Calculate Different Types of Spend - Pandas/Numpy - Python

Asked 8 years, 1 month ago Modified 8 years, 1 month ago Viewed 325 times

1

I have 2 dataframes :

```
df1
+------------+-------------+------+
| Product ID | Cost Method | Rate |
+------------+-------------+------+
|         10 | CPM         | 10   |
|         20 | CPC         | 0.3  |
|         30 | CPCV        | 0.4  |
|         40 | FLF         | 100  |
|         50 | VAD         | 0    |
|         60 | CPM         | 0.1  |
+------------+-------------+------+

df2
+--------+------------+-------------+--------+-----------------+
|  Date  | Product ID | Impressions | Clicks | Completed Views |
+--------+------------+-------------+--------+-----------------+
| 01-Jan |         10 |         300 |      4 |               0 |
| 02-Jan |         20 |          30 |      3 |               0 |
| 03-Jan |         30 |         200 |      4 |              20 |
| 02-Jan |         40 |         300 |      4 |               0 |
| 02-Jan |         40 |         500 |      4 |               0 |
| 03-Jan |         40 |         200 |      3 |               0 |
| 04-Jan |         90 |        3000 |      3 |               0 |
| 05-Jan |         50 |        3000 |      5 |               0 |
+--------+------------+-------------+--------+-----------------+
```

The ideal output is this:

```
 +--------+------------+-------------+--------+-----------------+--------+
|  Date  | Product ID | Impressions | Clicks | Completed Views | Spend  |
+--------+------------+-------------+--------+-----------------+--------+
| 01-Jan |         10 |         300 |      4 |               0 | $3     |
| 02-Jan |         20 |          30 |      3 |               0 | $1     |
| 03-Jan |         30 |         200 |      4 |              20 | $8     |
| 02-Jan |         40 |         300 |      4 |               0 | $50    |
| 02-Jan |         40 |         500 |      4 |               0 | $50    |
| 03-Jan |         40 |         200 |      3 |               0 | $-     |
| 04-Jan |         90 |        3000 |      3 |               0 | $-     |
| 05-Jan |         50 |        3000 |      5 |               0 | $-     |
+--------+------------+-------------+--------+-----------------+--------+
```

Where :

1. Product is Matched by its ID In case an ID can't be Matched, then the product spend is calculated at 0
2. Where FLF is calculated as the sum of total impressions for that product per day, and if that sums is over a certain minimum limit, e.g. 600 impressions, then the rate is applied. If there are two or more entries for the same day, then the rate is divided equally by the count of times it appears in the same day
3. Where, if a product is VAD, then the spend is 0
4. Where CPC is calculated as the rate times the number of clicks
5. Where CPM is calculated as rate*(impression / 1000)

python pandas numpy analysis

Share Improve this question Follow

asked Nov 10, 2016 at 7:58
Matteo M
147 ● 2 ● 8

1 Hi, I don't mean to be rude but this is not a homework service. Have you tried something? What specific road blocks did you encounter – Julien Marrec Nov 10, 2016 at 9:50

Hi Julien, absolutely! The biggest problem is making sure that FLF is calculated on the total of the day, and that then the valued is split by the times it occurs – Matteo M Nov 10, 2016 at 10:02

1 Answer

Sorted by: Highest score (default)

2

I'm going to answer to you even though I shouldn't really. You're new on Stack Overflow (SO), so let this be an educational post. Rest assured that the tone of this post isn't trying to be condescending or harsh.

First, to ask a proper question (read this please) you need to do two things:

- Explain what you have tried (provide a code sample!) and explain what your problem is. Your question in its current format definitely doesn't comply. There's like 5 or 6 completely different things in it, and it feels like you're just asking for someone to do your homework.
- Provide a workable example.

For the workable example, you kind of did this, but the format you choose is really annoying since one cannot directly use `pd.read_clipboard()` to load the data. People here are **volunteering** their time, and if they have to spend 5 or 10 minutes recreating your data they likely just won't do it.

Here's how I would have done it:

Here is the first dataframe, use `df1 = pd.read_clipboard(index_col=0)` to load it:

```
ProductID      CostMethod   Rate

10                CPM   10.0
20                CPC    0.3
30               CPCV    0.4
40                FLF  100.0
50                VAD    0.0
60                CPM    0.1
```

Here is the second dataframe, use `df2 = pd.read_clipboard(index_col=0)` to load it:

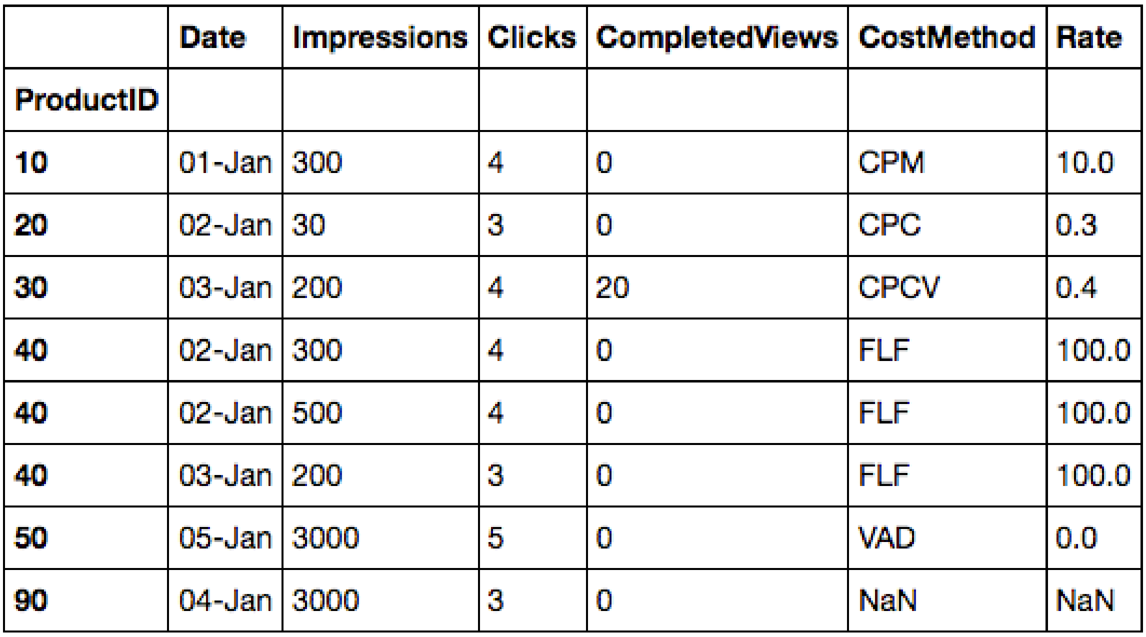
```
ProductID  Date   Impressions  Clicks  CompletedViews
10         01-Jan          300       4               0
20         02-Jan           30       3               0
30         03-Jan          200       4              20
40         02-Jan          300       4               0
40         02-Jan          500       4               0
40         03-Jan          200       3               0
90         04-Jan         3000       3               0
50         05-Jan         3000       5               0
```

Now, as far as doing your homework, here's a proposed solution. I trust that you will try to understand what this code does and not just reuse it.

Step 1: Merge both dataframes

I'm merging left on df2, that's really important. Read more in the pandas documentation on Merging

```
df3 = df2.merge(df1, left_index=True, right_index=True, how='left')
df3
```



	Date	Impressions	Clicks	CompletedViews	CostMethod	Rate
ProductID						
10	01-Jan	300	4	0	CPM	10.0
20	02-Jan	30	3	0	CPC	0.3
30	03-Jan	200	4	20	CPCV	0.4
40	02-Jan	300	4	0	FLF	100.0
40	02-Jan	500	4	0	FLF	100.0
40	03-Jan	200	3	0	FLF	100.0
50	05-Jan	3000	5	0	VAD	0.0
90	04-Jan	3000	3	0	NaN	NaN

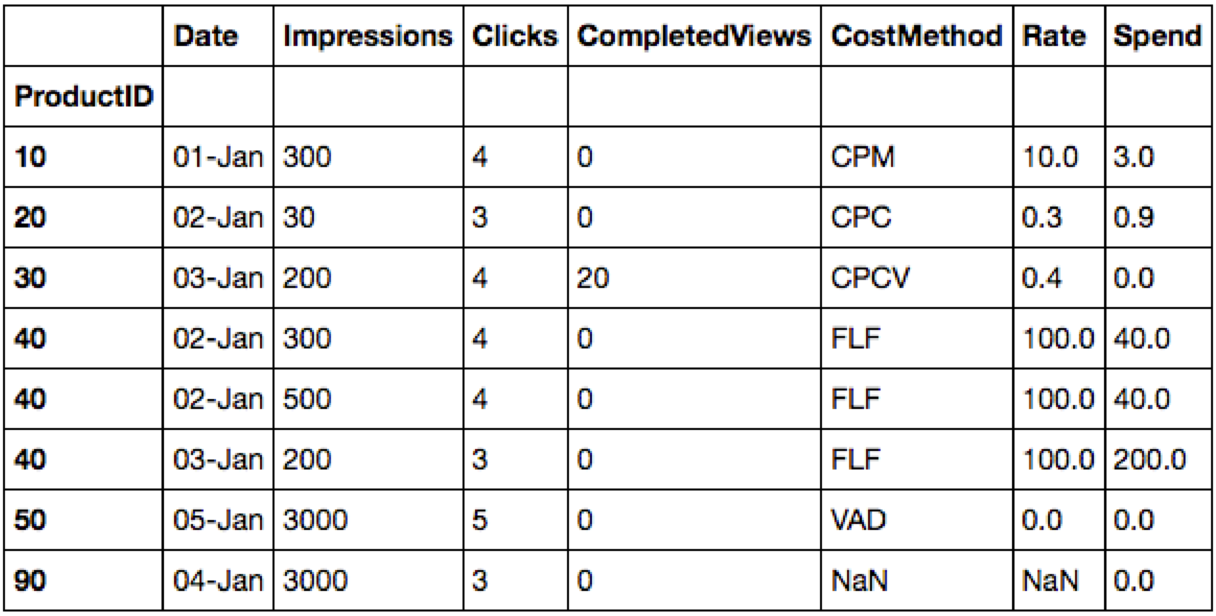
Step 2: calculate your spend

We're going to write a custom function and then do dataframe.apply

```
def calc_spend(row):
    """
    Accepts a row of the dataframe (df3.apply(calc_spend, axis=1)),
    and computes the spend according to these rules:
    * If costMethod is NaN, then zero
    * Where FLF is calculated as the sum of total impressions for that product
per day,
        and if that sums is over a certain minimum limit,
        e.g. 600 impressions, then the rate is applied.
        If there are two or more entries for the same day,
        then the rate is divided equally by the count of times it appears in
the same day
    * Where, if a product is VAD, then the spend is 0
    * Where CPC is calculated as the rate times the number of clicks
    * Where CPM is calculated as rate*(impression / 1000)
    """

    if row.CostMethod == 'FLF':
        # Calc the sum of total impressions for that product
        # I'm using boolean indexing to select the rows where both productID
and Date
        # are the same as the current row
        filterdateproductid = (df3.Date == row.Date) & (df3.index == row.name)
        total_impressions = df3.ix[filterdateproductid, 'Impressions'].sum()
        if total_impressions < 600:
            spend = total_impressions
        else:
            count = df3.ix[filterdateproductid].shape[0]
            rate = row.Rate / count # If you use python 2.7 make sure you do
"from future import division"
            spend = rate * total_impressions / 1000.0

    elif row.CostMethod == 'VAD':
        spend = 0

    elif row.CostMethod == 'CPC':
        spend = row.Rate * row.Clicks

    elif row.CostMethod == 'CPM':
        spend = row.Rate * row.Impressions / 1000.0

    else: # Includes the case where the costMethod is Na
        spend = 0

    return spend
```

Now we can just apply the function itself:

```
df3['Spend'] = df3.apply(calc_spend, axis=1)
df3
```



	Date	Impressions	Clicks	CompletedViews	CostMethod	Rate	Spend
ProductID							
10	01-Jan	300	4	0	CPM	10.0	3.0
20	02-Jan	30	3	0	CPC	0.3	0.9
30	03-Jan	200	4	20	CPCV	0.4	0.0
40	02-Jan	300	4	0	FLF	100.0	40.0
40	02-Jan	500	4	0	FLF	100.0	40.0
40	03-Jan	200	3	0	FLF	100.0	200.0
50	05-Jan	3000	5	0	VAD	0.0	0.0
90	04-Jan	3000	3	0	NaN	NaN	0.0

You'll perhaps notice that the "Spend" I calculated isn't exactly the same as yours, but this is because your initial specs on how to calculate it weren't so great. It will be easy for you to change the `calc_spend` function to match your requirements.

Share

Improve this answer

Follow

edited May 23, 2017 at 11:46

Community Bot

1 ● 1

answered Nov 11, 2016 at 15:13

Julien Marrec

11.9k ● 5 ● 50 ● 66

1 Hi Julien, Thank you so much for your help. I have started working with Python this week (coming from excel), and I am trying to migrate my work files from excel to python (learning as going along). Usually I managed to get things going. But here I was really stumped. I am really grateful for the time you took for answering this question. It really means alot to me! And I will be sure to format my questions better next time as suggested by you – Matteo M Nov 11, 2016 at 15:46