# **Queries and Doubts**

1. What is filler data?

Ans: dummy data

1. can we use news data or other data other than book and trade data?

Ans: No because we do not know the exact time! time\_id is *shuffled*

1. why do some stocks miss data or their analysis in graphs is not showing?

Ans: Those stocks have missing data.

1. When calculating realised volatility did you first ffill for the missing seconds in book data?

Ans: Actually, forward fill does not affect the volatility calculation because log (s\_t2/s\_t1) = log(1) = 0 in the ffill period. Prices/WAP remain the same. It doesn't affect volatility.

1. Are the bid\_price1 and bid\_price1 from different time\_ids and stocks comparable? E.g. if bid\_price1 = 0.9 in stock\_id = 10 and time\_id = 5 equal to bid\_price1 = 0.9 in stock id = 20 and time\_id = 11 equal? Similar question for ask\_price1 ?
   1. Is the price in book data and trade data comparable? Have they been normalised together or separately. If normalised together then they are comparable if separately then they are NOT comparable.

Possible ans: <https://www.kaggle.com/competitions/optiver-realized-volatility-prediction/discussion/249474>

Normalisation is done separately for time, stock id so different time, stock id prices should not be comparable, but prices across the same stock id for same time id should be comparable as they would have the same mean and std? -> makes sense

1. Probably together based on inference from the discussion thread -> you mean book and trade data are comparable for the same stock\_id and time\_id right?

yep

Thanks @fegetable for your reply 🙂. But then, how can we compare the calculated realized volatility using the WAP formula across time periods of the same stock? If the bid/ask prices are not comparable across different time\_ids then how is the calculated volatility comparable?

I was wondering about this also haha, <https://www.kaggle.com/c/optiver-realized-volatility-prediction/discussion/267327>

This thread in particular:

--“Here it seems to mean that all price start at 1, so probably dividing each time seris by the initial price. Global price have been deanonymised by looking at ticks.”

-- “Thanks for commenting. Dividing time series by a global (per stock) value, which happens to be the price at some time point, makes some sense, however I observed that that value is revisited often, see notebook above. This suggests that the divisor is rather local than global, which could mean that every time slot was independently "normalized".

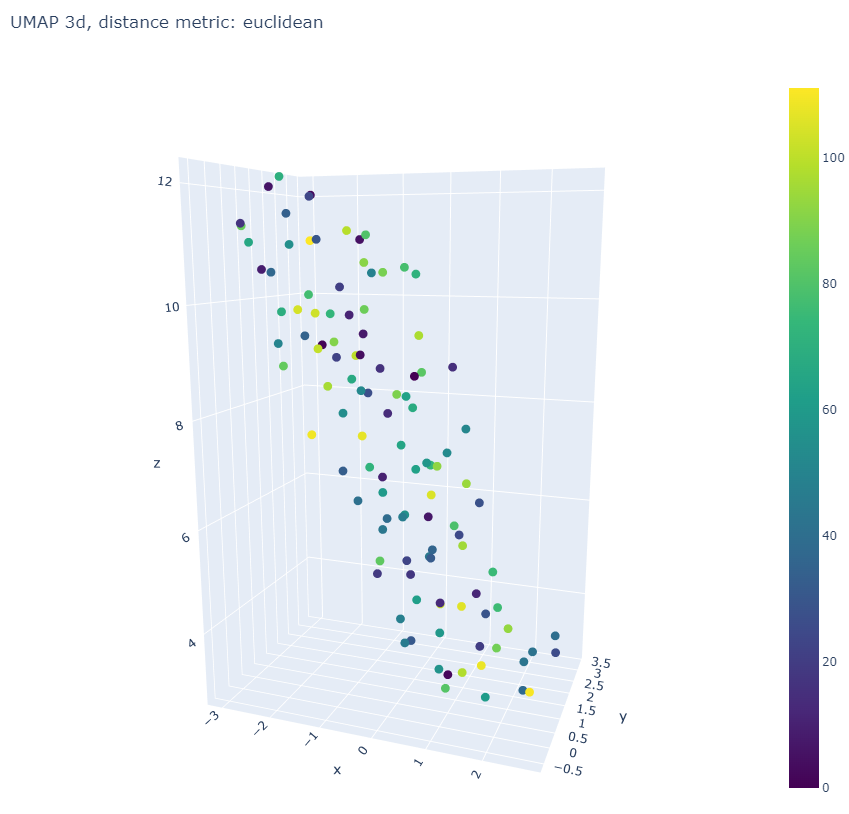
How about the standard deviation? Multiplication/Division by a constant changes it accordingly but we actually want the volatility to be consistent over time slots and stocks, do we?”

-- “Dividing all prices by a constant (per stock and per time bucket) value doesn't change the returns and the volatility: The return is defined as the ratio between two subsequent prices and this ratio is invariant under the chosen transformation.

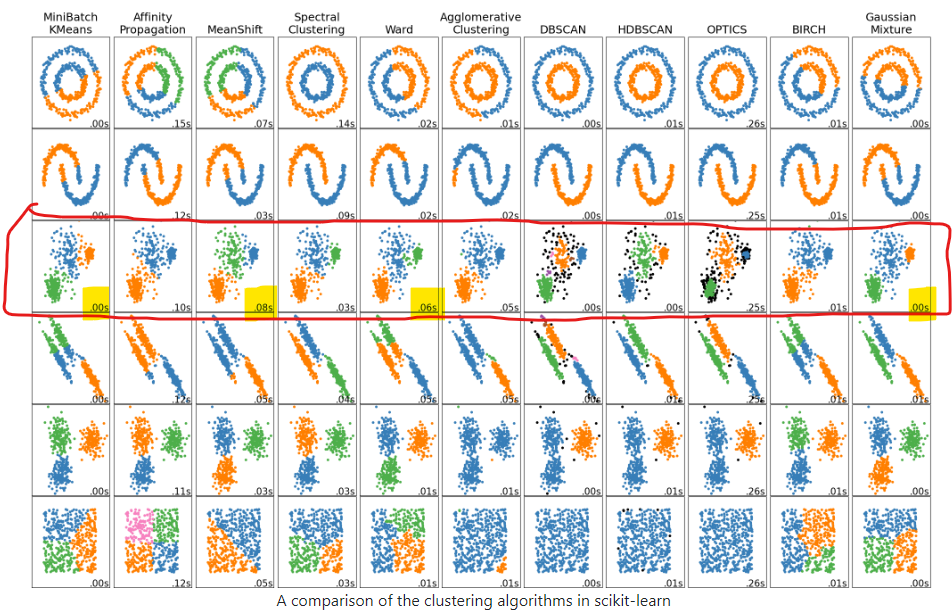
If you look at the logs: dividing the prices by a constant is equivalent to subtracting a constant from all logs of prices. The log return (as difference of two subsequent logs of prices) doesn't change under this transformation.”

From the discussion above, im understanding that since normalisation is done by dividing by a constant, the calculation of log returns and therefore realised vol for every individual stock and time id is not affected.

@fegetable, good job! Log( s\_t2/k / s\_t1/k ) is same as Log( s\_t2 / s\_t1 ) for all time\_id. We don't care about different k in different time\_id as they always get cancelled out.

6. What clustering algo. can separate the stocks into clusters (similar stocks based on summary stats. features) if stocks’ target volatility (in 2nd 10 mins) are distributed like below? Each colour represents one of the 112 stocks. The 3 dimensional view is after reducing 7 dimensions of ['mean\_vol','std\_vol','min\_vol','p25\_vol', 'median\_vol', 'p75\_vol','max\_vol'] using UMAP algo.

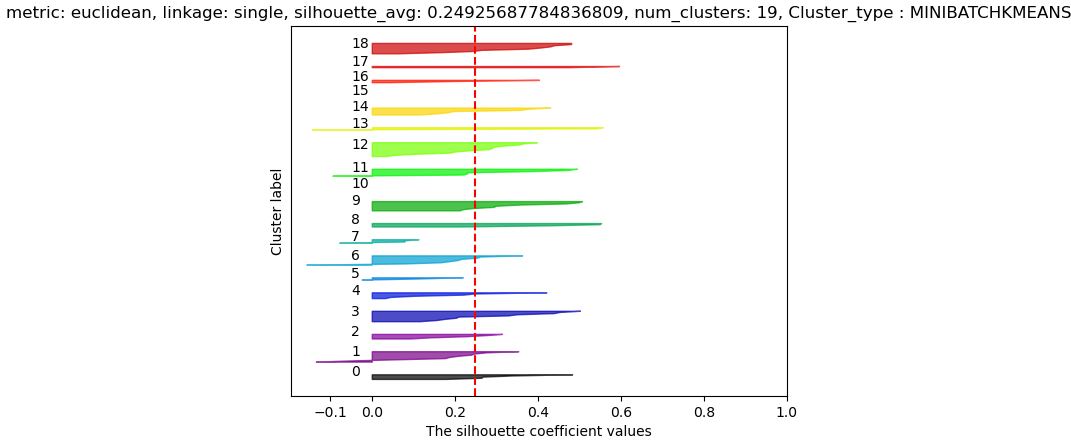
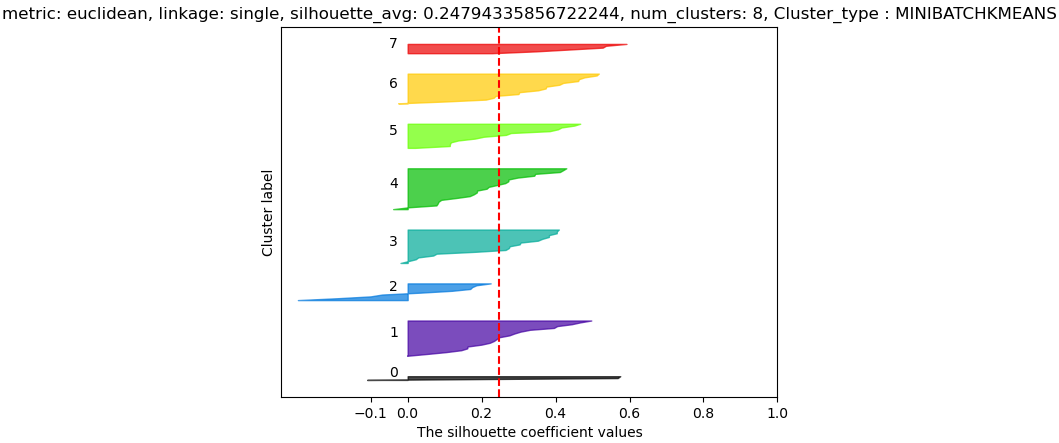
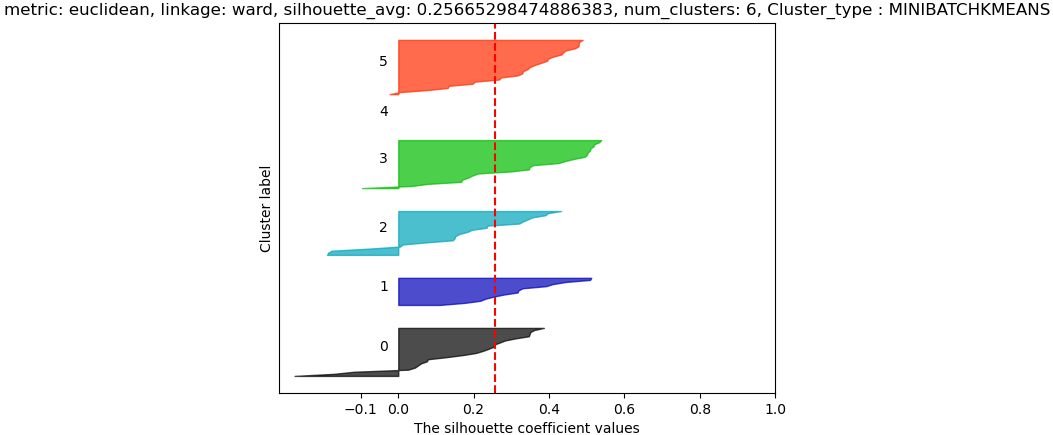
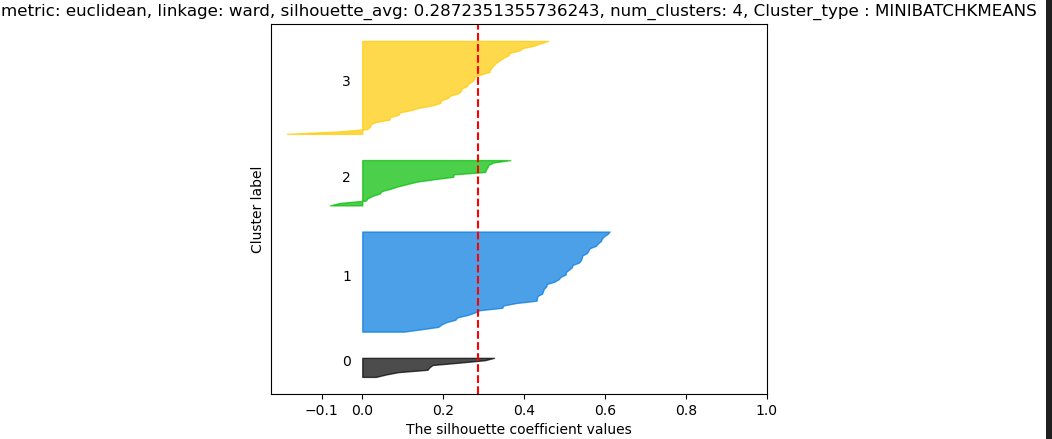
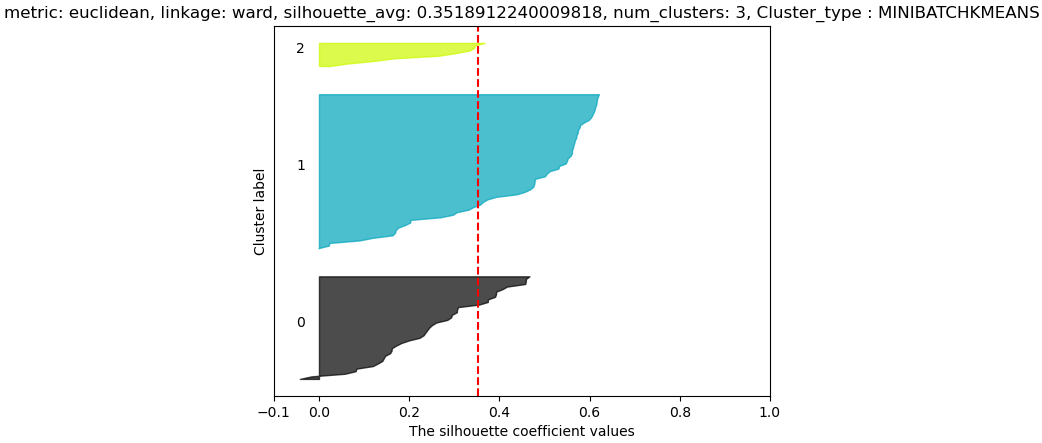
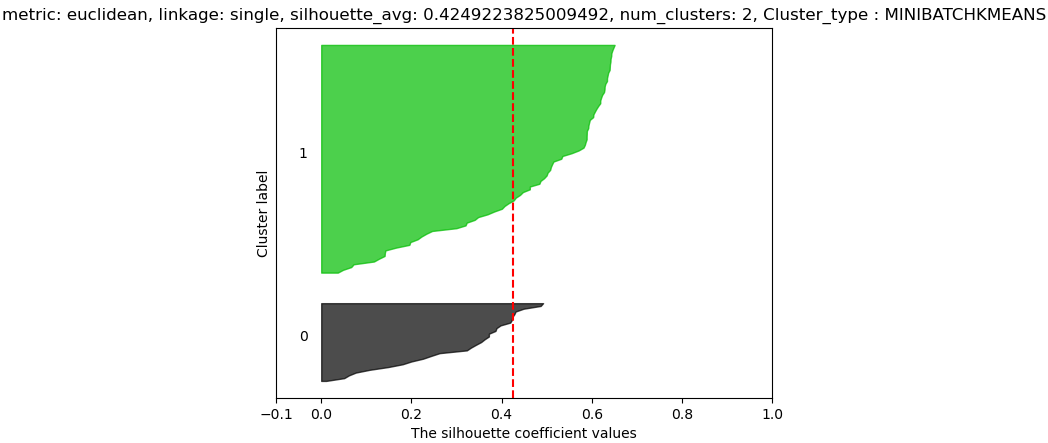
Possible answer: <https://scikit-learn.org/stable/modules/clustering.html>



Because our data distribution (image 3d projected to 2d) looks like the 3rd row above I will try minibatch kmeans,spectral clustering, meanshift, ward and gaussian mixture model clustering algos. WARD is very goo

Yep sounds good, seems like k-means is quite suitable

In file target\_eda\_across\_stocks.ipynb Tried all clustering but WARD is the best. Kmeans is second best. Updated in Key Insights file.



7) features derived from bidsize/asksize were affected by stock splits, may not be reliable as data did not account for stock splits?

In a stock split **the number of outstanding shares increases** and the price per share decreases proportionately, so stock split equally affects bidprice/askprice ??

Ahh yes, i meant like bidsize or asksize 😅not division haha, sorry for the confusion

Sorry, I did not fully understand this, perhaps we can discuss it in the meeting. 😅

8) temp\_df = temp\_df.reindex(unique\_time\_ids).ffill().bfill() ## forward and backward fill the missing values so that data is available at all time\_id

would bfill introduce look ahead bias

So when both ffill() and bfill() are used together like above ffill() is used all the time except for time\_ids that are even earlier than where data is available I have confirmed this with a toy example below. Notice only time id = 1 is backward filled others are all ffill()

Ahh okay thanks!

E,g. df = pd.DataFrame({'t':[2,6,11], 'v':[3,1,4]}).set\_index('t')

df



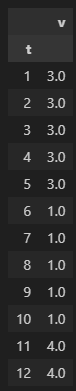
df = pd.DataFrame({'t':[2,6,11], 'v':[3,1,4]}).set\_index('t')

df

u\_t = np.array([1,2,3,4,5,6,7,8,9,10,11,12])

df = df.reindex(u\_t).ffill().bfill()

df



9) Verify that all time\_ids in train.csv and book\_train.parquet match for all the stocks

Yes they match!

Verified in data\_munging.ipynb

9) Verify that all time\_ids in train.csv and trade\_train.parquet match for all the stocks

Following do NOT match!

stock id 18 time ids do not match

missing in train\_st\_time\_ids []

missing in book\_train\_time\_ids [8524]

stock id 31 time ids do not match

missing in train\_st\_time\_ids []

missing in book\_train\_time\_ids [985, 3987, 5539, 5629, 6197, 8753, 8840, 9208, 12011, 13377, 13663, 15010, 20017, 22498, 28186, 32174]

stock id 37 time ids do not match

missing in train\_st\_time\_ids []

missing in book\_train\_time\_ids [62]

stock id 103 time ids do not match

missing in train\_st\_time\_ids []

missing in book\_train\_time\_ids [9664]

Need to use ffill()

Fdsf