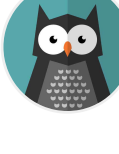
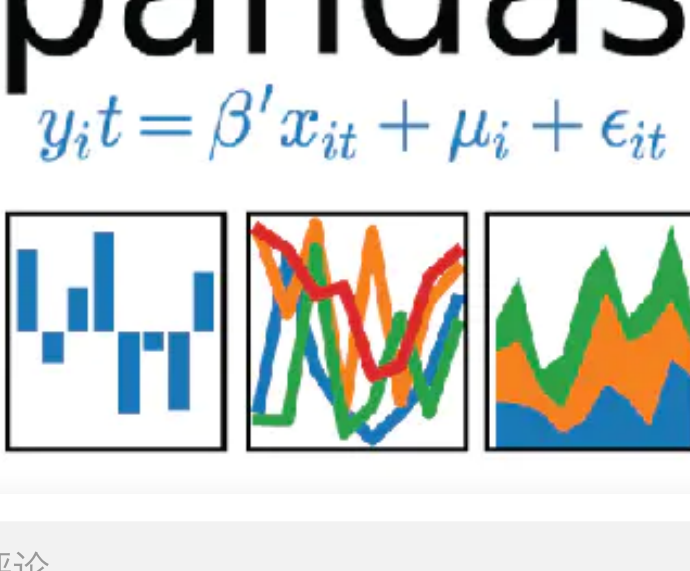


数据分析工具PANDAS技巧-如何过滤数据

python测试开发

2019.08.19 11:25:31 字数 947 阅读 4,278

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。



	BandName	WavelengthMax	WavelengthMin
0	CoastalAerosol	450	430
1	Blue	510	450
2	Green	590	530
3	Red	670	640
4	NearInfrared	880	850
5	ShortWaveInfrared_1	1650	1570
6	ShortWaveInfrared_2	2290	2110
7	Cirrus	1380	1360

写下你的评论...

评论1 赞14

图片.png

数据过滤的示例

它是预测建模或任何报告项目的数据准备的最初步骤之一。它也被称为“子集数据”。请参阅下面的一些数据过滤示例。

- 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：

```
1 | ['year', 'month', 'day', 'dep_time', 'dep_delay', 'arr_time', 'arr_delay', 'carrier', 'tailnum']
```

导入数据

```
1 | import pandas as pd
2 | 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

```
1 |
2 | >>> newdf = df[(df.origin == "JFK") & (df.carrier == "B6")]
3 | >>> newdf.head()
4 |
5 |   year  month  day  dep_time  dep_delay  arr_time  arr_delay  carrier  tailnum  flight  origin
6 |  7  2013     8   13   1920     85.0    2032     71.0      B6   N284JB   1407   JFK
7 | 10  2013     6   17   940      5.0    1059     -4.0      B6   N351JB   1407   JFK
8 | 14  2013    10   21   1217     -4.0    1322     -6.0      B6   N192JB   34    JFK
9 | 23  2013     7    7   2310    105.0    201     127.0     B6   N56JB   97    JFK
10 | 35  2013     4   12    840     20.0    1240     28.0     B6   N655JB  403   JFK
```

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。

```
1 |
2 | >>> newdf = df.query('origin == "JFK" & carrier == "B6"')
3 |
```

- 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。

```
1 |
2 | >>> newdf = df.loc[(df.origin == "JFK") & (df.carrier == "B6")]
3 |
```

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,]

```
1 | df.iloc[5:] #First 5 rows
2 | df.iloc[1:5] #Second to Fifth row
3 | df.iloc[5,0] #Sixth row and 1st column
4 | df.iloc[1:5,0] #Second to Fifth row, first column
5 | df.iloc[1:5,1:5] #Second to Fifth row, first 5 columns
6 | 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

```
1 | >>> x
2 |    col1
3 | 0     1
4 | 1     3
5 | 2     5
6 | 3     7
7 | 4     9
8 | 5    11
9 | 6    13
10 | 7    15
11 | 8    17
12 | 9    19
13 | >>> x.iloc[0:5]
14 |    col1
15 | 0     1
16 | 1     3
17 | 2     5
18 | 3     7
19 | 4     9
20 | >>> x.loc[0:5]
21 |    col1
22 | 0     9
23 | 1    11
24 | 2    13
25 | 3    15
26 | 4    17
27 | 5    19
```

参考资料

- [python测试开发项目实战-目录](#)
- [python工具书籍下载-持续更新](#)
- [python 3.7极速入门教程 - 目录](#)
- [讨论qq群630011153 144081101](#)
- [原文地址](#)
- [本文涉及的python测试开发库 谢谢点赞！](#)
- [本文相关海量书籍下载](#)
- <https://www.listendata.com/2019/06/pandas-drop-columns-from-dataframe.html>
- [python数据分析数据科学中文英文工具书籍下载-持续更新](#)

按行位置和列名称过滤pandas数据帧

```
1 |
2 | >>> df.loc[df.index[0:5],["origin","dest"]]
3 |
4 |   origin dest
5 | 0   JFK   LAX
6 | 1   JFK   SFO
7 | 2   JFK   LAX
8 | 3   JFK   FRA
9 | 4   LGA   ORD
```

列中选择多个值

```
1 | >>> newdf = df[df.origin.isin(["JFK", "LGA"])]
```

不等于

```
1 | >>> newdf = df.loc[(df.origin != "JFK") & (df.carrier == "B6")]
2 | >>> pd.unique(newdf.origin)
3 | array(['LGA', 'EWR'], dtype=object)
4 |
```

如何否定整个条件


```
1 | >>> newdf = df[~((df.origin == "JFK") & (df.carrier == "B6"))]
2 |
```


选择非缺失数据


```
1 | >>> newdf = df[~((df.origin == "JFK") & (df.carrier == "B6"))]
2 |
```


过滤Pandas DataFrame中的字符串

```
1 | >>> df = pd.DataFrame({"var1": ["AA_2", "B_1", "C_2", "A_2"]})
2 | >>> df
3 |
4 |   var1
5 | 0  AA_2
6 | 1  B_1
7 | 2  C_2
8 | 3  A_2
9 | >>> df[df['var1'].str[0] == 'A']
10 |
11 |   var1
12 | 0  AA_2
13 | 3  A_2
14 | >>> df[df['var1'].str.len()>3]
15 |
16 |   var1
17 | 0  AA_2
18 | 1  B_1
19 | 3  A_2
```

14人点赞



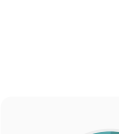
数据分析



"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下

python测试开发

码农，涉及python自动化测试、人工智能、爬虫、数据分析...

总资产2,304 (约123.85元) 共写了67.1W字 获得3,712个赞 共129,640个粉丝

关注

写下你的评论...

全部评论1 只看作者 按时间倒序 按时间正序

我的一生是传奇

2楼 2020.02.23 04:17


文笔优美，向你学习。

赞

回复

被以下专题收入，发现更多相似内容

工具癖

软件测试Python

python数据

推荐阅读

更多精彩内容>

Python 数据分析工具包 Pandas 学习笔记 - 1 - Pandas 三大对象: Se...

Pandas是一个基于Python的快速、强大、灵活、易于使用的开源数据分析工具包。Pandas是一个在...

苏文影月志 阅读 1,852 评论 0 赞 73

Python爬取股票数据，让你感受一下什么是一秒钟两千条数据

本文的文学及图片过滤网络，可以学习，交流使用，不具有任何商业用途，如有问题请及时联系我们以作处理。以下文章来源于...

松鼠爱吃饼干 阅读 194 评论 0 赞 2

python pandas -->loc、iloc用法

基础数据如下：一、loc 主要通过行标签、索引行数据，划重点，标签！标签！标签！ 1.1、loc选定index...

散宜 阅读 1,221 评论 0 赞 2

Python3.9的7个特性

作者[PADHMA编译]VK来源[Analytics Vidhya 介绍 正如著名作家韦恩·W·戴尔所说，改变你...

人工智能遇见霸剑 阅读 849 评论 0 赞 9

回归模型评价指标

机器学习回归模型的评价方法 1 均方误差 (mean squared error,mse) 1.定义：观察值与真实值...

温故知新h 阅读 115 评论 0 赞 1



python测试开发

总资产2,304 (约123.85元)

关注

软件工程快速入门教程8- 多层架构

阅读 236

测试管理快速入门7测试监控和测试控制

阅读 243

未来的软件测试需要具备那些技能？

阅读 242

