用Pandas完成Excel中常见的任务

2015/01/04 · 工具与框架, 框架, 系列教程 · 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.com。欢迎加入翻译组。

引言

本文的目的，是向您展示如何使用pandas 来执行一些常见的Excel任务。有些例子比较琐碎，但我觉得展示这些简单的东西与那些你可以在其他地方找到的复杂功能同等重要。作为额外的福利，我将会进行一些模糊字符串匹配，以此来展示一些小花样，以及展示pandas是如何利用完整的Python模块系统去做一些在Python中是简单，但在Excel中却很复杂的事情的。

有道理吧？让我们开始吧。

为某行添加求和项

我要介绍的第一项任务是把某几列相加然后添加一个总和栏。

首先我们将excel 数据 导入到pandas数据框架中。

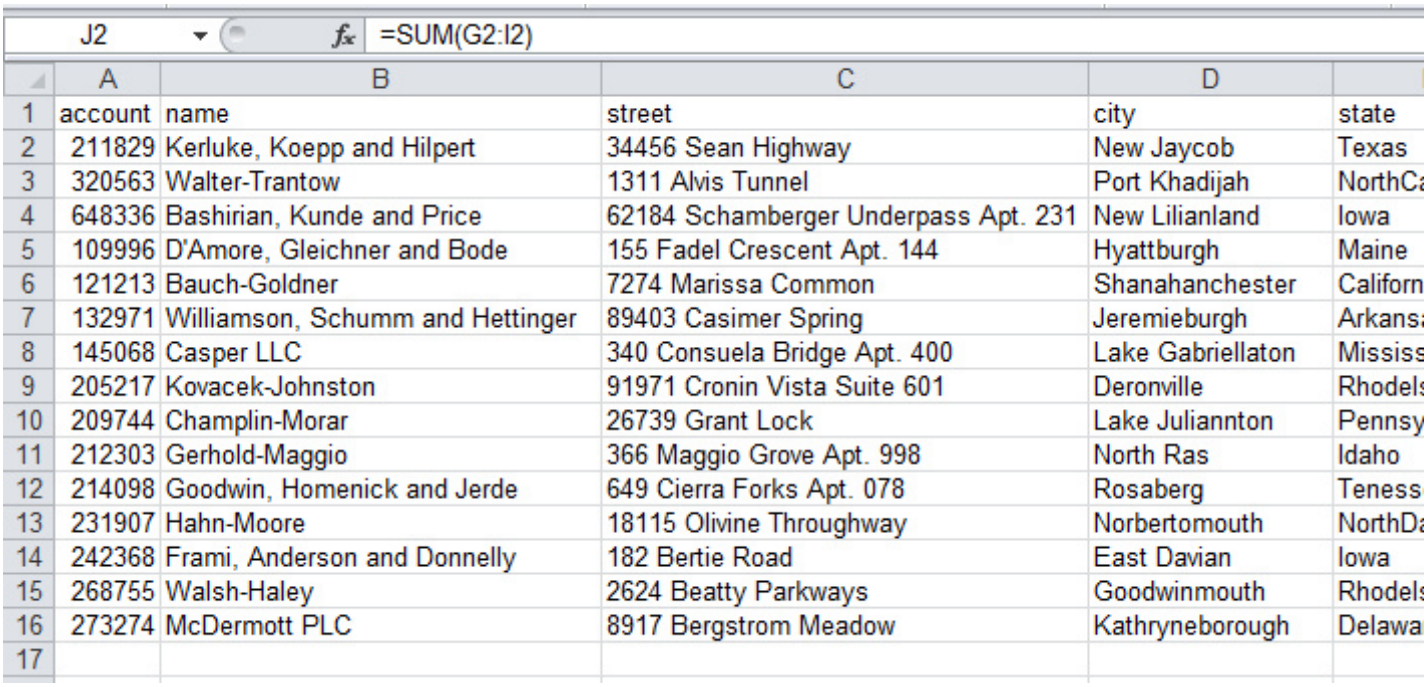
Python

```
import pandas as pd
import numpy as np
df = pd.read_excel("excel-comp-data.xlsx")
df.head()
```

	account	name	street	city	state
0	211829	Kerluke, Koepp and Hilpert	34456 Sean Highway	New Jaycob	Texas
1	320563	Walter-Trantow	1311 Alvis Tunnel	Port Khadijah	NorthCarol
2	648336	Bashirian, Kunde and Price	62184 Schamberger Underpass Apt. 231	New Lilianland	Iowa
3	109996	D'Amore, Gleichner and Bode	155 Fadel Crescent Apt. 144	Hyattburgh	Maine
4	121213	Bauch-Goldner	7274 Marissa Common	Shanahanchester	California

我们想要添加一个总和栏来显示Jan、Feb和Mar三个月的销售总额。

在Excel和pandas中这都是简单直接的。对于Excel，我在J列中添加了公式`sum(G2:I2)`。在Excel中看上去是这样的：



J2 =SUM(G2:I2)

	A	B	C	D	
1	account	name	street	city	state
2	211829	Kerluke, Koepp and Hilpert	34456 Sean Highway	New Jaycob	Texas
3	320563	Walter-Trantow	1311 Alvis Tunnel	Port Khadijah	NorthCa
4	648336	Bashirian, Kunde and Price	62184 Schamberger Underpass Apt. 231	New Lilianland	Iowa
5	109996	D'Amore, Gleichner and Bode	155 Fadel Crescent Apt. 144	Hyattburgh	Maine
6	121213	Bauch-Goldner	7274 Marissa Common	Shanahanchester	Californ
7	132971	Williamson, Schumm and Hettinger	89403 Casimer Spring	Jeremieburgh	Arkansa
8	145068	Casper LLC	340 Consuela Bridge Apt. 400	Lake Gabriellaton	Mississ
9	205217	Kovacek-Johnston	91971 Cronin Vista Suite 601	Deronville	Rhodels
10	209744	Champlin-Morar	26739 Grant Lock	Lake Juliannton	Pennsy
11	212303	Gerhold-Maggio	366 Maggio Grove Apt. 998	North Ras	Idaho
12	214098	Goodwin, Homenick and Jerde	649 Cierra Forks Apt. 078	Rosaberg	Teness
13	231907	Hahn-Moore	18115 Olivine Throughway	Norbertomouth	NorthDa
14	242368	Frami, Anderson and Donnelly	182 Bertie Road	East Davian	Iowa
15	268755	Walsh-Haley	2624 Beatty Parkways	Goodwinmouth	Rhodels
16	273274	McDermott PLC	8917 Bergstrom Meadow	Kathryneborough	Delawa
17					

下面，我们是这样在pandas中操作的：

Python

```
; html-script: false ]df["total"] = df["Jan"] + df["Feb"] + df["Mar"]
df.head()
```

	account	name	street	city	state	cod
0	211829	Kerluke, Koepp and Hilpert	34456 Sean Highway	New Jaycob	Texas	287
1	320563	Walter-Trantow	1311 Alvis Tunnel	Port Khadijah	NorthCarolina	383
2	648336	Bashirian, Kunde and Price	62184 Schamberger Underpass Apt. 231	New Lilianland	Iowa	765
3	109996	D'Amore, Gleichner and Bode	155 Fadel Crescent Apt. 144	Hyattburgh	Maine	460
4	121213	Bauch-Goldner	7274 Marissa Common	Shanahanchester	California	496

接下来，让我们对各列计算一些汇总信息以及其他值。如下Excel表所示，我们要做这些工作：

G17 =SUM(G2:G16)

	A	B	C	D	
1	account	name	street	city	state
2	211829	Kerluke, Koepp and Hilpert	34456 Sean Highway	New Jaycob	Texas
3	320563	Walter-Trantow	1311 Alvis Tunnel	Port Khadijah	NorthCa
4	648336	Bashirian, Kunde and Price	62184 Schamberger Underpass Apt. 231	New Lilianland	Iowa
5	109996	D'Amore, Gleichner and Bode	155 Fadel Crescent Apt. 144	Hyattburgh	Maine
6	121213	Bauch-Goldner	7274 Marissa Common	Shanahanchester	Californ
7	132971	Williamson, Schumm and Hettinger	89403 Casimer Spring	Jeremieburgh	Arkansa
8	145068	Casper LLC	340 Consuela Bridge Apt. 400	Lake Gabriellaton	Mississ
9	205217	Kovacek-Johnston	91971 Cronin Vista Suite 601	Deronville	Rhodels
10	209744	Champlin-Morar	26739 Grant Lock	Lake Juliannton	Pennsy
11	212303	Gerhold-Maggio	366 Maggio Grove Apt. 998	North Ras	Idaho
12	214098	Goodwin, Homenick and Jerde	649 Cierra Forks Apt. 078	Rosaberg	Teness
13	231907	Hahn-Moore	18115 Olivine Throughway	Norbertomouth	NorthDa
14	242368	Frami, Anderson and Donnelly	182 Bertie Road	East Davian	Iowa
15	268755	Walsh-Haley	2624 Beatty Parkways	Goodwinmouth	Rhodels
16	273274	McDermott PLC	8917 Bergstrom Meadow	Kathryneborough	Delawa
17					
18					

如你所见，我们在表示月份的列的第17行添加了`SUM(G2:G16)`，来取得每月的总和。

进行在pandas中进行列级别的分析很简单。下面是一些例子：

```
Python
1 ; html-script: false ]df["Jan"].sum(), df["Jan"].mean(),df["Jan"].min(),df["Jan"].max(
```

```
; html-script: false ](1462000, 97466.66666666672, 10000, 162000)
```

现在我们要把每月的总和相加得到它们的和。这里pandas和Excel有点不同。在Excel的单元格里把每个月的总和相加很简单。由于pandas需要维护整个DataFrame的完整性，所以需要一些额外的步骤。

首先，建立所有列的总和栏

Python

```
; html-script: false ]sum_row=df[["Jan","Feb","Mar","total"]].sum()
sum_row
```

Python

```
; html-script: false ]Jan      1462000
Feb      1507000
Mar       717000
total    3686000
dtype: int64
```

这很符合直觉，不过如果你希望将总和值显示为表格中的单独一行，你还需要做一些微调。

我们需要把数据进行变换，把这一系列数字转换为DataFrame，这样才能更加容易的把它合并进已经存在的数据中。`T` 函数可以让我们把按行排列的数据变换为按列排列。

Python

```
; html-script: false ]df_sum=pd.DataFrame(data=sum_row).T
df_sum
```

	Jan	Feb	Mar	total
0	1462000	1507000	717000	3686000

在计算总和之前我们要做的最后一件事情是添加丢失的列。我们使用`reindex`来帮助我们完成。技巧是添加全部的列然后让pandas去添加所有缺失的数据。

Python

```
; html-script: false ]df_sum=df_sum.reindex(columns=df.columns)
df_sum
```

	account	name	street	city	state	postal-code	Jan	Feb	Mar	total
0	NaN	NaN	NaN	NaN	NaN	NaN	1462000	1507000	717000	3686000

现在我们已经有了一个格式良好的DataFrame，我们可以使用`append`来把它加入到已有的内容中。

Python

```
; html-script: false ]df_final=df.append(df_sum,ignore_index=True)
df_final.tail()
```

	account	name	street	city	state	code
11	231907	Hahn-Moore	18115 Olivine Throughway	Norbertomouth	NorthDakota	3141
12	242368	Frami, Anderson and Donnelly	182 Bertie Road	East Davian	Iowa	7268
13	268755	Walsh-Haley	2624 Beatty Parkways	Goodwinmouth	RhodeIsland	3191
14	273274	McDermott PLC	8917 Bergstrom Meadow	Kathryneborough	Delaware	2793
15	NaN	NaN	NaN	NaN	NaN	NaN

额外的数据变换

另外一个例子，让我们尝试给数据集添加状态的缩写。

对于Excel，最简单的方式是添加一个新的列，对州名使用vlookup函数并填充缩写栏。

我进行了这样的操作，下面是其结果的截图：

G2 =VLOOKUP(E2,Sheet2!A:B,2,FALSE)

	A	B	C	D	
1	account	name	street	city	state
2	211829	Kerluke, Koepp and Hilpert	34456 Sean Highway	New Jaycob	Texas
3	320563	Walter-Trantow	1311 Alvis Tunnel	Port Khadijah	NorthCa
4	648336	Bashirian, Kunde and Price	62184 Schamberger Underpass Apt. 231	New Lilianland	Iowa
5	109996	D'Amore, Gleichner and Bode	155 Fadel Crescent Apt. 144	Hyattburgh	Maine
6	121213	Bauch-Goldner	7274 Marissa Common	Shanahanchester	Californ
7	132971	Williamson, Schumm and Hettinger	89403 Casimer Spring	Jeremieburgh	Arkansa
8	145068	Casper LLC	340 Consuela Bridge Apt. 400	Lake Gabriellaton	Mississ
9	205217	Kovacek-Johnston	91971 Cronin Vista Suite 601	Deronville	Rhodels
10	209744	Champlin-Morar	26739 Grant Lock	Lake Juliannton	Pennsy
11	212303	Gerhold-Maggio	366 Maggio Grove Apt. 998	North Ras	Idaho
12	214098	Goodwin, Homenick and Jerde	649 Cierra Forks Apt. 078	Rosaberg	Teness
13	231907	Hahn-Moore	18115 Olivine Throughway	Norbertomouth	NorthDa
14	242368	Frami, Anderson and Donnelly	182 Bertie Road	East Davian	Iowa
15	268755	Walsh-Haley	2624 Beatty Parkways	Goodwinmouth	Rhodels
16	273274	McDermott PLC	8917 Bergstrom Meadow	Kathryneborough	Delawa
17					
18					

你可以注意到，在进行了vlookup后，有一些数值并没有被正确的取得。这是因为我们拼错了一些州的名字。在Excel中处理这一问题是一个巨大的挑战（对于大型数据集而言）

幸运的是，使用pandas我们可以利用强大的python生态系统。考虑如何解决这类麻烦的数据问题，我考虑进行一些模糊文本匹配来决定正确的值。

题。首先要确保你安装了他。

我们需要的另外一段代码是州名与其缩写的映射表。而不是亲自去输入它们，谷歌一下你就能找到这段代码code。

首先导入合适的fuzzywuzzy函数并且定义我们的州名映射表。

```
; html-script: false ]from fuzzywuzzy import fuzz
from fuzzywuzzy import process
state_to_code = {"VERMONT": "VT", "GEORGIA": "GA", "IOWA": "IA", "Armed Forces Pacific
                 "KANSAS": "KS", "FLORIDA": "FL", "AMERICAN SAMOA": "AS", "NORTH CAROL
                 "NEW YORK": "NY", "CALIFORNIA": "CA", "ALABAMA": "AL", "IDAHO": "ID",
                 "Armed Forces Americas": "AA", "DELAWARE": "DE", "ALASKA": "AK", "ILL
                 "Armed Forces Africa": "AE", "SOUTH DAKOTA": "SD", "CONNECTICUT": "CT
                 "PUERTO RICO": "PR", "Armed Forces Canada": "AE", "NEW HAMPSHIRE": "N
                 "MISSISSIPPI": "MS", "TENNESSEE": "TN", "PALAU": "PW", "COLORADO": "C
                 "NEW JERSEY": "NJ", "UTAH": "UT", "MICHIGAN": "MI", "WEST VIRGINIA":
                 "MINNESOTA": "MN", "OREGON": "OR", "VIRGINIA": "VA", "VIRGIN ISLANDS'
                 "WYOMING": "WY", "OHIO": "OH", "SOUTH CAROLINA": "SC", "INDIANA": "IN
                 "NORTHERN MARIANA ISLANDS": "MP", "NEBRASKA": "NE", "ARIZONA": "AZ",
                 "Armed Forces Europe": "AE", "PENNSYLVANIA": "PA", "OKLAHOMA": "OK",
                 "DISTRICT OF COLUMBIA": "DC", "ARKANSAS": "AR", "MISSOURI": "MO", "TE
```

这里有些介绍模糊文本匹配函数如何工作的例子。

```
; html-script: false ]process.extractOne("Minnesotta",choices=state_to_code.keys())
```

```
; html-script: false ]('MINNESOTA', 95)
```

```
; html-script: false ]process.extractOne("AlaBAMMazzz",choices=state_to_code.keys(),sc
```

现在我知道它是如何工作的了，我们创建自己的函数来接受州名这一列的数据然后把他转换为一个有效的缩写。这里我们使用score_cutoff的值为80。你可以做一些调整，看看哪个值对你的数据来说比较好。你会注意到，返回值要么是一个有效的缩写，要么是一个np. nan 所以域中会有一些有效的值。

```
; html-script: false ]def convert_state(row):
    abbrev = process.extractOne(row["state"],choices=state_to_code.keys(),score_cutoff
    if abbrev:
        return state_to_code[abbrev[0]]
    return np.nan
```

把这列添加到我们想要填充的单元格，然后用NaN填充它

```
; html-script: false ]df_final.insert(6, "abbrev", np.nan)
df_final.head()
```

	account	name	street	city	state	code
11	231907	Hahn-Moore	18115 Olivine Throughway	Norbertomouth	NorthDakota	3141
12	242368	Frami, Anderson and Donnelly	182 Bertie Road	East Davian	Iowa	7268
13	268755	Walsh-Haley	2624 Beatty Parkways	Goodwinmouth	RhodeIsland	3191
14	273274	McDermott PLC	8917 Bergstrom Meadow	Kathryneborough	Delaware	2793
15	NaN	NaN	NaN	NaN	NaN	NaN

我们使用apply 来把缩写添加到合适的列中。

Python

```
; html-script: false ]df_final['abbrev'] = df_final.apply(convert_state, axis=1)
df_final.tail()
```

	account	name	street	city	state	postal-code	abbre
11	231907	Hahn-Moore	18115 Olivine Throughway	Norbertomouth	NorthDakota	31415	ND
12	242368	Frami, Anderson and Donnelly	182 Bertie Road	East Davian	Iowa	72686	IA
13	268755	Walsh-Haley	2624 Beatty Parkways	Goodwinmouth	RhodeIsland	31919	RI
14	273274	McDermott PLC	8917 Bergstrom Meadow	Kathryneborough	Delaware	27933	DE
15	NaN	NaN	NaN	NaN	NaN	NaN	NaN

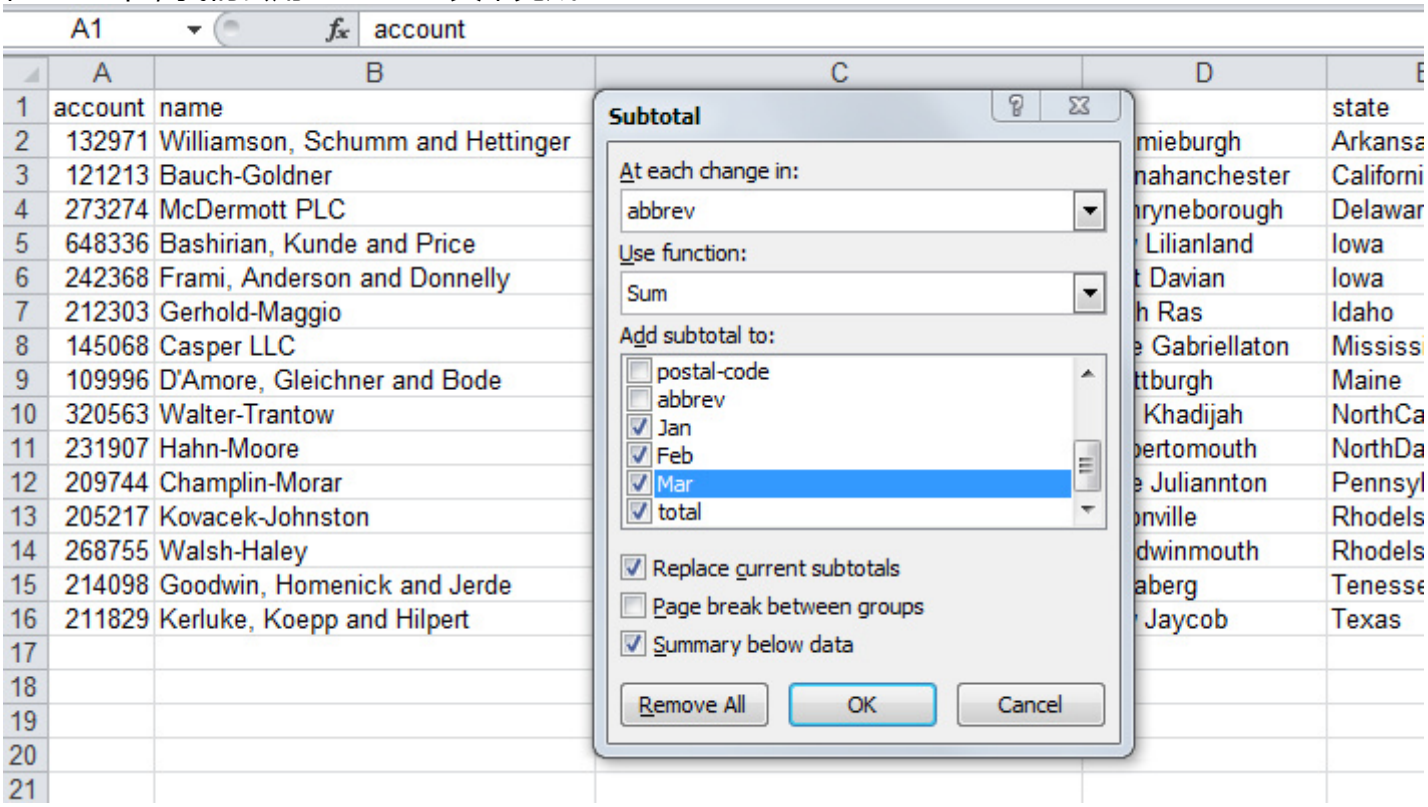
我觉的这很酷。我们已经开发出了一个非常简单的流程来智能的清理数据。显然，当你只有15行左

J 。

分类汇总

在本文的最后一节中，让我们按州来做一些分类汇总（subtotal）。

在Excel中，我们会用`subtotal` 工具来完成。



输出如下：

	abbrev	Jan	Feb	Mar	total
	AR Total	$ 150,000	$ 120,000	$ 35,000	$ 305,000
	CA Total	$ 162,000	$ 120,000	$ 35,000	$ 317,000
	DE Total	$ 150,000	$ 120,000	$ 70,000	$ 340,000
	IA Total	$ 253,000	$ 240,000	$ 70,000	$ 563,000
	ID Total	$ 70,000	$ 120,000	$ 35,000	$ 225,000
	MC Total	$ 62,000	$ 120,000	$ 70,000	$ 252,000
	ME Total	$ 45,000	$ 120,000	$ 10,000	$ 175,000
	NC Total	$ 95,000	$ 45,000	$ 35,000	$ 175,000
	ND Total	$ 150,000	$ 10,000	$ 162,000	$ 322,000
	PA Total	$ 70,000	$ 95,000	$ 35,000	$ 200,000
	RI Total	$ 200,000	$ 215,000	$ 70,000	$ 485,000
	TN Total	$ 45,000	$ 120,000	$ 55,000	$ 220,000
	TX Total	$ 10,000	$ 62,000	$ 35,000	$ 107,000
	Grand Tota	$ 1,462,000	$ 1,507,000	$ 717,000	$ 3,686,000

在pandas中创建分类汇总，是使用`groupby` 来完成的。

Python

```
; html-script: false ]df_sub=df_final[["abbrev","Jan","Feb","Mar","total"]].groupby('
df_sub
```

abbrev				
AR	150000	120000	35000	305000
CA	162000	120000	35000	317000
DE	150000	120000	70000	340000
IA	253000	240000	70000	563000
ID	70000	120000	35000	225000
ME	45000	120000	10000	175000
MS	62000	120000	70000	252000
NC	95000	45000	35000	175000
ND	150000	10000	162000	322000
PA	70000	95000	35000	200000
RI	200000	215000	70000	485000
TN	45000	120000	55000	220000
TX	10000	62000	35000	107000

然后，我们想要通过对data frame中所有的值使用 `applymap` 来把数据单位格式化为货币。

Python

```
; html-script: false ]def money(x):
    return "${:,.0f}".format(x)

formatted_df = df_sub.applymap(money)
formatted_df
```

abbrev				
AR	$150,000	$120,000	$35,000	$305,000
CA	$162,000	$120,000	$35,000	$317,000
DE	$150,000	$120,000	$70,000	$340,000
IA	$253,000	$240,000	$70,000	$563,000
ID	$70,000	$120,000	$35,000	$225,000
ME	$45,000	$120,000	$10,000	$175,000
MS	$62,000	$120,000	$70,000	$252,000
NC	$95,000	$45,000	$35,000	$175,000
ND	$150,000	$10,000	$162,000	$322,000
PA	$70,000	$95,000	$35,000	$200,000
RI	$200,000	$215,000	$70,000	$485,000
TN	$45,000	$120,000	$55,000	$220,000
TX	$10,000	$62,000	$35,000	$107,000

格式化看上去进行的很顺利，现在我们可以像之前那样获取总和了。

Python

```
; html-script: false ]sum_row=df_sub[["Jan","Feb","Mar","total"]].sum()
sum_row
```

Python

```
; html-script: false ]Jan      1462000
Feb      1507000
Mar       717000
total    3686000
dtype: int64
```

把值变换为列然后进行格式化。

Python

```
; html-script: false ]df_sub_sum=pd.DataFrame(data=sum_row).T
df_sub_sum=df_sub_sum.applymap(money)
df_sub_sum
```

	Jan	Feb	Mar	total
0	$1,462,000	$1,507,000	$717,000	$3,686,000

最后，把总和添加到DataFrame中。

```
; html-script: false ]final_table = formatted_df.append(df_sub_sum)
final_table
```

	Jan	Feb	Mar	total
AR	$150,000	$120,000	$35,000	$305,000
CA	$162,000	$120,000	$35,000	$317,000
DE	$150,000	$120,000	$70,000	$340,000
IA	$253,000	$240,000	$70,000	$563,000
ID	$70,000	$120,000	$35,000	$225,000
ME	$45,000	$120,000	$10,000	$175,000
MS	$62,000	$120,000	$70,000	$252,000
NC	$95,000	$45,000	$35,000	$175,000
ND	$150,000	$10,000	$162,000	$322,000
PA	$70,000	$95,000	$35,000	$200,000
RI	$200,000	$215,000	$70,000	$485,000
TN	$45,000	$120,000	$55,000	$220,000
TX	$10,000	$62,000	$35,000	$107,000
0	$1,462,000	$1,507,000	$717,000	$3,686,000

你可以注意到总和行的索引号是‘0’。我们想要使用`rename`来重命名它。

Python

```
; html-script: false ]final_table = final_table.rename(index={0:"Total"})
final_table
```

AR	$150,000	$120,000	$35,000	$305,000
CA	$162,000	$120,000	$35,000	$317,000
DE	$150,000	$120,000	$70,000	$340,000
IA	$253,000	$240,000	$70,000	$563,000
ID	$70,000	$120,000	$35,000	$225,000
ME	$45,000	$120,000	$10,000	$175,000
MS	$62,000	$120,000	$70,000	$252,000
NC	$95,000	$45,000	$35,000	$175,000
ND	$150,000	$10,000	$162,000	$322,000
PA	$70,000	$95,000	$35,000	$200,000
RI	$200,000	$215,000	$70,000	$485,000
TN	$45,000	$120,000	$55,000	$220,000
TX	$10,000	$62,000	$35,000	$107,000
Total	$1,462,000	$1,507,000	$717,000	$3,686,000

结论

到目前为止，大部分人都已经知道使用pandas可以对数据做很多复杂的操作——就如同Excel一样。因为我一直在学习pandas，但我发现我还是会尝试记忆我是如何在Excel中完成这些操作的而不是在pandas中。我意识到把它俩作对比似乎不是很公平——它们是完全不同的工具。但是，我希望能接触到哪些了解Excel并且想要学习一些可以满足分析他们数据需求的其他替代工具的那些人。我希望这些例子可以帮助到其他人，让他们有信心认为他们可以使用pandas来替换他们零碎复杂的Excel，进行数据操作。

我发现这个练习会帮助我加强记忆。我希望这对你来说同样有帮助。如果你有一些其他的Excel任务想知道如何用pandas来完成它，请通过评论来告诉我，我会尽力帮助你。

> **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