用Pandas完成Excel中常见的任务（2）

2015/01/29 · 工具与框架, 框架, 系列教程 · 1 评论 · Excel, Pandas

本文由 伯乐在线 - 艾凌风 翻译，Daetalus 校稿。未经许可，禁止转载！
英文出处：pbpython。欢迎加入翻译组。

介绍

读者对于本系列第一篇文章的回应，让我感到很兴奋。感谢大家正面的反馈。我想把本系列继续下去，重点介绍其他的一些你经常使用Excel完成的任务，并且展示给你如何在pandas 中使用相同的功能。

在第一篇文章中，我着重介绍了Excel中常见的数学计算工作，以及在pandas如何完成这些工作。在本文中，我们将着重介绍一些常见的选择和筛选任务，并且介绍如何在pandas中完成同样的事情。

设置

如果您想要继续下去，您可以下载本excel文件。

导入pandas和numpy模块。

```python
import pandas as pd
import numpy as np
```

```python
df = pd.read_excel("sample-salesv3.xlsx")
```

快速浏览一下数据类型，以确保所以事情都能如预期一样运行。

```python
df.dtypes
```

```
account number int64
name object
sku object
quantity int64
unit price float64
ext price float64
date object
dtype: object
```

你会注意到，我们的date列，显示的是一个通用对象。我们准备把它转换为日期对象，来简化将来会用到的一些选择操作。

```python
df['date'] = pd.to_datetime(df['date'])
df.head()
```

	account number	name	sku	quantity	unit price	ext price	date
0	740150	Barton LLC	B1-20000	39	86.69	3380.91	2014-01-01 07:21:51
1	714466	Trantow-Barrows	S2-77896	-1	63.16	-63.16	2014-01-01 10:00:47
2	218895	Kulas Inc	B1-69924	23	90.70	2086.10	2014-01-01 13:24:58
3	307599	Kassulke, Ondricka and Metz	S1-65481	41	21.05	863.05	2014-01-01 15:05:22
4	412290	Jerde-Hilpert	S2-34077	6	83.21	499.26	2014-01-01 23:26:55

```python
df.dtypes
```

```
account number int64
name object
sku object
quantity int64
unit price float64
ext price float64
date datetime64[ns]
```

现在，data变成了一个datetime类型的对象，这对于将来的操作是很有用的。

筛选数据

我认为在Excel中最方便的功能是筛选。我想几乎每一次有人拿到一个任意大小的Excel文件，当他们想要筛选数据的时候，都会使用这个功能。

如图，对本数据集使用该功能：

同Excel中的筛选功能一样，你可以使用pandas来筛选和选择某个特定数据的子集。

比方说，如果我们仅仅想查看一个特定的账号，我们可以简单是在Excel中完成，或是使用pandas完成操作。

下面是Excel的筛选解决方案：

在pandas中执行相关操作比Excel中更加直观。注意，我将会使用`head` 函数来显示前面几个结果。这仅仅是为了让本文保持简短。

```python
df[df["account number"]==307599].head()
```

你还可以以数值为基准来进行筛选。我就不再举任何Excel的例子了。我相信你能明白。

```python
df[df["quantity"] &gt; 22].head()
```

	account number	name	sku	quantity	unit price	ext price	date
0	740150	Barton LLC	B1-20000	39	86.69	3380.91	2014-01-01 07:21:51
2	218895	Kulas Inc	B1-69924	23	90.70	2086.10	2014-01-01 13:24:58
3	307599	Kassulke, Ondricka and Metz	S1-65481	41	21.05	863.05	2014-01-01 15:05:22
14	737550	Fritsch, Russel and Anderson	B1-53102	23	71.56	1645.88	2014-01-04 08:57:48
15	239344	Stokes LLC	S1-06532	34	71.51	2431.34	2014-01-04 11:34:58

如果我们想要更多复杂的筛选，我们可以可以使用`map` 来以多重标准进行筛选。在这个例子中，从B1中查找以“sku”中起始的项目。

	account number	name	sku	quantity	unit price	ext price	date
0	740150	Barton LLC	B1-20000	39	86.69	3380.91	2014-01-01 07:21:51
2	218895	Kulas Inc	B1-69924	23	90.70	2086.10	2014-01-01 13:24:58
6	218895	Kulas Inc	B1-65551	2	31.10	62.20	2014-01-02 10:57:23
14	737550	Fritsch, Russel and Anderson	B1-53102	23	71.56	1645.88	2014-01-04 08:57:48
17	239344	Stokes LLC	B1-50809	14	16.23	227.22	2014-01-04 22:14:32

把两个或更多的语句连接起来很简单，用&就可以。

Python

```
df[df["sku"].map(lambda x: x.startswith('B1')) &amp;amp; (df["quantity"] &amp;gt; 22)].head()
```

	account number	name	sku	quantity	unit price	ext price	date
0	740150	Barton LLC	B1-20000	39	86.69	3380.91	2014-01-01 07:21:51
2	218895	Kulas Inc	B1-69924	23	90.70	2086.10	2014-01-01 13:24:58
14	737550	Fritsch, Russel and Anderson	B1-53102	23	71.56	1645.88	2014-01-04 08:57:48
26	737550	Fritsch, Russel and Anderson	B1-53636	42	42.06	1766.52	2014-01-08 00:02:11
31	714466	Trantow-Barrows	B1-33087	32	19.56	625.92	2014-01-09 10:16:32

pandas支持的另外一个很有用的函数是`isin`。它使得我们可以定义一个列表，里面包含我们所希望查找的值

在这个例子中，我们查找包含两个特定account number值的全部项目。

Python

```
df[df[&amp;quot;account number&amp;quot;].isin([714466,218895])].head()
```

	account number	name	sku	quantity	unit price	ext price	date
1	714466	Trantow-Barrows	S2-77896	-1	63.16	-63.16	2014-01-01 10:00:47

			69924				13:24:58
5	714466	Trantow-Barrows	S2-77896	17	87.63	1489.71	2014-01-02 10:07:15
6	218895	Kulas Inc	B1-65551	2	31.10	62.20	2014-01-02 10:57:23
8	714466	Trantow-Barrows	S1-50961	22	84.09	1849.98	2014-01-03 11:29:02

pandas支持的另外一个函数叫做`query`，它使得我们可以有效的再数据集中选择数据。使用它需要安装numexpr ，所以请确保你在进行下面步骤前已经进行了安装。

如果你想要通过名字来得到一个消费者列表，你可以使用query来完成，和前面展示的python语法类似。

Python

```
df.query('name == ["Kulas Inc","Barton LLC"]').head()
```

	account number	name	sku	quantity	unit price	ext price	date
0	740150	Barton LLC	B1-20000	39	86.69	3380.91	2014-01-01 07:21:51
2	218895	Kulas Inc	B1-69924	23	90.70	2086.10	2014-01-01 13:24:58
6	218895	Kulas Inc	B1-65551	2	31.10	62.20	2014-01-02 10:57:23
33	218895	Kulas Inc	S1-06532	3	22.36	67.08	2014-01-09 23:58:27
36	218895	Kulas Inc	S2-34077	16	73.04	1168.64	2014-01-10 12:07:30

这里只是做个简单的示例，query函数能做到的还不止这些。我在此展示这些函数的用法，以便当你有需要的时候，会意识到可以用它。

处理日期

使用pandas，你可以对日期进行更加复杂的筛选。在我们处理日期前，我建议你把日期栏进行一个排序，以便返回的结果如你所愿。

Python

```
df = df.sort('date')
df.head()
```

	account number	name	sku	quantity	unit price	ext price	date

	number				price	price	
0	740150	Barton LLC	B1-20000	39	86.69	3380.91	2014-01-01 07:21:51
1	714466	Trantow-Barrows	S2-77896	-1	63.16	-63.16	2014-01-01 10:00:47
2	218895	Kulas Inc	B1-69924	23	90.70	2086.10	2014-01-01 13:24:58
3	307599	Kassulke, Ondricka and Metz	S1-65481	41	21.05	863.05	2014-01-01 15:05:22
4	412290	Jerde-Hilpert	S2-34077	6	83.21	499.26	2014-01-01 23:26:55

在操作日期前，为您展示python的筛选语法。

Python

```
1 df[df['date'] &gt;='20140905'].head()
```

	account number	name	sku	quantity	unit price	ext price	date
1042	163416	Purdy-Kunde	B1-38851	41	98.69	4046.29	2014-09-05 01:52:32
1043	714466	Trantow-Barrows	S1-30248	1	37.16	37.16	2014-09-05 06:17:19
1044	729833	Koepp Ltd	S1-65481	48	16.04	769.92	2014-09-05 08:54:41
1045	729833	Koepp Ltd	S2-11481	6	26.50	159.00	2014-09-05 16:33:15
1046	737550	Fritsch, Russel and Anderson	B1-33364	4	76.44	305.76	2014-09-06 08:59:08

pandas的一个特别棒的特性是它能够理解日期，所以它允许我们进行部分筛选。如果我只想要查看最近几个月的日期数据，我可以这样做。

Python

```
1 df[df['date'] &gt;='2014-03'].head()
```

	account number	name	sku	quantity	unit price	ext price	date
242	163416	Purdy-Kunde	S1-30248	19	65.03	1235.57	2014-03-01 16:07:40
243	527099	Sanford and Sons	S2-82423	3	76.21	228.63	2014-03-01 17:18:01
244	527099	Sanford and Sons	B1-50809	8	70.78	566.24	2014-03-01 18:53:09
		Fritsch, Russel and	B1-				2014-03-01

246	688981	Keeling LLC	B1-86481	-1	97.16	-97.16	2014-03-02 01:46:44

当然，你可以把筛选标准链接起来。

Python

```
df[(df['date'] >='20140701') & (df['date'] <= '20140715')].head()
```

	account number	name	sku	quantity	unit price	ext price	date
778	737550	Fritsch, Russel and Anderson	S1-65481	35	70.51	2467.85	2014-07-01 00:21:58
779	218895	Kulas Inc	S1-30248	9	16.56	149.04	2014-07-01 00:52:38

	account number	name	sku	quantity	unit price	ext price	date
780	163416	Purdy-Kunde	S2-82423	44	68.27	3003.88	2014-07-01 08:15:52
781	672390	Kuhn-Gusikowski	B1-04202	48	99.39	4770.72	2014-07-01 11:12:13
782	642753	Pollich LLC	S2-23246	1	51.29	51.29	2014-07-02 04:02:39

由于pandas可以理解日期列，所以可以将日期值设为不同的格式，都会得到正确的结果。

	account number	name	sku	quantity	unit price	ext price	date
1168	307599	Kassulke, Ondricka and Metz	S2-23246	6	88.90	533.40	2014-10-08 06:19:50
1169	424914	White-Trantow	S2-10342	25	58.54	1463.50	2014-10-08 07:31:40
1170	163416	Purdy-Kunde	S1-27722	22	34.41	757.02	2014-10-08 09:01:18
1171	163416	Purdy-Kunde	B1-33087	7	79.29	555.03	2014-10-08 15:39:13
1172	672390	Kuhn-Gusikowski	B1-38851	30	94.64	2839.20	2014-10-09 00:22:33

Python

```
df[df['date'] &gt;= '10-10-2014'].head()
```

	account	name	sku	quantity	unit	ext	date

1174	257198	Cronin, Oberbrunner and Spencer	S2-34077	13	12.24	159.12	2014-10-10 02:59:06
1175	740150	Barton LLC	S1-65481	28	53.00	1484.00	2014-10-10 15:08:53
1176	146832	Kiehn-Spinka	S1-27722	15	64.39	965.85	2014-10-10 18:24:01
1177	257198	Cronin, Oberbrunner and Spencer	S2-16558	3	35.34	106.02	2014-10-11 01:48:13
1178	737550	Fritsch, Russel and Anderson	B1-53636	10	56.95	569.50	2014-10-11 10:25:53

当操作时间序列数据时，如果你把数据进行转化，以日期作为索引，我们可以做一些变相的筛选。

使用`set_index`来设置新的索引。

Python

```
df2 = df.set_index(['date'])
df2.head()
```

	account number	name	sku	quantity	unit price	ext price
date						
2014-01-01 07:21:51	740150	Barton LLC	B1-20000	39	86.69	3380.91
2014-01-01 10:00:47	714466	Trantow-Barrows	S2-77896	-1	63.16	-63.16
2014-01-01 13:24:58	218895	Kulas Inc	B1-69924	23	90.70	2086.10
2014-01-01 15:05:22	307599	Kassulke, Ondricka and Metz	S1-65481	41	21.05	863.05
2014-01-01 23:26:55	412290	Jerde-Hilpert	S2-34077	6	83.21	499.26

你可以通过切分数据来获取一段区间。

Python

```
df2["20140101":"20140201"].head()
```

	account number	name	sku	quantity	unit price	ext price
date						
2014-01-01 07:21:51	740150	Barton LLC	B1-20000	39	86.69	3380.91
2014-01-01 10:00:47	714466	Trantow-Barrows	S2-77896	-1	63.16	-63.16

13:24:58	218895	Kulas Inc	69924	23	90.70	2086.10
2014-01-01 15:05:22	307599	Kassulke, Ondricka and Metz	S1-65481	41	21.05	863.05
2014-01-01 23:26:55	412290	Jerde-Hilpert	S2-34077	6	83.21	499.26

再一次的，我们可以使用不同的日期表示方法来避免模棱两可的日期命名惯例。

Python

```
df2["2014-Jan-1":"2014-Feb-1"].head()
```

	account number	name	sku	quantity	unit price	ext price
date						
2014-01-01 07:21:51	740150	Barton LLC	B1-20000	39	86.69	3380.91
2014-01-01 10:00:47	714466	Trantow-Barrows	S2-77896	-1	63.16	-63.16
2014-01-01 13:24:58	218895	Kulas Inc	B1-69924	23	90.70	2086.10
2014-01-01 15:05:22	307599	Kassulke, Ondricka and Metz	S1-65481	41	21.05	863.05
2014-01-01 23:26:55	412290	Jerde-Hilpert	S2-34077	6	83.21	499.26

Python

```
df2["2014-Jan-1":"2014-Feb-1"].tail()
```

	account number	name	sku	quantity	unit price	ext price
date						
2014-01-31 22:51:18	383080	Will LLC	B1-05914	43	80.17	3447.31
2014-02-01 09:04:59	383080	Will LLC	B1-20000	7	33.69	235.83
2014-02-01 11:51:46	412290	Jerde-Hilpert	S1-27722	11	21.12	232.32
2014-02-01 17:24:32	412290	Jerde-Hilpert	B1-86481	3	35.99	107.97
2014-02-01 19:56:48	412290	Jerde-Hilpert	B1-20000	23	78.90	1814.70

```
df2["2014"].head()
```

	account number	name	sku	quantity	unit price	ext price
date						
2014-01-01 07:21:51	740150	Barton LLC	B1-20000	39	86.69	3380.91
2014-01-01 10:00:47	714466	Trantow-Barrows	S2-77896	-1	63.16	-63.16
2014-01-01 13:24:58	218895	Kulas Inc	B1-69924	23	90.70	2086.10
2014-01-01 15:05:22	307599	Kassulke, Ondricka and Metz	S1-65481	41	21.05	863.05
2014-01-01 23:26:55	412290	Jerde-Hilpert	S2-34077	6	83.21	499.26

Python

```
df2["2014-Dec"].head()
```

	account number	name	sku	quantity	unit price	ext price
date						
2014-12-01 20:15:34	714466	Trantow-Barrows	S1-82801	3	77.97	233.91
2014-12-02 20:00:04	146832	Kiehn-Spinka	S2-23246	37	57.81	2138.97
2014-12-03 04:43:53	218895	Kulas Inc	S2-77896	30	77.44	2323.20
2014-12-03 06:05:43	141962	Herman LLC	B1-53102	20	26.12	522.40
2014-12-03 14:17:34	642753	Pollich LLC	B1-53636	19	71.21	1352.99

正如你所见到的那样，在进行基于日期的排序或者筛选时，可以有很多选择。

额外的字符串方法

Pandas同样已经支持了矢量字符串方法。

如果我们想识别出sku栏中包含某一特定值的全部值。我们可以使用`str.contains`。在这个例子中，我们已知sku总是以一种相同的方式表示，所以B1仅会出现在sku的前面。你需要理解你的数据来保证你能够得到你想要的结果。

Python

	account number	name	sku	quantity	unit price	ext price	date
0	740150	Barton LLC	B1-20000	39	86.69	3380.91	2014-01-01 07:21:51
2	218895	Kulas Inc	B1-69924	23	90.70	2086.10	2014-01-01 13:24:58
6	218895	Kulas Inc	B1-65551	2	31.10	62.20	2014-01-02 10:57:23
14	737550	Fritsch, Russel and Anderson	B1-53102	23	71.56	1645.88	2014-01-04 08:57:48
17	239344	Stokes LLC	B1-50809	14	16.23	227.22	2014-01-04 22:14:32

我们可以把查询连接起来并且使用排序来控制数据的顺序。

```
Python
1 df[(df['sku'].str.contains('B1-531')) &amp;amp; (df['quantity']&amp;gt;40)].sort(columns=['qua
```

	account number	name	sku	quantity	unit price	ext price	date
684	642753	Pollich LLC	B1-53102	46	26.07	1199.22	2014-06-08 19:33:33
792	688981	Keeling LLC	B1-53102	45	41.19	1853.55	2014-07-04 21:42:22
176	383080	Will LLC	B1-53102	45	89.22	4014.90	2014-02-11 04:14:09
1213	604255	Halvorson, Crona and Champlin	B1-53102	41	55.05	2257.05	2014-10-18 19:27:01
1215	307599	Kassulke, Ondricka and Metz	B1-53102	41	93.70	3841.70	2014-10-18 23:25:10
1128	714466	Trantow-Barrows	B1-53102	41	55.68	2282.88	2014-09-27 10:42:48
1001	424914	White-Trantow	B1-53102	41	81.25	3331.25	2014-08-26 11:44:30

彩蛋任务

在Excel中，我发现我自己经常会尝试从一个冗长的列表中，得到一个包含不重复项的小列表。在Excel中这件事情需要分几步来完成，但是在Pandas中却非常简单。有一种方式是使用Excel中提供的高级筛选工具来完成。

```python
df["name"].unique()
```

```
array([u'Barton LLC', u'Trantow-Barrows', u'Kulas Inc',
u'Kassulke, Ondricka and Metz', u'Jerde-Hilpert', u'Koepp Ltd',
u'Fritsch, Russel and Anderson', u'Kiehn-Spinka', u'Keeling LLC',
u'Frami, Hills and Schmidt', u'Stokes LLC', u'Kuhn-Gusikowski',
u'Herman LLC', u'White-Trantow', u'Sanford and Sons',
u'Pollich LLC', u'Will LLC', u'Cronin, Oberbrunner and Spencer',
u'Halvorson, Crona and Champlin', u'Purdy-Kunde'], dtype=object)
If we wanted to include the account number, we could use drop_duplicates .
```

如果我们想要包含账户号，我们可以使用drop_duplicates。

```python
df.drop_duplicates(subset=["account number","name"]).head()
```

	account number	name	sku	quantity	unit price	ext price	date
0	740150	Barton LLC	B1-20000	39	86.69	3380.91	2014-01-01 07:21:51
1	714466	Trantow-Barrows	S2-77896	-1	63.16	-63.16	2014-01-01 10:00:47
2	218895	Kulas Inc	B1-69924	23	90.70	2086.10	2014-01-01 13:24:58
3	307599	Kassulke, Ondricka and Metz	S1-65481	41	21.05	863.05	2014-01-01 15:05:22
4	412290	Jerde-Hilpert	S2-34077	6	83.21	499.26	2014-01-01 23:26:55

很显然我们放入了的数据超过了我们的需要，得到了一些无用的信息，因此，使用ix 来仅仅选择第一第二列。

```python
df.drop_duplicates(subset=["account number","name"]).ix[:,[0,1]]
```

	account number	name
0	740150	Barton LLC
1	714466	Trantow-Barrows
2	218895	Kulas Inc
3	307599	Kassulke, Ondricka and Metz
4	412290	Jerde-Hilpert
7	729833	Koepp Ltd
9	737550	Fritsch, Russel and Anderson

10	146832	Kiehn-Spinka
11	688981	Keeling LLC
12	786968	Frami, Hills and Schmidt
15	239344	Stokes LLC
16	672390	Kuhn-Gusikowski
18	141962	Herman LLC
20	424914	White-Trantow
21	527099	Sanford and Sons
30	642753	Pollich LLC
37	383080	Will LLC
51	257198	Cronin, Oberbrunner and Spencer
67	604255	Halvorson, Crona and Champlin
106	163416	Purdy-Kunde

我认为这个记住这个单独的命令比记忆Excel的各步操作更容易。

如果你想要查看我的笔记 请随意下载。

结论

在我发表了我的第一篇文章之后，Dave Proffer在Twitter上转发了我的文章并评论到“打破你#excel沉迷的一些好技巧”。我觉得这句话非常准确，它描述了在我们的生活中使用Excel是有多么的频繁。大多数的人只管伸手去用却从来没有意识到它的局限性。我希望这个系列的文章可以帮助大家认识到我们还有其他的替代工具，Python+Pandas是一个极其强大的组合。

> **打赏支持我翻译更多好文章，谢谢！**
>
> ¥ 打赏译者

关于作者：艾凌风

翻译组的勤务员（联系此人请微博私信或hanxiaomax@qq.com。试译申请保证回复，如长时间没收到请邮催我

个人主页 · 我的文章 · 95 ·