

Microservice for Rounding and Matrix Update

Lars Wentzel 2022-03-30

If you are interested to test the below services, you can contact me, and I will give you the server address.

Lars Wentzel

lars.wentzel@aplensia.se

The problem

Why do you need to round a sequence or ta matrix (table) of decimal numbers? Well, the simple answer is that the rounded total (sum) is not equal to the sum of the single rounded values.

You can get this problem in several situations but for me it stated with a block update of a table or matrix. That means that I want to change the grand total to a new higher or lower value. Then I will try to keep the mix of the single numbers. The assumption is then that you only want integers in the result like if you have vehicles, you don't want decimal vehicles. So, when you have calculated the new numbers some of them are decimal values. Then you would like to make ta matrix (table) round.

This handling is useful in e.g., planning. Let's take an example from the Swedish Systembolaget (the state-owned alcohol shops). The sales statistics is on city, product and week and you measure everything in hL (100 liters). The products can be grouped on in lots of groups e.g., beer, wine, and liquor or country and type. The time (weeks) can also be aggregated to quarters, half year, or years.

I will simplify this only having two cities and a few products.

Base forecast based on statistics

City	Group	Country	Type	Product	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10	w11	w12	w13	q1
Stockholm	Wine	Italy	Red	A1	4	5	9	1	5	9	1	5	5	4	2	4	7	61
Stockholm	Wine	Italy	Sparkling	A2	8	2	7	3	5	8	9	9	9	5	9	5	5	84
Stockholm	Wine	Italy	White	A3	7	9	3	3	6	6	8	6	3	4	8	9	2	74
Stockholm	Wine	France	Red	A4	7	8	3	7	8	8	7	7	7	3	6	6	6	83
Stockholm	Wine	France	Sparkling	A5	1	7	1	9	9	8	6	7	7	5	2	4	2	68
Stockholm	Wine	France	White	A6	1	6	2	2	5	2	1	8	2	4	6	9	5	53
Stockholm	Wine	Chile	Red	A7	5	8	3	6	8	1	9	8	7	7	7	8	7	84
Stockholm	Wine	Chile	White	A8	4	7	5	8	3	9	1	4	9	4	2	6	5	67
Stockholm	Wine	Germany	White	A9	4	9	1	7	5	3	1	6	6	5	5	1	7	60
Stockholm	Wine	Germany	Sparkling	A10	4	2	3	7	8	4	6	4	9	9	6	4	3	69
Göteborg	Wine	Italy	Red	A1	3	2	1	2	2	9	3	1	2	6	9	3	2	45
Göteborg	Wine	Italy	Sparkling	A2	1	1	4	6	5	8	1	5	4	1	2	7	6	51
Göteborg	Wine	Italy	White	A3	3	8	2	8	8	6	7	5	3	6	8	8	2	74
Göteborg	Wine	France	Red	A4	7	3	9	1	5	8	9	2	2	5	4	3	3	61
Göteborg	Wine	France	Sparkling	A5	5	3	6	8	3	1	5	3	9	1	9	5	4	62
Göteborg	Wine	France	White	A6	9	9	2	4	4	9	1	3	4	8	1	4	1	59
Göteborg	Wine	Chile	Red	A7	6	8	8	2	4	4	3	5	3	9	6	7	7	72
Göteborg	Wine	Chile	White	A8	2	1	2	6	9	3	1	1	3	6	7	1	5	47
Göteborg	Wine	Germany	White	A9	2	4	3	7	8	7	6	1	6	4	4	5	2	59
Göteborg	Wine	Germany	Sparkling	A10	5	8	2	2	7	4	1	7	6	4	6	3	8	63
Tot					88	110	76	99	117	117	86	97	106	100	109	102	89	1296

You then want to increase the sparkling wine during q1. You then have a sub-table.

Stockholm	Wine	Italy	Sparkling	A2	8	2	7	3	5	8	9	9	9	5	9	5	5	84
Stockholm	Wine	France	Sparkling	A5	1	7	1	9	9	8	6	7	7	5	2	4	2	68
Stockholm	Wine	Germany	Sparkling	A10	4	2	3	7	8	4	6	4	9	9	6	4	3	69
Göteborg	Wine	Italy	Sparkling	A2	1	1	4	6	5	8	1	5	4	1	2	7	6	51
Göteborg	Wine	France	Sparkling	A5	5	3	6	8	3	1	5	3	9	1	9	5	4	62
Göteborg	Wine	Germany	Sparkling	A10	5	8	2	2	7	4	1	7	6	4	6	3	8	63
Tot					24	23	23	35	37	33	28	35	44	25	34	28	28	397

You update from 397 to 450 and keep the mix. You multiply each cell with 450/397. Then you round each cell to integers, and you get:

Stockholm	Wine	Italy	Sparkling	A2	9	2	8	3	6	9	10	10	10	6	10	6	6	95
Stockholm	Wine	France	Sparkling	A5	1	8	1	10	10	9	7	8	8	6	2	5	2	77
Stockholm	Wine	Germany	Sparkling	A10	5	2	3	8	9	5	7	5	10	10	7	5	3	79
Gothenburg	Wine	Italy	Sparkling	A2	1	1	5	7	6	9	1	6	5	1	2	8	7	59
Gothenburg	Wine	France	Sparkling	A5	6	3	7	9	3	1	6	3	10	1	10	6	5	70
Gothenburg	Wine	Germany	Sparkling	A10	6	9	2	2	8	5	1	8	7	5	7	3	9	72
Tot					28	25	26	39	42	38	32	40	50	29	38	33	32	452

Now you have a problem because the total is not 450.

To adjust this you use the service (function) **MtxRound** to adjust the rounding and you will get.

Stockholm	Wine	Italy	Sparkling	A2	9	2	8	3	6	9	10	10	10	6	10	6	6	95
Stockholm	Wine	France	Sparkling	A5	1	8	1	10	10	9	7	8	8	6	2	5	2	77
Stockholm	Wine	Germany	Sparkling	A10	4	3	3	8	9	4	7	5	11	10	7	4	3	78
Gothenburg	Wine	Italy	Sparkling	A2	1	1	5	7	6	9	1	6	4	1	2	8	7	58
Gothenburg	Wine	France	Sparkling	A5	6	3	7	9	3	1	6	3	10	1	10	6	5	70
Gothenburg	Wine	Germany	Sparkling	A10	6	9	2	3	8	5	1	8	7	4	7	3	9	72
Tot					27	26	26	40	42	37	32	40	50	28	38	32	32	450

You can use the **MtxRound** service separately or you can use the **MtxUpdate** service to do the complete job to update the table.

The services

These are HTTP JSON services to round a vector (number series) or a matrix. The purpose is to get only integers from a vector or matrix of decimal number.

Finally, it is also a service to make a proportional block update of a matrix of integers. This may result in decimal numbers. These numbers are then rounded to integers.

These services handle only 0 or positive numbers.

The services are

VectRound Rounds a vector (number series) of decimal numbers to integers keeping the total sum.

MtxRound Rounds a matrix of decimal numbers to integers keeping the column totals and the row totals as good as possible.

MtxUpdate Updates an integer matrix with a new grand total **or** new column totals in proportion to the earlier values. Then rounds the values to integers. There is also a possibility to give a second matrix with the lowest possible values that each cell can have.

HTTP request should look like this

POST /service HTTP/1.1 where service is the name of one of the services above

Content-Length: nnn

Followed by the JSON code.

The response will look like this

HTTP/1.1 200 OK Content-Type application/json Server Lars test

Content-Length: nnn

Followed by the JSON code.

Examples

VectRound

You have a series of numbers

3.9 2.8 10.7 4.6 6.5 7 1 9 8 11.6 5.6

The sum of this is 70.7. This is rounded to 71.

If you round each number to integer so that .500-.9 becomes 1 and 0.1-.499 becomes 0 you will get 73.

4 3 11 5 7 7 1 9 8 12 6 Sum=73

Now the purpose with this service is to keep the sum. Then you need to adjust some of the integers.

4 3 11 4 6 7 1 9 8 12 6

The service will take those numbers that are closest to be rounded the other way, up or down.

If there are several values that are equal like 4.6 and 5.6 here a random mechanism is used to find a reasonable spreading.

Zeros are not touched.

Service to send: *VectRound*

JSON to send

```
{"vector": [3.9, 2.8, 10.7, 4.6, 6.5, 7, 1, 9, 8, 11.6, 5.6]}
```

Result

```
{"rounded": [4, 3, 11, 4, 6, 7, 1, 9, 8, 12, 6]}
```

MtxRound

You have a matrix of numbers

							Totals	Rounded
	0	9.4	0	1.5	5.5	0	16.4	17
	2.4	4.3	3.5	0	9.4	6.5	26.1	26
	1.3	5.4	8.4	2.5	4.3	3.4	25.3	25
	7.5	9.3	6.5	1.5	5.5	0	30.3	30
	0	0	3.5	7.4	9.4	0	20.3	20
	1.4	5.4	8.3	0	4.3	3.4	22.8	23
	7.5	9.5	6.4	1.4	0	8.3	33.1	33
Totals	20.1	43.3	36.6	14.3	38.4	21.6	174.3	
Rounded	20	43	37	14	38	22		174

The result will be

							Totals
	0	10	0	1	6	0	17
	2	4	4	0	10	6	26
	1	5	9	2	4	4	25
	8	9	6	2	5	0	30
	0	0	3	8	9	0	20
	1	5	9	0	4	4	23
	8	10	6	1	0	8	33
Totals	20	43	37	14	38	22	174

The grand total (rounded 174) must always be kept.

The rounded column totals are adapted to the grand total using the logic in **VectRound**.

And the row totals are also adapted to the grand total.

The rounding of the matrix values is done so the column totals are always kept.

The row totals are kept *if it is possible*. The zeroes are never touched.

Service to send: *MtxRound*

JSON to send

```
{"matrix":[[1,2,3.5],[5.1,0,8.97]]}
```

The matrix is expressed as a series of row series. So, the matrix represented in the above JSON is:

```
1      2      3.5
5.1    0      8.97
```

Result

```
{"rounded":[[1,2,4],[5,0,9]]}
```

MtxUpdate

1. Change the grand total

You have a matrix of integers. The grand total is 174. Update with a new grand total 200.

	0	10	0	1	6	0	Totals	17
	2	4	4	0	10	6		26
	1	5	9	2	4	4		25
	8	9	6	2	5	0		30
	0	0	3	8	9	0		20
	1	5	9	0	4	4		23
	8	10	6	1	0	8		33
Totals	20	43	37	14	38	22		174

The result will be

	0	11	0	1	7	0	Totals	19
	3	4	5	0	11	7		30
	1	6	10	2	5	5		29
	9	10	7	3	6	0		35
	0	0	4	9	10	0		23
	1	6	10	0	5	4		26
	9	12	7	1	0	9		38
Totals	23	49	43	16	44	25		200

Each single number is multiplied with $200/174$ (1.1494...). Then the matrix is rounded with *MtxRound*.

Service to send: *MtxUpdate*

JSON to send

```
{"matrix":[[0,10,0,1,6,0],[2,4,4,0,10,6],[1,5,9,2,4,4],[8,9,6,2,5,0],[0,0,3,8,9,0],[1,5,9,0,4,4],[8,10,6,1,0,8]],  
"total":200}
```

Result

```
{"updated":[[0,11,0,1,7,0],[3,4,5,0,11,7],[1,6,10,2,5,5],[9,10,7,3,6,0],[0,0,4,9,10,0],[1,6,10,0,5,4],[9,12,7,  
1,0,9]]}
```

2. Change the column totals

You have a matrix of integer numbers. The column totals are 23 49 43 16 44 25. Change them to 25 44 40 19 39 31.

	0	11	0	1	7	0	Totals	19
	3	4	5	0	11	7		30
	1	6	10	2	5	5		29
	9	10	7	3	6	0		35
	0	0	4	9	10	0		23
	1	6	10	0	5	4		26
	9	12	7	1	0	9		38
Totals	23	49	43	16	44	25		200

The result will be

	0	10	0	1	6	0	Totals
	3	3	5	0	10	9	17
	1	6	9	3	4	6	30
	10	9	6	4	5	0	29
	0	0	4	10	9	0	34
	1	5	9	0	5	5	23
	10	11	7	1	0	11	25
Totals	25	44	40	19	39	31	40
							198

For each column the numbers are multiplied with new value divided with previous value.
E.g., in the last column you multiply with 31/25. Then the matrix is rounded with *MtxRound*.

Service to send: *MtxUpdate*

JSON to send

```
{"matrix":[[0,11,0,1,7,0],[3,4,5,0,11,7],[1,6,10,2,5,5],[9,10,7,3,6,0],[0,0,4,9,10,0],[1,6,10,0,5,4],[9,12,7,1,0,9]],"total":[25,44,40,19,39,31]}
```

result

```
{"updated":[[0,10,0,1,6,0],[3,3,5,0,10,9],[1,6,9,3,4,6],[10,9,6,4,5,0],[0,0,4,10,9,0],[1,5,9,0,5,5],[10,11,7,1,0,11]]}
```

3. Change the grand/total or the column totals but give also lowest possible values for each cell.
The grand total is changed to 131

	0	10	0	1	6	0	Totals
	2	4	4	0	10	6	17
	1	5	9	2	4	4	26
	8	9	6	2	5	0	25
	0	0	3	8	9	0	30
	1	5	9	0	4	4	20
	8	10	6	1	0	8	23
Totals	20	43	37	14	38	22	33
							174

The lowest possible values are

	0	5	0	1	3	0	Total
	1	2	2	0	8	1	9
	0	1	5	1	2	2	14
	7	1	1	2	0	0	11
	0	0	1	2	3	0	11
	0	0	0	0	0	0	6
	8	10	6	1	0	8	0
Total	16	19	15	7	16	11	33
							84

The result is

	0	6	0	1	4	0	Total 11
	1	2	4	0	8	4	19
	0	3	7	1	3	3	17
	7	7	3	2	4	0	23
	0	0	2	5	7	0	14
	0	3	6	0	3	2	14
	8	10	6	1	0	8	33
Total	16	31	28	10	29	17	131

As in earlier updates the single values are multiplied with the update: 131/174. However, in this case you can never go below the corresponding minimum value when rounding.

Service to send: *MtxUpdate*

JSON to send

```
{"matrix":[[0,10,0,1,6,0],[2,4,4,0,10,6],[1,5,9,2,4,4],[8,9,6,2,5,0],[0,0,3,8,9,0],[1,5,9,0,4,4],[8,10,6,1,0,8]],  
"minimum":[[0,5,0,1,3,0],[1,2,2,0,8,1],[0,1,5,1,2,2],[7,1,1,2,0,0],[0,0,1,2,3,0],[0,0,0,0,0,0],[8,10,6,1,0,8]],  
"total":131}
```

result

```
{"updated":[[0,6,0,1,4,0],[1,2,4,0,8,4],[0,3,7,1,3,3],[7,7,3,2,4,0],[0,0,2,5,7,0],[0,3,6,0,3,2],[8,10,6,1,0,8]]}
```