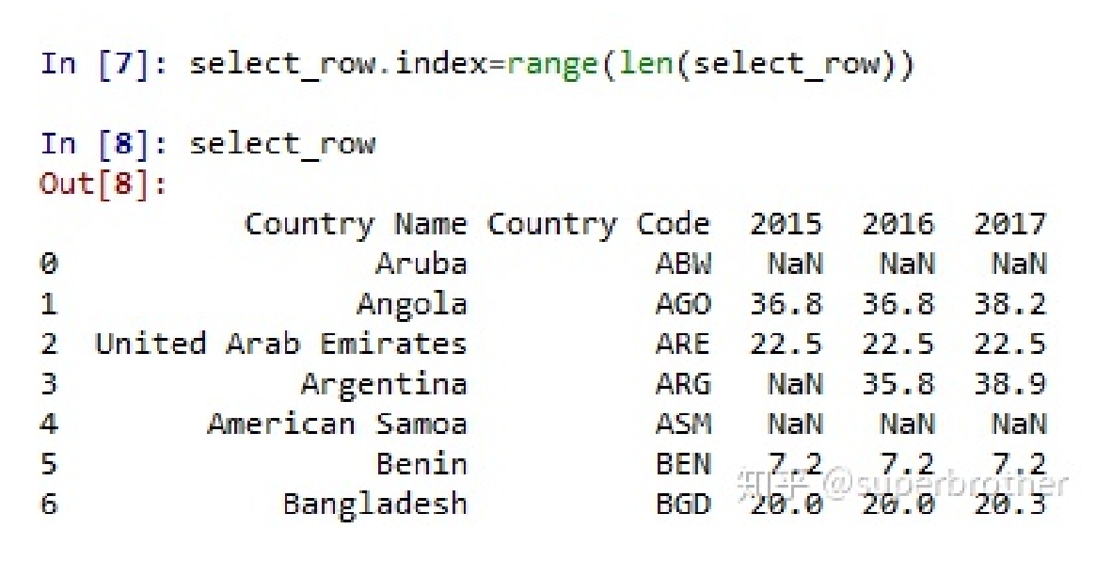
loc是以index及column的key作为索引对象的

而iloc是以行列所在的整数索引作为对象（从0开始）

有读者觉得这不是“新的”DataFrame啊，因为原来的index还在啊。没关系，不必这么执念，**一行代码改index：**

```
select_row.index=range(len(select_row))
```

效果如下图：



```
In [7]: select_row.index=range(len(select_row))

In [8]: select_row
Out[8]:
            Country Name Country Code  2015  2016  2017
0                  Aruba          ABW   NaN   NaN   NaN
1                 Angola          AGO  36.8  36.8  38.2
2   United Arab Emirates          ARE  22.5  22.5  22.5
3              Argentina          ARG   NaN  35.8  38.9
4         American Samoa          ASM   NaN   NaN   NaN
5                  Benin          BEN   7.2   7.2   7.2
6             Bangladesh          BGD  20.0  20.0  20.3
```

更新index

好了，心心念念的“新”DataFrame生成了！

编辑于 2018-05-08

星空流
读书/数据分析/摄影爱好者

最常用的一般就是[ ]、loc[ ]、iloc[ ]，搭配一些条件判断就可以实现数据选取。

当然还有更加简洁高效的方法，比如query()，写法很简单，效率也不错。

loc[ ]的用法可以参考下面这个：

星空流：Python_Pandas_Select_Data_loc[ ]
zhuanlan.zhihu.com

发布于 2020-03-28

▲ 赞同 ▼ 添加评论 分享 收藏 喜欢

石头三颗
Python和JavaScript爱好者

要么百度搜索dataframe，要么把代码贴出来。

发布于 2016-03-23

▲ 赞同 ▼ 1 条评论 分享 收藏 喜欢

keail

直接赋到一个新的变量上就好了

发布于 2017-07-24

▲ 赞同 ▼ 添加评论 分享 收藏 喜欢

写回答