

ForeseeMark

码龄8年 暂无认证

25

16万+

7万+

20万+

等级

原创

周排名

总排名

访问

等级

1348

27

49

33

120

积分

粉丝

获赞

评论

收藏

头像

私信

关注

搜博文文章

Q

热门文章

pandas中DataFrame的查询方法 ([], loc, iloc, at, iat, ix) 92294

Linux shell 用for循环100次的方法 23273

Windows的Tex环境配置 (TeX Live+WinEdt+SumatraPDF) 20666

TeX Live安装指南 13784

Python合并字典键值并去除重复元素 11679

分类专栏

杂文

1篇

环境配置

7篇

python技巧

5篇

算法题

13篇

图论

3篇

指针

1篇

最新评论

Ubuntu安装ROUGE教程
有点费头发、~、make 或者make test出现错误的、不要放弃、可以删除已经解压的...
Linux shell 用for循环100次的方法
料大小虫: 你说不是四种、是三种方法哈、再者、for i `seq 1 100`少了in、应该是for...
Linux shell 用for循环100次的方法
盲盲的烂笔头: 干货满满、很详细、评论占个坑。
Ubuntu安装ROUGE教程
iiiiiwonderful: 你好 这篇文章的网盘链接失效了 你的可以给我发一下么 110970143...
TeX Live安装指南
wiki_hi: 感谢! 清华的镜像慢、ctan超快...
Linux shell 用for循环100次的方法

最新文章

Mac 下 vim 环境配置及插件设置
vim多行注释、取消多行注释、多行复制、多行删除
Linux shell 用for循环100次的方法

2018年 1篇

2017年 18篇

2016年 9篇

目录

数据介绍

[]切片方法

loc

iloc

at

iat

ix

pandas中DataFrame的查询方法 ([], loc, iloc, at, iat, ix)

ForeseeMark

2017-03-23 17:19:25

92340 收藏 55

分类专栏: python技巧 文章标签: python pandas 索引 查询 DataFrame

版权

pandas为我们提供了多种切片方法, 而要是太不了解这些方法, 就会经常容易混淆。下面举例对这些切片方法进行说明。

数据介绍

先随机生成一组数据:

```
1 In [5]: rnd_1 = [random.randrange(1,20) for x in xrange(1000)]
2         ...: rnd_2 = [random.randrange(1,20) for x in xrange(1000)]
3         ...: rnd_3 = [random.randrange(1,20) for x in xrange(1000)]
4         ...: fecha = pd.date_range("2012-4-10", "2015-1-4")
5
6         ...: data = pd.DataFrame({'fecha':fecha, 'rnd_1': rnd_1, 'rnd_2': rnd_2, 'rnd_3': rnd_3})
7
8 In [6]: data.describe()
9 Out[6]:
```

| | rnd_1 | rnd_2 | rnd_3 |
|-------|-------------|-------------|-------------|
| count | 1000.000000 | 1000.000000 | 1000.000000 |
| mean | 9.946000 | 9.825000 | 9.894000 |
| std | 5.553911 | 5.559432 | 5.423484 |
| min | 1.000000 | 1.000000 | 1.000000 |
| 25% | 5.000000 | 5.000000 | 5.000000 |
| 50% | 10.000000 | 10.0000 | |
| 75% | 15.000000 | 15.0000 | |

点赞Mark关注该博主, 随时了解TA的最新博文

点赞过26 评论2 分享 收藏55 打赏 举报 关注 一键三连

[]切片方法

使用方括号能够对DataFrame进行切片, 有点类似于python的列表切片。按照索引能够实现行选择或列选择或区域选择。

```
1 # 行选择
2 In [7]: data[1:5]
3 Out[7]:
```

| | fecha | rnd_1 | rnd_2 | rnd_3 |
|---|------------|-------|-------|-------|
| 1 | 2012-04-11 | 1 | 16 | 3 |
| 2 | 2012-04-12 | 7 | 6 | 1 |
| 3 | 2012-04-13 | 2 | 16 | 7 |
| 4 | 2012-04-14 | 4 | 17 | 7 |

```
10 # 列选择
11 In [10]: data[['rnd_1', 'rnd_3']]
12 Out[10]:
```

| | rnd_1 | rnd_3 |
|-------------------------|-------|-------|
| 0 | 8 | 12 |
| 1 | 1 | 3 |
| 2 | 7 | 1 |
| 3 | 2 | 7 |
| 4 | 4 | 7 |
| 5 | 12 | 8 |
| 6 | 2 | 12 |
| 7 | 9 | 8 |
| 8 | 13 | 17 |
| 9 | 4 | 7 |
| 10 | 14 | 14 |
| 11 | 19 | 16 |
| 12 | 2 | 12 |
| 13 | 15 | 18 |
| 14 | 13 | 18 |
| 15 | 13 | 11 |
| 16 | 17 | 7 |
| 17 | 14 | 10 |
| 18 | 9 | 6 |
| 19 | 11 | 15 |
| 20 | 16 | 13 |
| 21 | 18 | 9 |
| 22 | 1 | 18 |
| 23 | 4 | 3 |
| 24 | 6 | 11 |
| 25 | 2 | 13 |
| 26 | 7 | 17 |
| 27 | 11 | 8 |
| 28 | 3 | 12 |
| 29 | 4 | 2 |
| ... | ... | ... |
| 970 | 8 | 14 |
| 971 | 19 | 5 |
| 972 | 13 | 2 |
| 973 | 8 | 10 |
| 974 | 8 | 17 |
| 975 | 6 | 16 |
| 976 | 3 | 2 |
| 977 | 12 | 6 |
| 978 | 12 | 10 |
| 979 | 15 | 13 |
| 980 | 8 | 4 |
| 981 | 17 | 3 |
| 982 | 1 | 17 |
| 983 | 11 | 5 |
| 984 | 7 | 7 |
| 985 | 13 | 14 |
| 986 | 6 | 19 |
| 987 | 13 | 9 |
| 988 | 3 | 15 |
| 989 | 19 | 6 |
| 990 | 7 | 11 |
| 991 | 11 | 7 |
| 992 | 19 | 12 |
| 993 | 2 | 15 |
| 994 | 10 | 4 |
| 995 | 14 | 13 |
| 996 | 12 | 11 |
| 997 | 11 | 15 |
| 998 | 17 | 14 |
| 999 | 3 | 8 |
| [1000 rows x 2 columns] | | |
| | rnd_1 | rnd_2 |
| 0 | 8 | 17 |
| 1 | 1 | 16 |
| 2 | 7 | 6 |
| 3 | 2 | 16 |
| 4 | 4 | 17 |
| 5 | 12 | 19 |
| 6 | 2 | 7 |

不过对于多列选择, 不能像行选择时一样使用1: 5这样的方法来选择。

```
1 In [12]: data[['rnd_1':'rnd_3']]
2 File "<ipython-input-13-6291b6a83eb0>", line 1
3     data[['rnd_1':'rnd_3']]
4           ^
5 SyntaxError: invalid syntax
```

loc

loc可以让你按照索引来进行行列选择。

```
1 In [13]: data.loc[1:5]
2 Out[13]:
```

| | fecha | rnd_1 | rnd_2 | rnd_3 |
|---|------------|-------|-------|-------|
| 1 | 2012-04-11 | 1 | 16 | 3 |
| 2 | 2012-04-12 | 7 | 6 | 1 |
| 3 | 2012-04-13 | 2 | 16 | 7 |
| 4 | 2012-04-14 | 4 | 17 | 7 |
| 5 | 2012-04-15 | 12 | 19 | 8 |

这里需要注意的是, loc与第一种方法不同之处在于会把第5行也选择进去, 而第一种方法只会选择到第4行为止。

```
1 data.loc[2:4, ['rnd_2', 'fecha']]
2 Out[14]:
```

| | rnd_2 | fecha |
|---|-------|------------|
| 2 | 6 | 2012-04-12 |
| 3 | 16 | 2012-04-13 |
| 4 | 17 | 2012-04-14 |

loc能够选择在两个特定日期之间的数据, 需要注意的是这两个日期都必须要在索引中。

```
1 In [15]: data_fecha = data.set_index('fecha')
2         ...: data_fecha.head()
3 Out[15]:
```

| | rnd_1 | rnd_2 | rnd_3 |
|------------|-------|-------|-------|
| fecha | | | |
| 2012-04-10 | 8 | 17 | 12 |
| 2012-04-11 | 1 | 16 | 3 |
| 2012-04-12 | 7 | 6 | 1 |
| 2012-04-13 | 2 | 16 | 7 |
| 2012-04-14 | 4 | 17 | 7 |

```
11
12 In [16]: # 生成两个特定日期
13         ...: fecha_1 = dt.datetime(2013, 4, 14)
14         ...: fecha_2 = dt.datetime(2013, 4, 18)
15         ...:
16         ...: # 生成切片数据
17         ...: data_fecha.loc[fecha_1: fecha_2]
18 Out[16]:
```

| | rnd_1 | rnd_2 | rnd_3 |
|------------|-------|-------|-------|
| fecha | | | |
| 2013-04-14 | 17 | 10 | 5 |
| 2013-04-15 | 14 | 4 | 9 |
| 2013-04-16 | 1 | 2 | 18 |
| 2013-04-17 | 9 | 15 | 1 |
| 2013-04-18 | 16 | 7 | 17 |

更新: 如果没有特殊需求, 强烈建议使用loc而尽量少使用[], 因为loc在对DataFrame进行重新赋值操作时会避免chained indexing问题, 使用[]时编译器很可能会给出SettingWithCopy的警告。具体可以参见官方文档: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

iloc

如果说loc是按照索引 (index) 的值来选取的话, 那么iloc就是按照索引的位置来进行选取。iloc不关心索引的具体值是多少, 只关心位置是多少, 所以使用iloc时方括号中只能使用数值。

```
1 # 行选择
2 In [17]: data_fecha[10: 15]
3 Out[17]:
```

| | rnd_1 | rnd_2 | rnd_3 |
|------------|-------|-------|-------|
| fecha | | | |
| 2012-04-20 | 14 | 6 | 14 |
| 2012-04-21 | 19 | 14 | 16 |
| 2012-04-22 | 2 | 6 | 12 |
| 2012-04-23 | 15 | 8 | 18 |
| 2012-04-24 | 13 | 8 | 18 |

```
12 # 列选择
13 In [18]: data_fecha.iloc[:,[1,2]].head()
14 Out[18]:
```

| | rnd_2 | rnd_3 |
|------------|-------|-------|
| fecha | | |
| 2012-04-10 | 17 | 12 |
| 2012-04-11 | 16 | 3 |
| 2012-04-12 | 6 | 1 |
| 2012-04-13 | 16 | 7 |
| 2012-04-14 | 17 | 7 |

```
23 # 切片选择
24 In [19]: data_fecha.iloc[[1,12,34],[0,2]]
25 Out[19]:
```

| | rnd_1 | rnd_3 |
|------------|-------|-------|
| fecha | | |
| 2012-04-11 | 1 | 3 |
| 2012-04-22 | 2 | 12 |
| 2012-05-14 | 17 | 10 |

at

at的使用方法与loc类似, 但是比loc有更快的访问数据的速度, 而且只能访问单个元素, 不能访问多个元素。

```
1 In [20]: timeit data_fecha.at[fecha_1,'rnd_1']
2         ...: data_fecha.head()
3 Out[15]:
```

| | rnd_1 | rnd_2 | rnd_3 |
|------------|-------|-------|-------|
| fecha | | | |
| 2012-04-10 | 8 | 17 | 12 |
| 2012-04-11 | 1 | 16 | 3 |
| 2012-04-12 | 7 | 6 | 1 |
| 2012-04-13 | 2 | 16 | 7 |
| 2012-04-14 | 4 | 17 | 7 |

```
11
12 In [16]: # 生成两个特定日期
13         ...: fecha_1 = dt.datetime(2013, 4, 14)
14         ...: fecha_2 = dt.datetime(2013, 4, 18)
15         ...:
16         ...: # 生成切片数据
17         ...: data_fecha.loc[fecha_1: fecha_2]
18 Out[16]:
```

| | rnd_1 | rnd_2 | rnd_3 |
|------------|-------|-------|-------|
| fecha | | | |
| 2013-04-14 | 17 | 10 | 5 |
| 2013-04-15 | 14 | 4 | 9 |
| 2013-04-16 | 1 | 2 | 18 |
| 2013-04-17 | 9 | 15 | 1 |
| 2013-04-18 | 16 | 7 | 17 |

```
21
22
23 # 切片选择
24 In [19]: data_fecha.iloc[[1,12,34],[0,2]]
25 Out[19]:
```

| | rnd_1 | rnd_3 |
|------------|-------|-------|
| fecha | | |
| 2012-04-11 | 1 | 3 |
| 2012-04-22 | 2 | 12 |
| 2012-05-14 | 17 | 10 |

```
1 In [20]: timeit data_fecha.at[fecha_1,'rnd_1']
2         ...: data_fecha.head()
3 Out[15]:
```

| | rnd_1 | rnd_2 | rnd_3 |
|------------|-------|-------|-------|
| fecha | | | |
| 2012-04-10 | 8 | 17 | 12 |
| 2012-04-11 | 1 | 16 | 3 |
| 2012-04-12 | 7 | 6 | 1 |
| 2012-04-13 | 2 | 16 | 7 |
| 2012-04-14 | 4 | 17 | 7 |

```
11
12 In [16]: # 生成两个特定日期
13         ...: fecha_1 = dt.datetime(2013, 4, 14)
14         ...: fecha_2 = dt.datetime(2013, 4, 18)
15         ...:
16         ...: # 生成切片数据
17         ...: data_fecha.loc[fecha_1: fecha_2]
18 Out[16]:
```

| | rnd_1 | rnd_2 | rnd_3 |
|------------|-------|-------|-------|
| fecha | | | |
| 2013-04-14 | 17 | 10 | 5 |
| 2013-04-15 | 14 | 4 | 9 |
| 2013-04-16 | 1 | 2 | 18 |
| 2013-04-17 | 9 | 15 | 1 |
| 2013-04-18 | 16 | 7 | 17 |

```
21
22
23 # 切片选择
24 In [19]: data_fecha.iloc[[1,12,34],[0,2]]
25 Out[19]:
```

| | rnd_1 | rnd_3 |
|------------|-------|-------|
| fecha | | |
| 2012-04-11 | 1 | 3 |
| 2012-04-22 | 2 | 12 |
| 2012-05-14 | 17 | 10 |

ix

以上说过的几种方法都要求查询的秩在索引中, 或者位置不超过长度范围, 而ix允许你得到不在DataFrame索引中的数据。

```
1 In [28]: date_1 = dt.datetime(2013, 1, 10, 8, 30)
2         ...: date_2 = dt.datetime(2013, 1, 13, 4, 20)
3         ...:
4         ...: # 生成切片数据
5         ...: data_fecha.ix[date_1: date_2]
6 Out[28]:
```

| | rnd_1 | rnd_2 | rnd_3 |
|------------|-------|-------|-------|
| fecha | | | |
| 2013-01-11 | 19 | 17 | 19 |
| 2013-01-12 | 10 | 9 | 17 |
| 2013-01-13 | 15 | 3 | 10 |

如上面的例子所示, 2013年1月10号并没有被选择进去, 因为这个时间点被看作为0点0分, 比8点30分要早一些。

Python数据分析实战-Pandas 03-12
深度学习、机器学习与数据分析必须用pandas。pandas是在python直接流行的数据处理框架。可以说, 如果不会使用pandas, 就谈不上会用python做数...

Pandas 模糊查询与替换 qq_39928861的博客 17万+
主要用到的工具: Pandas & fuzzywuzzy Pandas 是基于numpy的一种工具, 专门为分析大量数据而生。它包含大量的处理数据的函数和方法, 以下为Pan...

tsing_9521: 博主用的是python 3 吗, py3里data.loc[1:5]好像不好使, 只能data.iloc[1:5],或者data.loc(index1:index2) 2年前 回复 +...

nameyx 回复: 中括号吧 2年前 回复 +...

记录一个pandas执行模糊查询sql的坑 onwingsofsong的博客 1592
查询引擎使用了presto, 在sql中使用了模糊查询。engine = create_engine(presto_url,encoding='utf-8') sql_exe = ""select id,title,tags from source_base.ta...

DataFrame查找 weixin_30319153的博客 255
一通过索引取数据 (ix/loc/iloc) loc (根据索引名称取数据, 适合单列) iloc (根据索引序号取数据, 适合多列) at (和loc类似, 只用于取单列, 性能更好) iat (...

...pandas中DataFrame的查询方法([], loc, iloc, at, i... 12-12
以上这篇浅谈pandas中DataFrame的查询方法([], loc, iloc, at, iat, ix)就是小编分享给大家的全部内容了,希望能给大家一个参考,也希望大家多多支持博客网...

pandas中DataFrame的查询方法([], loc, iloc, at, iat, ix) 10-14
pandas为我们提供了多种切片方法,而要是太不了解这些方法,就会经常容易混淆。下面举例对这些切片方法进行说明。 数据介绍 先随机生成一组数据: ...

Python 连接数据库 对DataFrame进行正则模糊查询 m0_43432838的博客 282
1.需求介绍 有些时候我们经常需要对数据库进行多次反复的数据查询, 但是这样的会对我们程序效率大打折扣, 我们完全可以先将数据库数据全部保存到p...

Pandas中at、iat函数详解 W_weiyang的博客 4137
at函数: 通过行名和列名来取数据 (取行名为a, 列名为A的值) iat 函数: 通过行号和列号来取数据 (取第1行, 第1列的值) 本文给出at、iat常见的用法, 并...

...Pandas DataFrame的查询方法(loc,iloc,at,iat,ix,... 12-18
本文标题: 详解Pandas DataFrame的查询方法(loc,iloc,at,iat,ix)的用法和区别) 本文地址: http://www.cppcms.com/jaoban/python/267356.html...

pandas DataFrame的查询方法(loc,iloc,at,iat,ix)的用法和区别) 12-26
pandas DataFrame的查询方法(loc,iloc,at,iat,ix)的用法和区别) 肯定经常会用到loc,iloc,at等函数,各个函数看起来差不多,但是还是有很多区别的,我们一起来看下吧。 首先,还是列出一个我们用的Dat...

python数据预处理_1_单条件筛选1.2_多条件筛选1.3_排除特定行2_索引操作2.2_iloc函数2.3_2.4 ix函数2.5 at函数2.6 iat函数 众所周知pan... 14-11
文章目录 1.条件筛选1.1 单条件筛选1.2 多条件筛选1.3 排除特定行2 索引操作2.2 iloc函数2.3 2.4 ix函数2.5 at函数2.6 iat函数 众所周知pan...

Pandas中DataFrame数据查询 qq_34399105的博客 949
import quandl # Data from http://www.quandl.com/SIDC/USNSPOTS_ASUnspot-Numbers-Annual # PyPi https://pypi.python.org/pypi/Quandl unspots = ...

...pandas DataFrame的查询方法(loc,iloc,at,iat,ix,... 1-5
pandas DataFrame的查询方法(loc,iloc,at,iat,ix,...) 肯定经常会用到loc,iloc,at等函数,各个函数看起来差不多,但是还是有很多区别的,我们一...

pandas中DataFrame的查询方法([], loc, iloc, at, iat, ix) 1-15
注意loc与[]的不同之处在于会把3也选择进去,使用[]是不包含的。 In [15]: data_fecha.head() Out[15]: rnd_1 rnd_2 rnd_3 ...

DataFrame 中的查询方式 (iloc, loc, iat,at) JiaShilin_的博客 311
1.loc单个标签 (label) 接受一个"标签" (label) 参数, 返回一个Series, 返回通过这个标签定位的行值。注意这里是通过标签定位, 而不是通过中括号号...

python dataframe 查找数据总结 1784
比如我们有个dataframe如下: 查找第一行 a.iloc[0] a.iloc[0] 查找a列 a['a'] 查找指定行用的值 a['a'][0] a['a'].iloc[0] a['a'].iloc[0] 说明 最好用loc和iloc, 因为当...

pandas中DataFrame的查询方法([], loc, iloc, at, iat, ix) 12-25
pandasdataframe数据提取的方法 摸梦问道的博客 4万+ DataFrame 筛选数据与loc用法 BetterWorld的专栏 2万+ ©2020 CSDN皮肤主题: 大白 设计师:CSDN...

pandas中DataFrame的查询方法([], loc, iloc, at, iat, ix,... 12-23
iat对于iloc的关系就像at对于loc的关系,是一种更快的基于索引位置的选择方法,同at一样只能访问单个元素。 In [15]: frame.iat[1,2] Out[16]: 2001.6 ix 以...

Pandas的DataFrame的过滤查询 03-24
df3中的uid代表用户ID, uid中的用户ID有重复的, 对应着不同eid(事件)下的其他列的数据。ctime为数据创建时间。现在想取出每个uid下, 最后一次C...

pandas读取DataFrame的时候df.loc[x]和普通的df[x]取值范围是不同的 遨游小林的博客 215
1 如下例: df = pd.DataFrame({'a':['1,2,3'],'b':['4,5,6]}) 如上图, 使用df.loc[2]读取时包括了右边是'闭'的。df[2] 如上图, 使用df[2]读取时包括了右...

Pandas DataFrame的loc、iloc、ix和iat/iat切片_我是尘客哥 12-19
2) loc.at,at选列只是列名,不能是position 3) iloc.at,iat选列是否是position,不能是列名 4) df[]只能进行行选择,或列选择,不能同时进行行列选择,行列选择...

Pandas中ix,iloc,iloc有什么区别? huoziliang 2万+
直接看例子: >>> data = pd.Series(np.arange(10), index=[49,48,47,46,45, 1, 2, 3, 4, 5]) >>> data 49 0 48 1 47 2 46 3 45 4 1 5 2 6 3 4 5 9 dtype: int64 ...

技术教程: AI程序员2021如何跳出舒适圈? ! CSDN学院 2万+
近日, IDC调研机构与浪潮联合发布《2020-2021 中国人工智能算力发展评估报告》。报告显示, 预计2020 年中国AI市场规模将达到 62.7亿美元, 201...

DataFrame的ix方法 1762
关于Pandas——loc、iloc、ix 函数区别和用法, 可见博客: https://blog.csdn.net/u012736685/article/details/86610946 这里只是说一下, ix方法是包含首...

data.describe(): JiaShilin 3461
data.describe() describe() 函数可以查看数据的基本情况, 包括: count 非空值数、mean 平均值、std 标准差、max 最大值、min 最小值、(25%、5...

Pandas中loc和iloc函数用法详解 (源码+实例) W_weiyang的博客 17万+
loc函数: 通过行索引 "index" 中的具体值来取数据 (如取"index"为"A"的行) iloc函数: 通过行号来取数据 (如取第二行的数据) 本文给出loc、iloc常...

pandas str.contains 列表 (list) 模糊匹配 weixin_41018467的博客 10万+
使用pandas时, str.contains()进行一次模糊匹配多个值 方法1: search = [python,'java','go'] found = [a[language].str.contains(x) for x in searchfor] 方法...

Pandas想必会有人在DataFrame中检索需要的数据, 以下是一些常用方法的汇总: 方法 说明 df[val] 从DataFrame中选取一列或者若干列。有一些特殊... 7616
有许多方式可以从一个DataFrame中检索需要的数据, 以下是一些常用方法的汇总: 方法 说明 df[val] 从DataFrame中选取一列或者若干列。有一些特殊...

©2020 CSDN 皮肤主题: 大白 设计师:CSDN官方博客 返回首页

关于我们 招贤纳士 广告服务 开发助手 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00
公安备案号11010502030143 京ICP备19004658号 京网文(2020) 1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 网络110报警服务 中国互联网举报中心 家长监护 Chrome商店下载 ©1999-2021北京创新乐知网络技术有限公司 版权与免责声明 版权申诉