Question 1

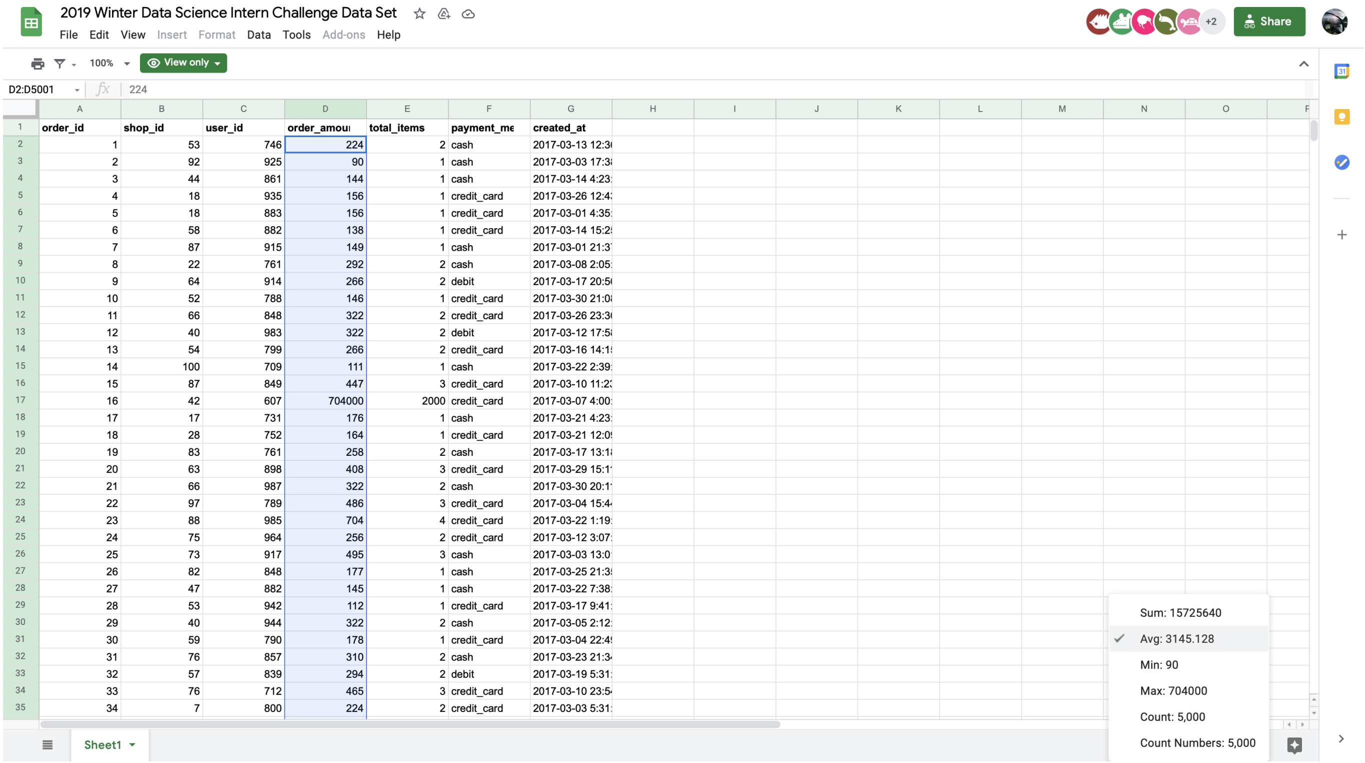
To run the python program simply type

```
python aov.py
```

a.

What went wrong with the calculation:

Intuitively, the problem could be that the total sum of the order amount was divided by a smaller number, like the total number of orders made (as compared to sum of the number of items). I confirmed this by selecting the values in the `order_amount` column on the google sheet provided and voila (bottom right), the default average over all the orders was equal to the wrongly calculated value that was given!



D2:D5001 | 224

	A	B	C	D	E	F	G
1	order_id	shop_id	user_id	order_amou	total_items	payment_me	created_at
2	1	53	746	224	2	cash	2017-03-13 12:3
3	2	92	925	90	1	cash	2017-03-03 17:3
4	3	44	861	144	1	cash	2017-03-14 4:23
5	4	18	935	156	1	credit_card	2017-03-26 12:4
6	5	18	883	156	1	credit_card	2017-03-01 4:35
7	6	58	882	138	1	credit_card	2017-03-14 15:2
8	7	87	915	149	1	cash	2017-03-01 21:3
9	8	22	761	292	2	cash	2017-03-08 2:05
10	9	64	914	266	2	debit	2017-03-17 20:5
11	10	52	788	146	1	credit_card	2017-03-30 21:0
12	11	66	848	322	2	credit_card	2017-03-26 23:3
13	12	40	983	322	2	debit	2017-03-12 17:5
14	13	54	799	266	2	credit_card	2017-03-16 14:1
15	14	100	709	111	1	cash	2017-03-22 2:39
16	15	87	849	447	3	credit_card	2017-03-10 11:2
17	16	42	607	704000	2000	credit_card	2017-03-07 4:00
18	17	17	731	176	1	cash	2017-03-21 4:23
19	18	28	752	164	1	credit_card	2017-03-21 12:0
20	19	83	761	258	2	cash	2017-03-17 13:1
21	20	63	898	408	3	credit_card	2017-03-29 15:1
22	21	66	987	322	2	cash	2017-03-30 20:1
23	22	97	789	486	3	credit_card	2017-03-04 15:4
24	23	88	985	704	4	credit_card	2017-03-22 1:19
25	24	75	964	256	2	credit_card	2017-03-12 3:07
26	25	73	917	495	3	cash	2017-03-03 13:0
27	26	82	848	177	1	cash	2017-03-25 21:3
28	27	47	882	145	1	cash	2017-03-22 7:38
29	28	53	942	112	1	credit_card	2017-03-17 9:41
30	29	40	944	322	2	cash	2017-03-05 2:12
31	30	59	790	178	1	credit_card	2017-03-04 22:4
32	31	76	857	310	2	cash	2017-03-23 21:3
33	32	57	839	294	2	debit	2017-03-19 5:31
34	33	76	712	465	3	credit_card	2017-03-10 23:5
35	34	7	800	224	2	credit_card	2017-03-03 5:31

Sum: 15725640
✓ Avg: 3145.128
Min: 90
Max: 704000
Count: 5,000
Count Numbers: 5,000

b.

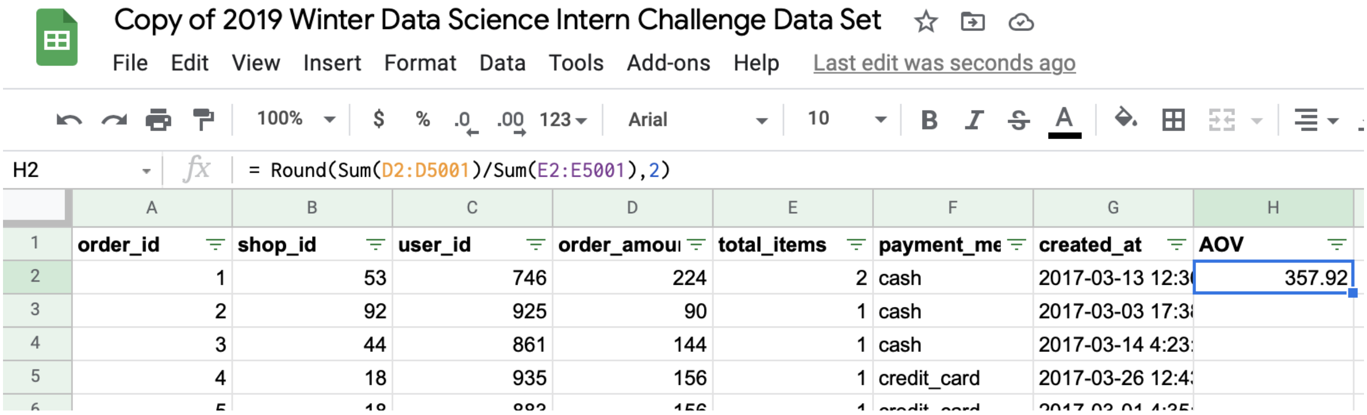
What metric would you report for this dataset

To calculate the average order value we must divide the total revenue by the number of items bought.

So in this case the following formula would:

```
Average order value (AOV) = sum(order_amount)/sum(total_items)
```

First, I created a copy of the given data to get an approximate target in google sheets itself:



Copy of 2019 Winter Data Science Intern Challenge Data Set

= Round(Sum(D2:D5001)/Sum(E2:E5001),2)

	A	B	C	D	E	F	G	H
1	order_id	shop_id	user_id	order_amou	total_items	payment_me	created_at	AOV
2	1	53	746	224	2	cash	2017-03-13 12:3	357.92
3	2	92	925	90	1	cash	2017-03-03 17:3	
4	3	44	861	144	1	cash	2017-03-14 4:23	
5	4	18	935	156	1	credit_card	2017-03-26 12:4	
6	5	18	883	156	1	credit_card	2017-03-01 4:35	

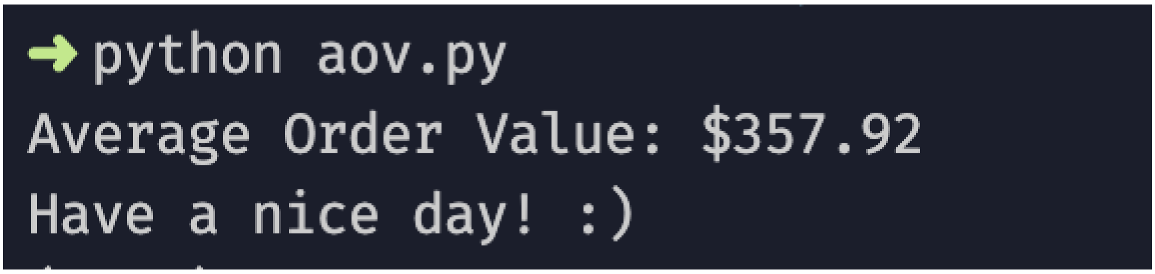
The value I predict the AOV should be is `$357.92`, which is the average order amount for every sneaker sold.

c.

What is its value:

Then, I downloaded the original data set as a csv file and wrote a python script to sanitize the data and calculate the AOV for *each* sneaker.

Here is the output when I run the program:



```
➜ python aov.py
Average Order Value: $357.92
Have a nice day! :)
```

The value calculated is printed out to be `$357.92`, just as I predicted and the program greets the user a good day! :^)

`Final value: $357.92`

(PS: The instructions to run the script is on top of this page)