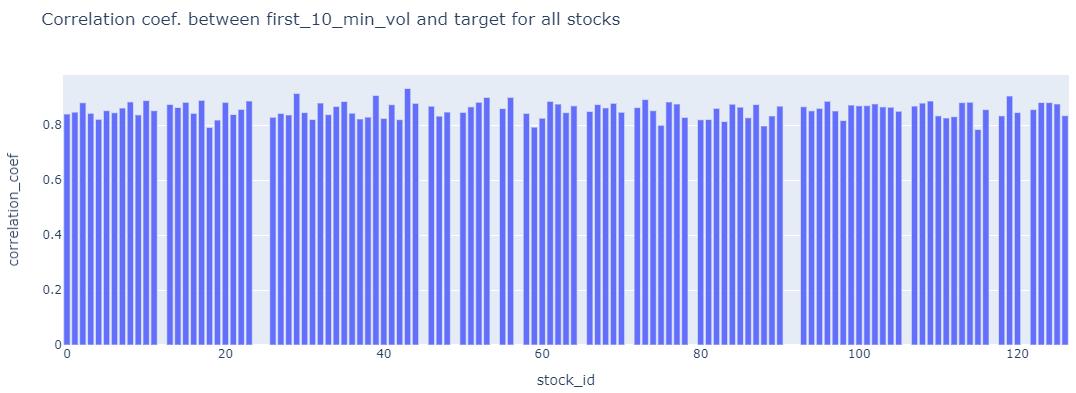
**IDEAS to gain insights**

# **Feature DEFINITION**

##### 

##### **1) Check correlation between target and book WAP realized\_vol in first 10 mins. Aggregate book WAP price using the log returns and realised volatility formulas for each time id. Plot scatterplot this against the corresponding target for all time id.**

yes confirmed!



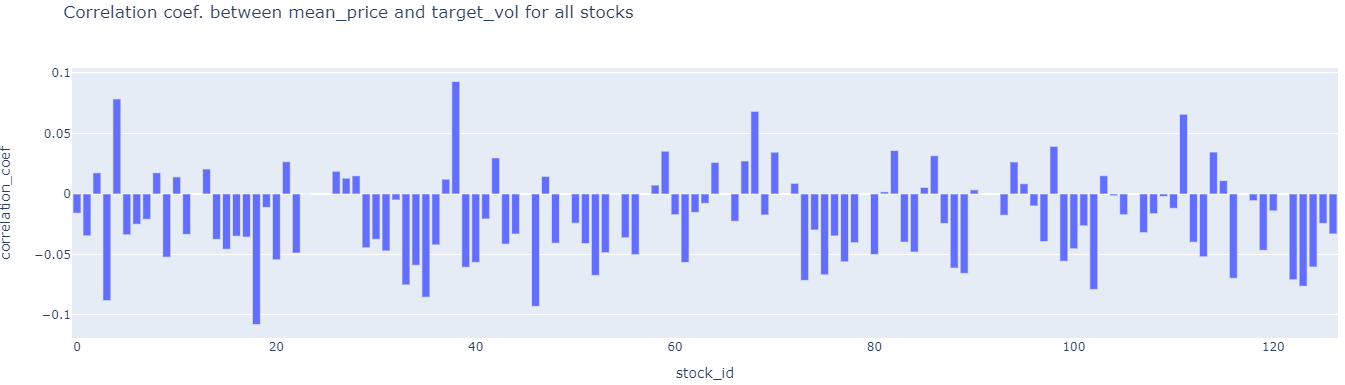
##### **2) Check correlation between**

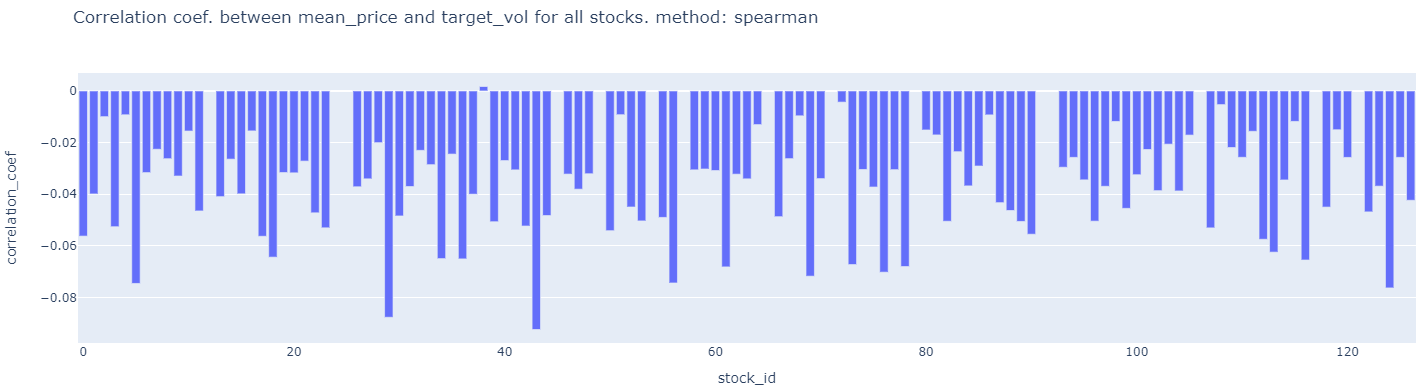
##### **i) target (2nd 10 mins vol.) and trade execution stock price at the available times.**

Aggregate **trade execution price using the log returns and realised volatility formulas for each time id. Plot scatterplot this against the corresponding target for all time id.**

**Yes confirmed**

****

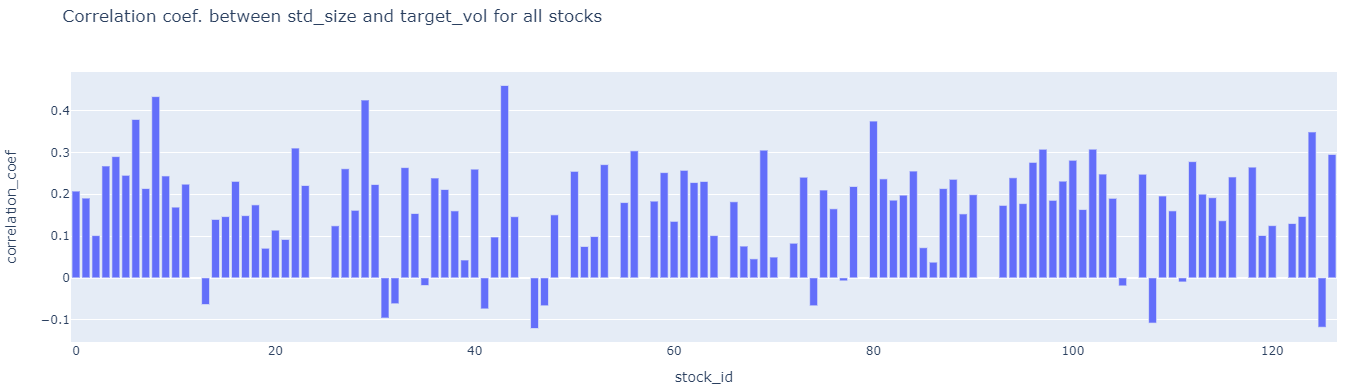
****

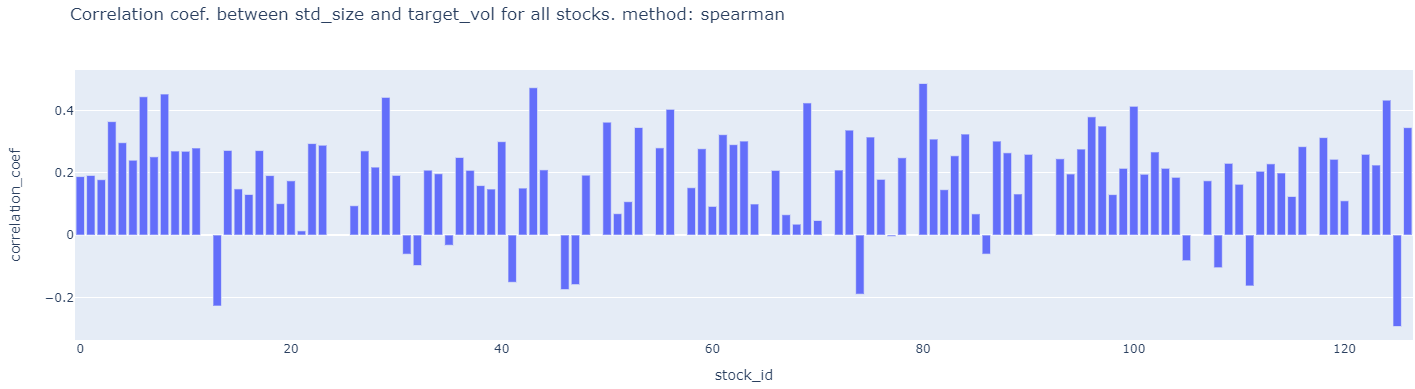
****

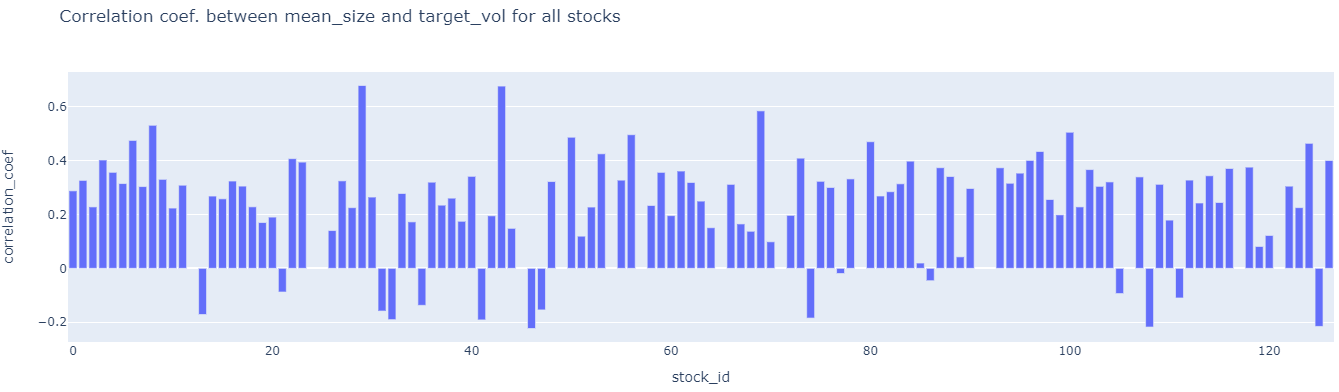
****

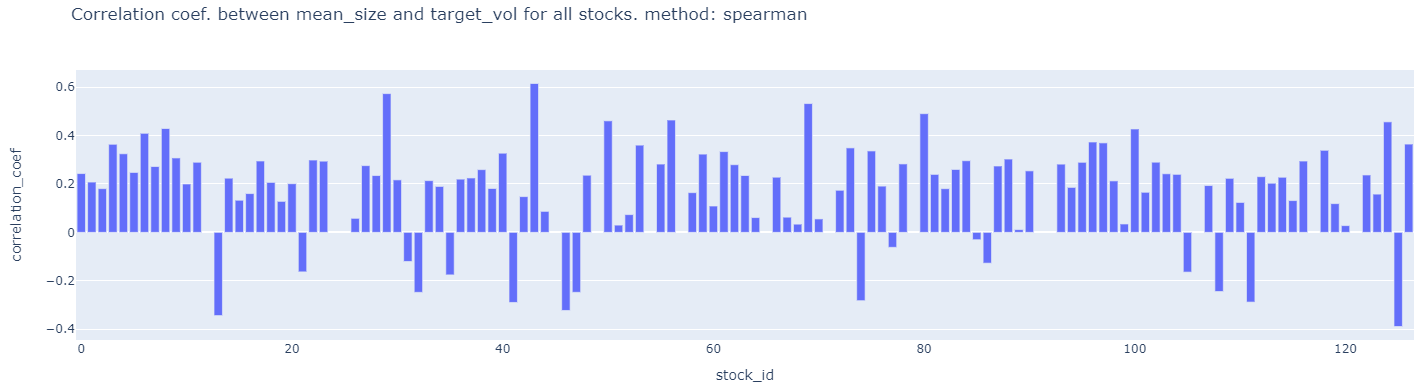
##### **ii) target (2nd 10 mins vol.) and size at the available times.**

Aggregate size by taking standard deviation or mean at each time id. **Plot scatterplot of this against the corresponding target for each time id.**

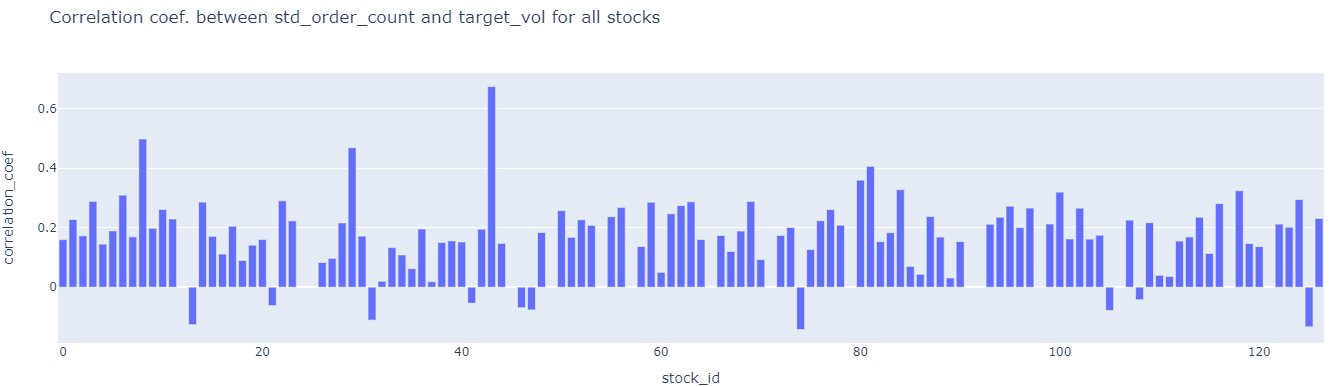
****

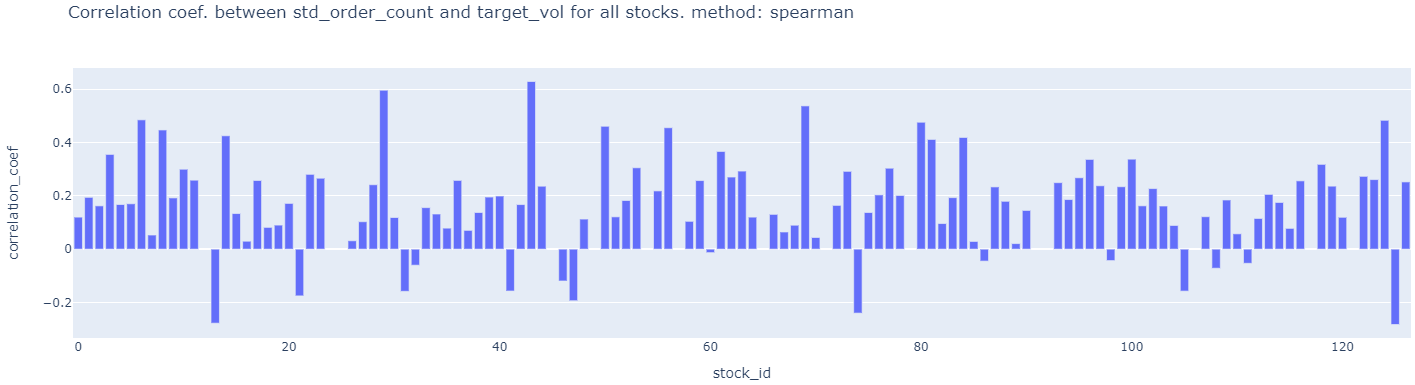
****

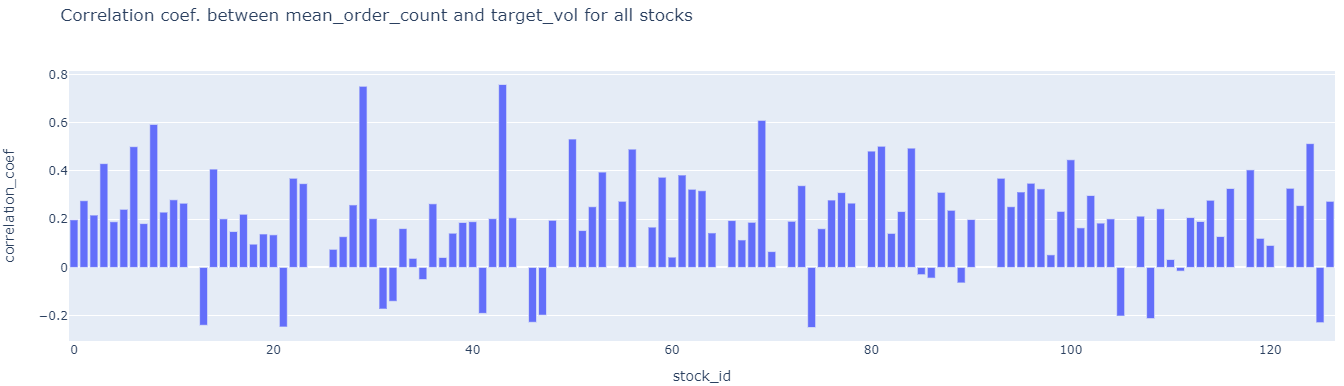
****

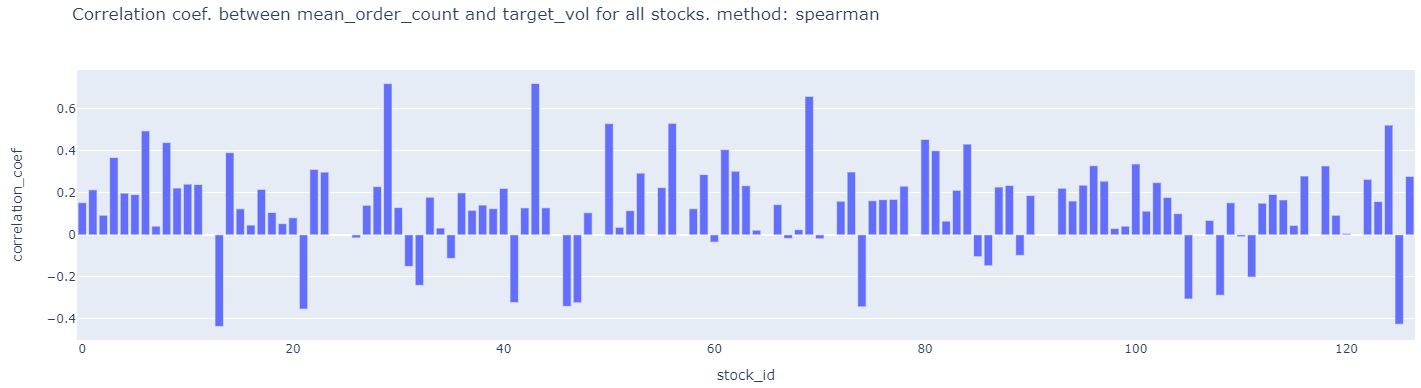
****

##### **ii) target (2nd 10 mins vol.) and order\_count at the available times.** Aggregate **order\_count** by taking standard deviation or mean at each time id. **Plot scatterplot of this against the corresponding target for each time id.**







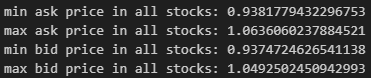


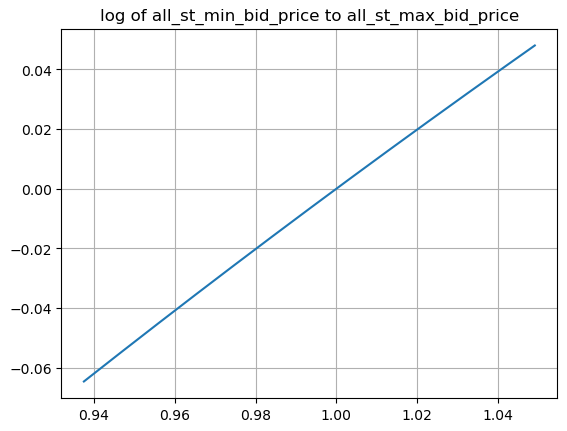
##### **2a)** Try to calculate spearman’s rank correlation for the cases of low linear pearson’s correlation. It might be higher due to non-linear correlation??? **DONE ABOVE. VERY SIMILAR.**

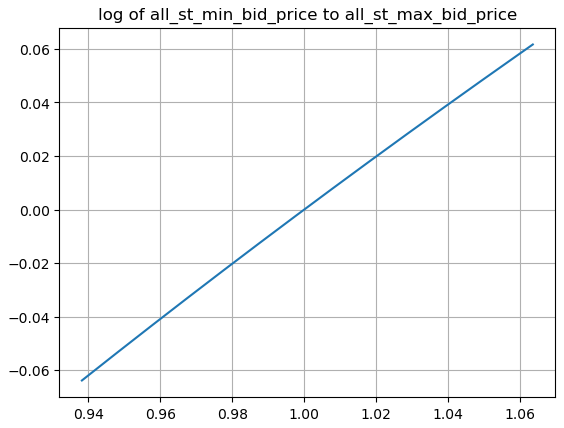
##### **3) check correlation in the first 10 mins. Target volatility between any two stocks and then identify which stocks have highest correlation. This is possible because even though time\_id is shuffled, they are the same for each stock so the scatter plot does not care about time. 112 choose 2 = 6216. compute this numerically. DO NOT PLOT. Rank the correlations.**

**DONE and updated in Key Insights file.**

4) ## check how prices are distributed in the book data so that we can see how log(s\_t2/s\_t1) transformation of prices below and above 0 affects volatility. ## due to log nature check if non-linearity is visible. A point in graph below is for a single stock.



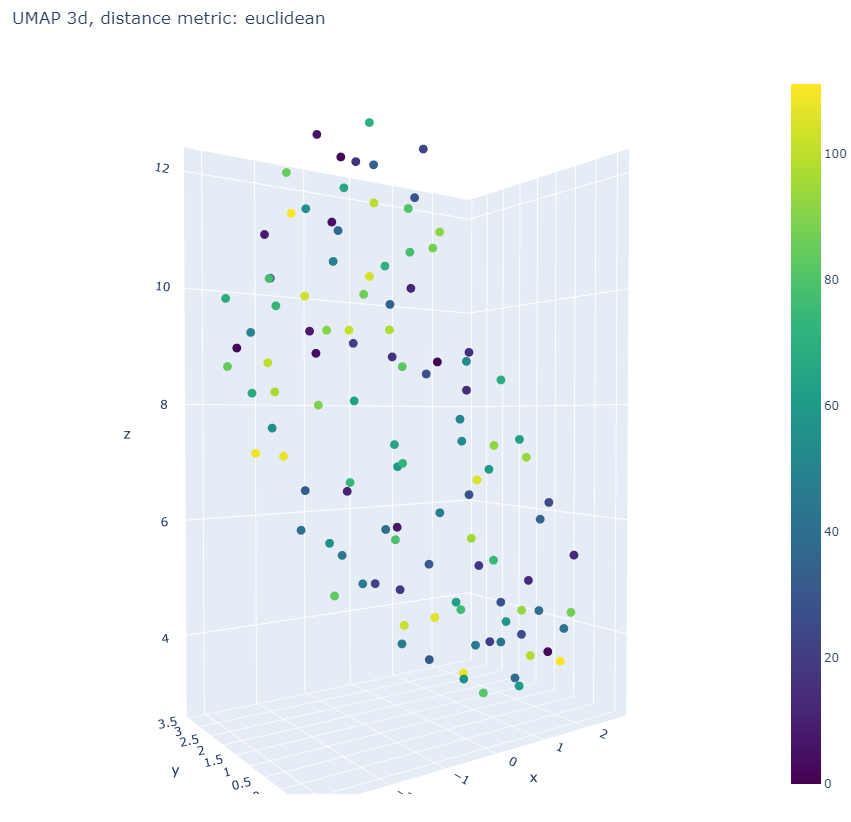




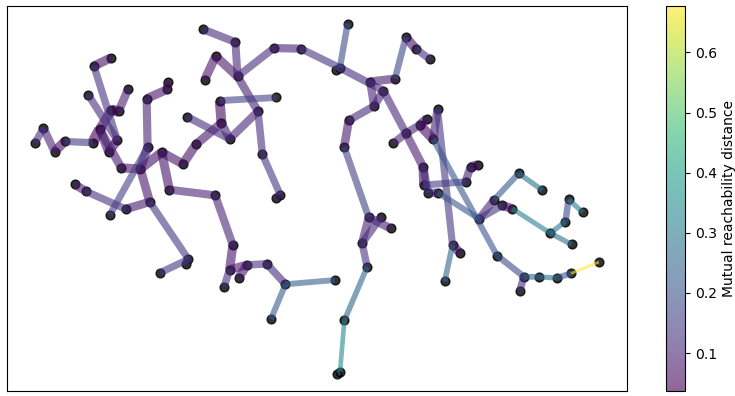
## Non-linearity is NOT visible as variance in price is low. so NO need to treat prices < 1 differently from prices > 1.

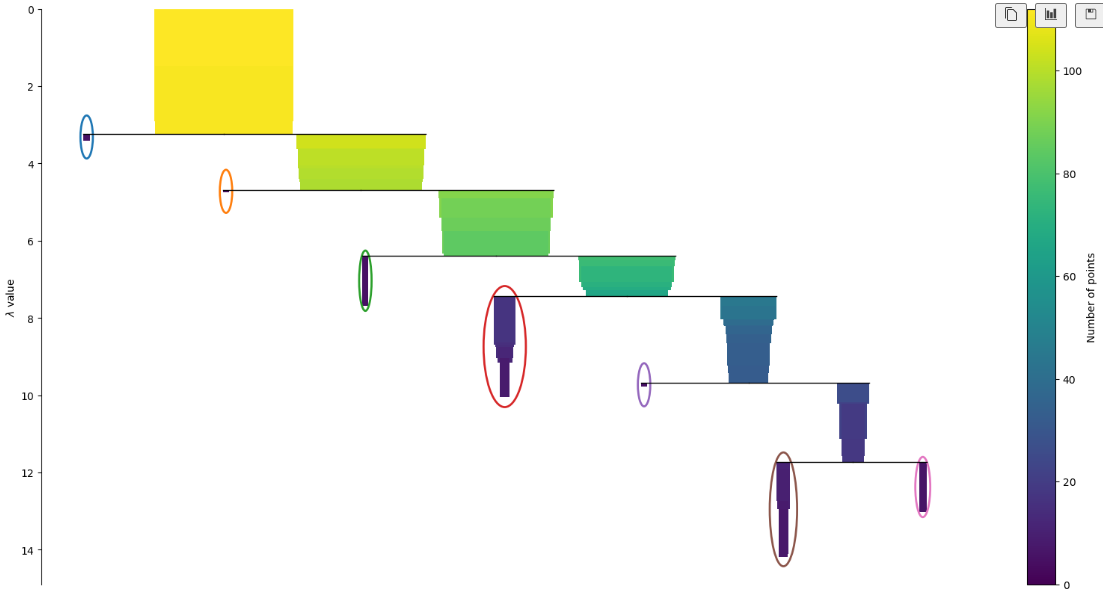
5) perform target volatility clustering across stocks using summary statistics features like mean, median, min, max etc..

Ans: The stocks volatilities are so homogeneous that it's difficult to separate/cluster them using summary stats features and hdbscan clustering algorithm. Maybe better features and/or different clustering algo. Might work. This is done in target\_eda\_across\_stocks.ipynb file

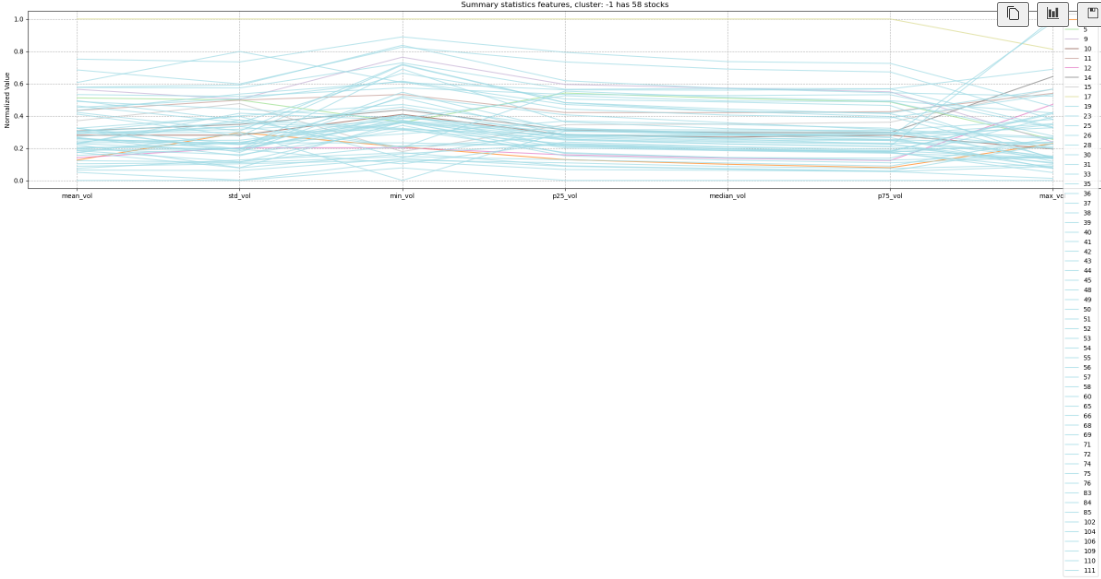
Colors represent different stocks. 

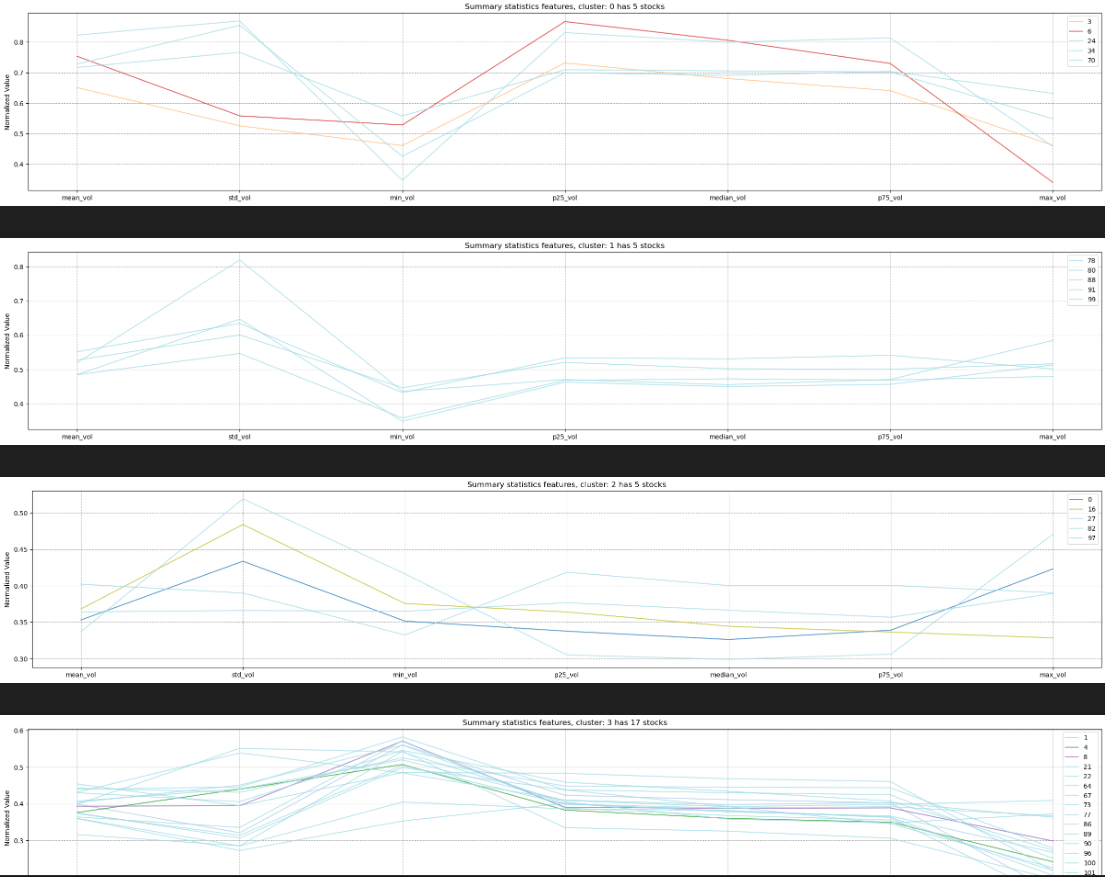
HDBSCAN analysis below was NOT fruitful!! .

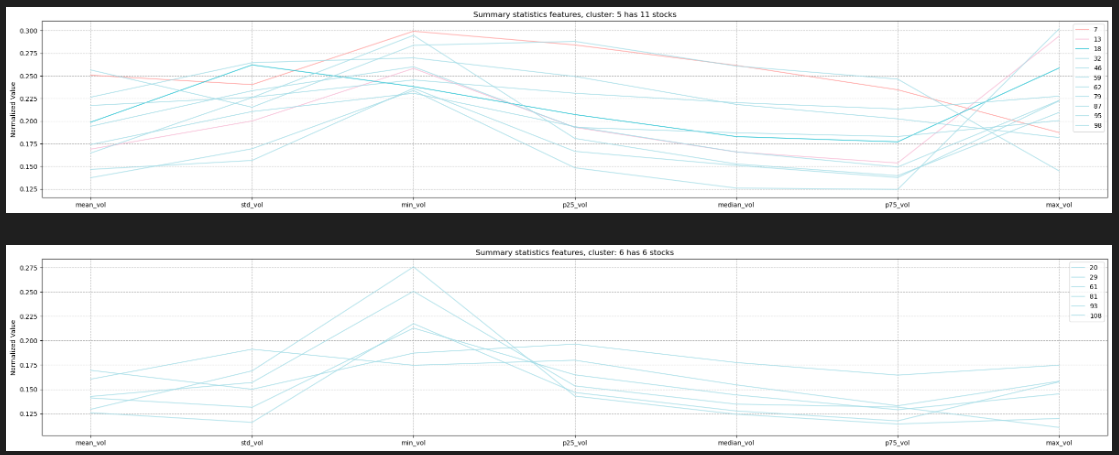




There are too many outliers in this clustering method.

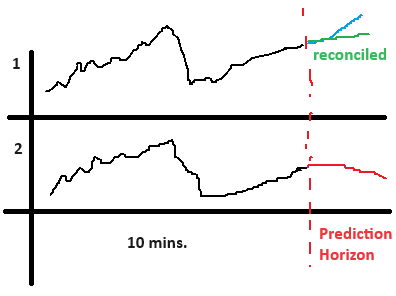






Hierarchical Agglomerative clustering using WARD linkage is better.

6) If two stocks (i.e. time series within a time\_id i.e. 10 min interval) are highly correlated for many time\_ids then we can improve the independent forecast of one stock by using the independent prediction of the correlated other stock. Reconcile the two predictions for the first stock to get more accurate forecast for 1st stock. Correlation captures shared patterns between the two stocks while independent stock prediction is based solely on within stock patterns. Do similarly for stock 2 using stock 1’s prediction. The other stock is acting like a covariate.



This analysis is SAME as in suggestion 3) which is **DONE and updated in Key Insights file.** Although data analysed is from TARGET and NOT from input features i.e. book\_train and trade\_train data. It’s better still.

7a) Check if minimum/maximum of bidsize1 and asksize1 in a time\_id correlated with target realized volatitlity for the same time\_id?

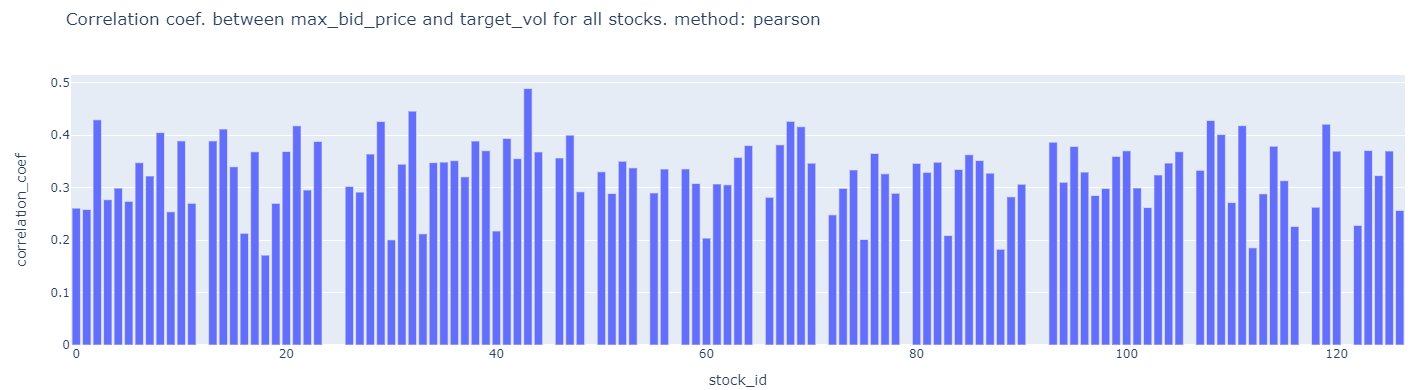
Graphs without level are level 1

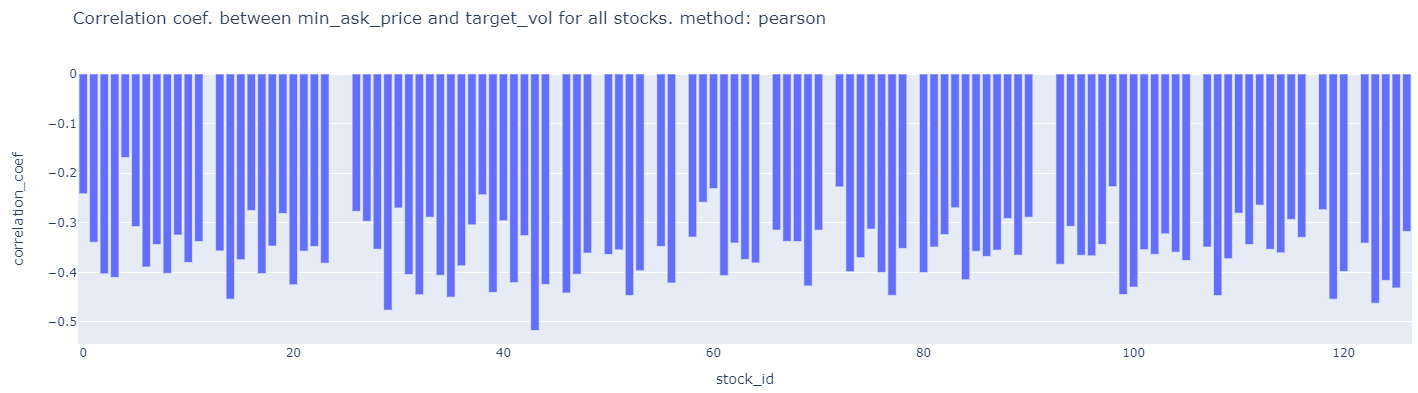


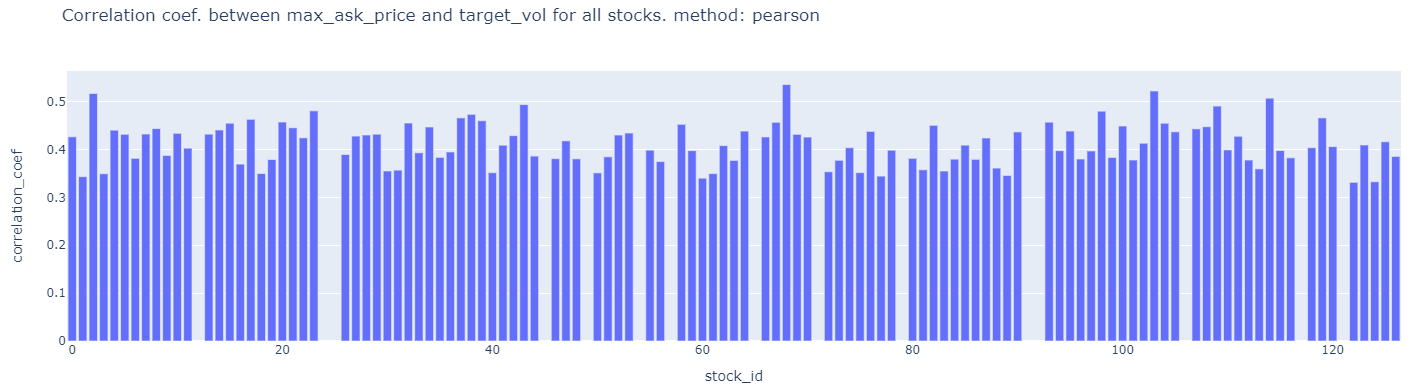


7b) Check if minimum/maximum of bidprice1 and askprice1 is correlated with target realized volatitlity for the same time\_id?





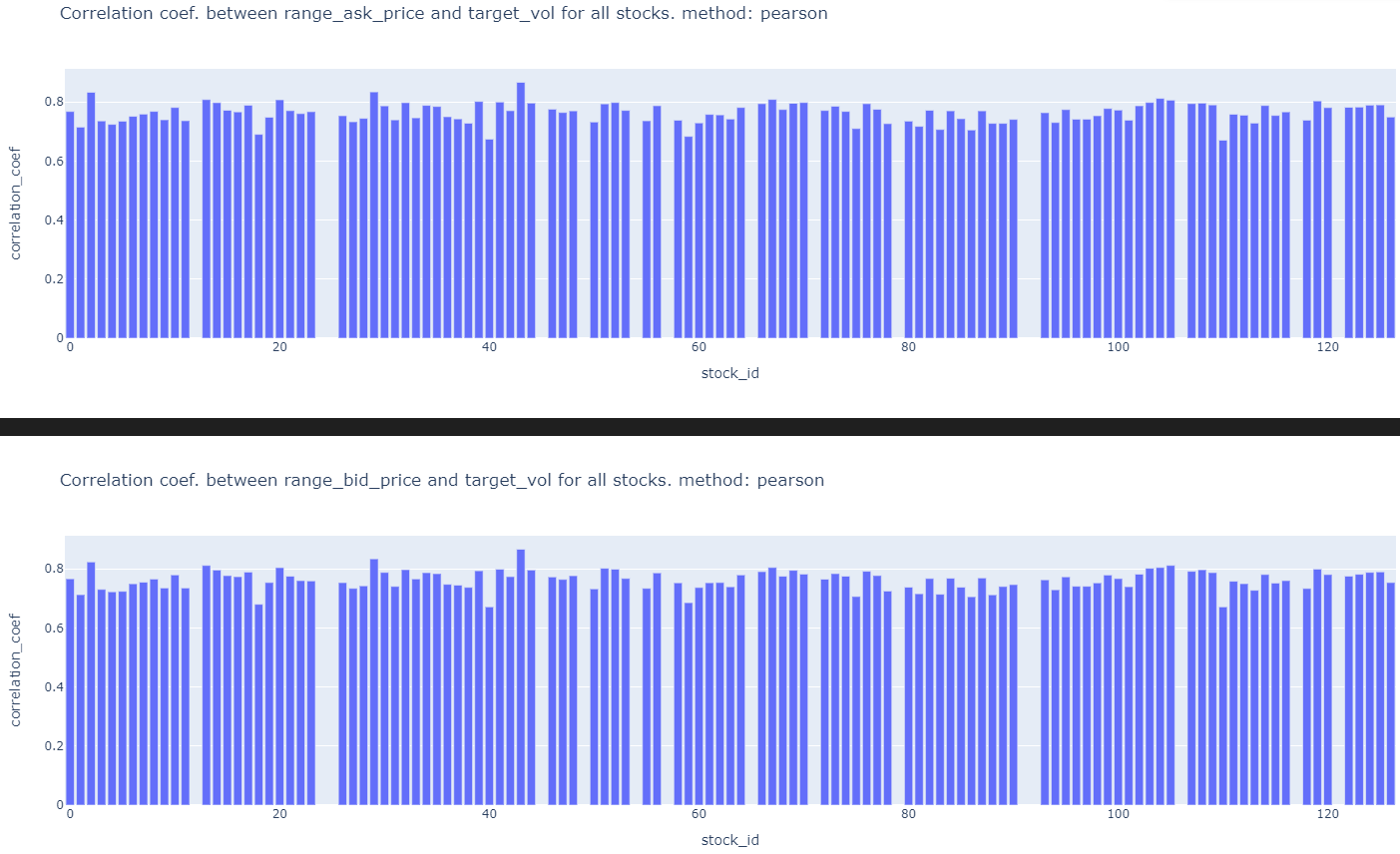




7c) check if the difference between minimum and maximum (i.e. range) of bidsize1 and range of asksize1 is correlated with target realized volatitlity for the same time\_id?

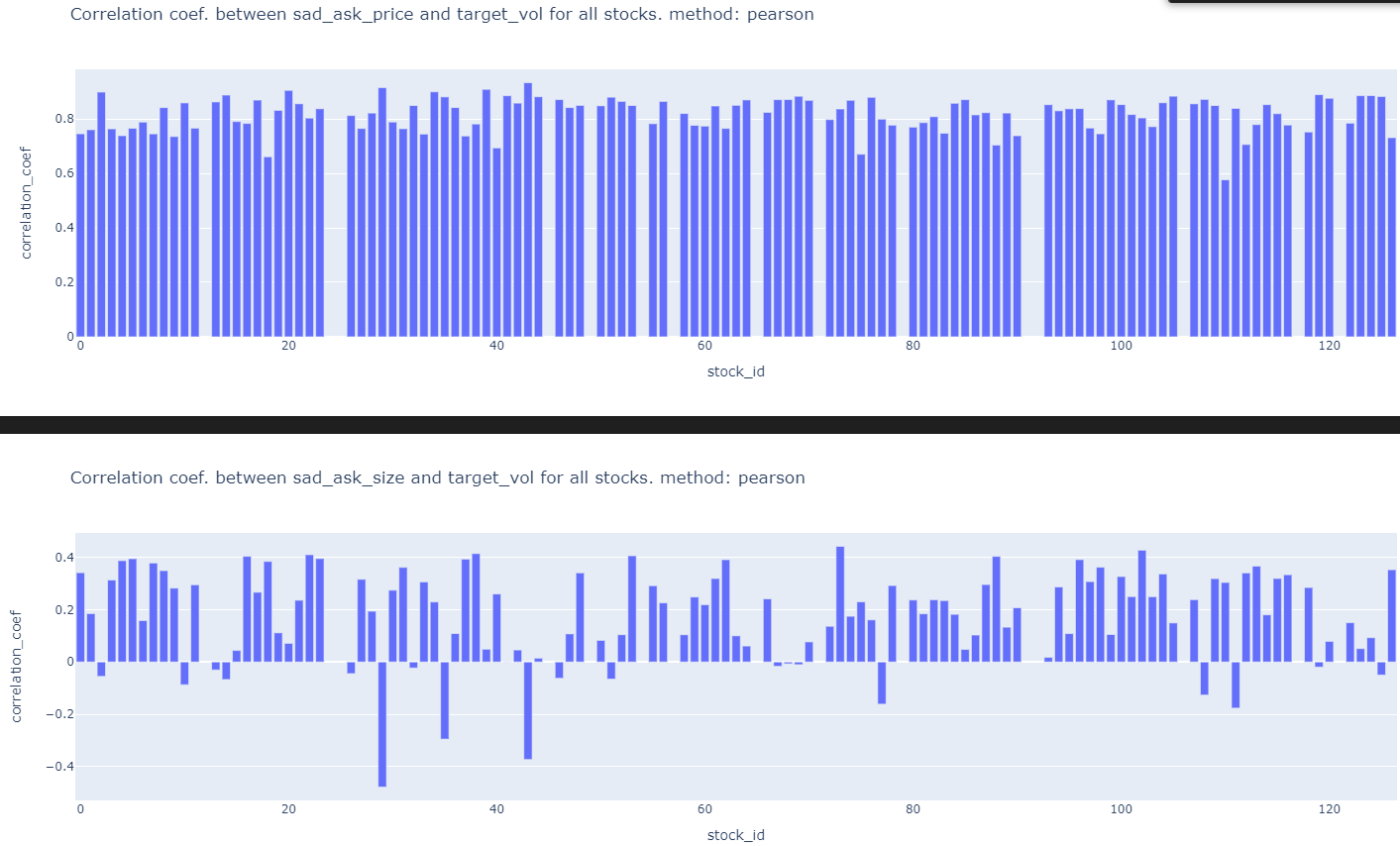


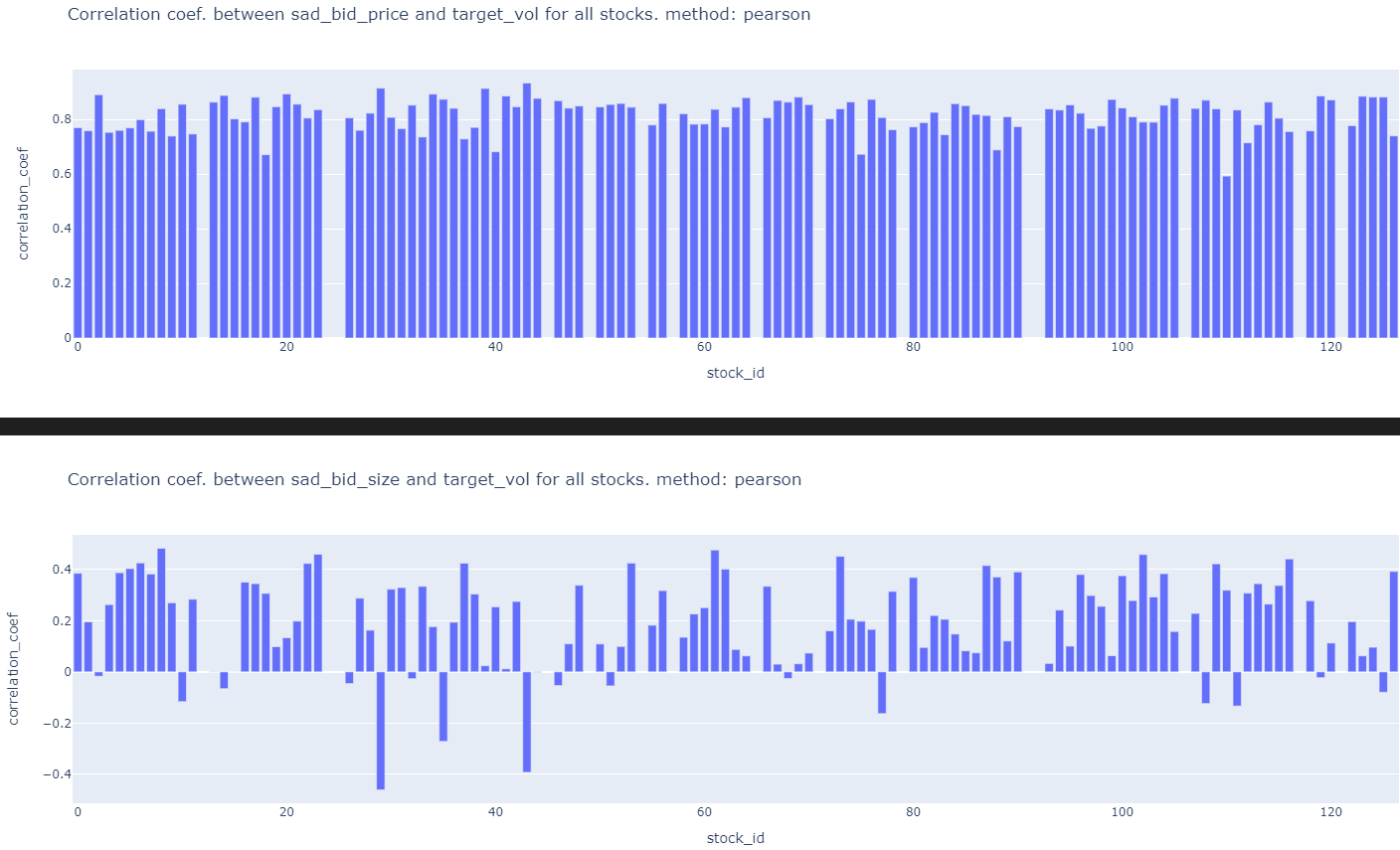
7d) check if the difference between minimum and maximum (i.e. range) of bidprice1 and range of askprice1 is correlated with target realized volatitlity for the same time\_id?



7e)Also check the correlation between (minimum – maximum) (i.e. range) vs. realized volatitliy of target. WHAT??

7f) Also check if the ~~average~~ sum of absolute bid\_price1 in t2 minus bid\_ price1 in t1 within a time\_id is correlated with target. Check the same for ask\_price1. Sum of absolute differences is better than average because price changes are too small and there are 600 seconds in a bucket so average values will be too small.





7g) calculate correlation between bid\_price1 and ask\_price1 in a time\_id (positive correlation leads to a larger/smaller wap’s numerator) then check correlation with target realized voaltitlity.

7h) calculate correlation between bid\_size1 and ask\_size1 in a time\_id (positive correlation leads to a larger/smaller wap’s denominator)

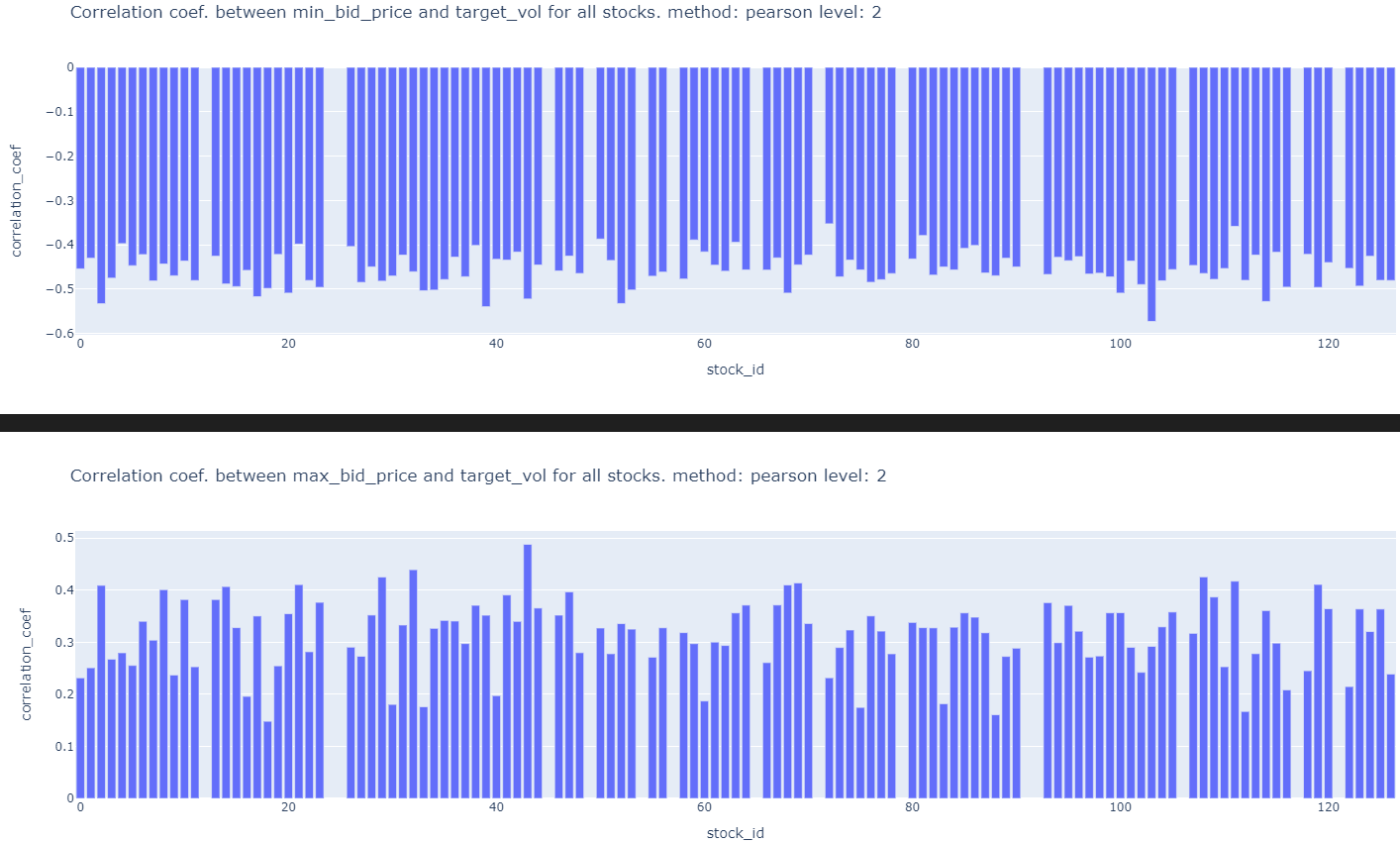
7i) Calculate pairs of correlation between a time series in this list [bid\_price1, ask\_price1] and this list [bid\_size1, ask\_size1] in a time\_id (negative correlation leads to a larger/smaller wap) i.e. corr(bid\_price1, ask\_size1), corr(bid\_price1, bid\_size1), corr(ask\_price1, bid\_size1), corr(ask\_price1, bid\_size1)

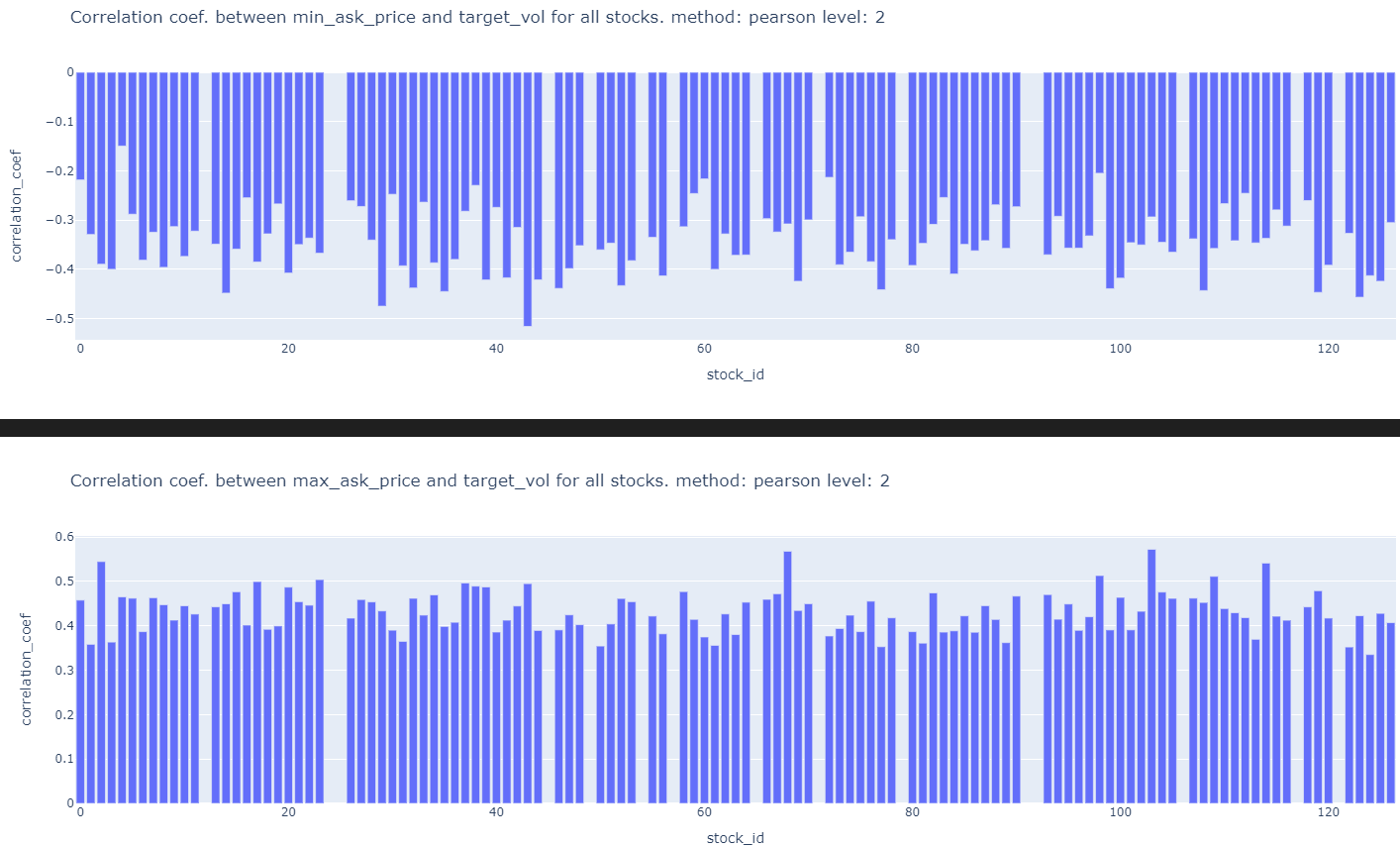






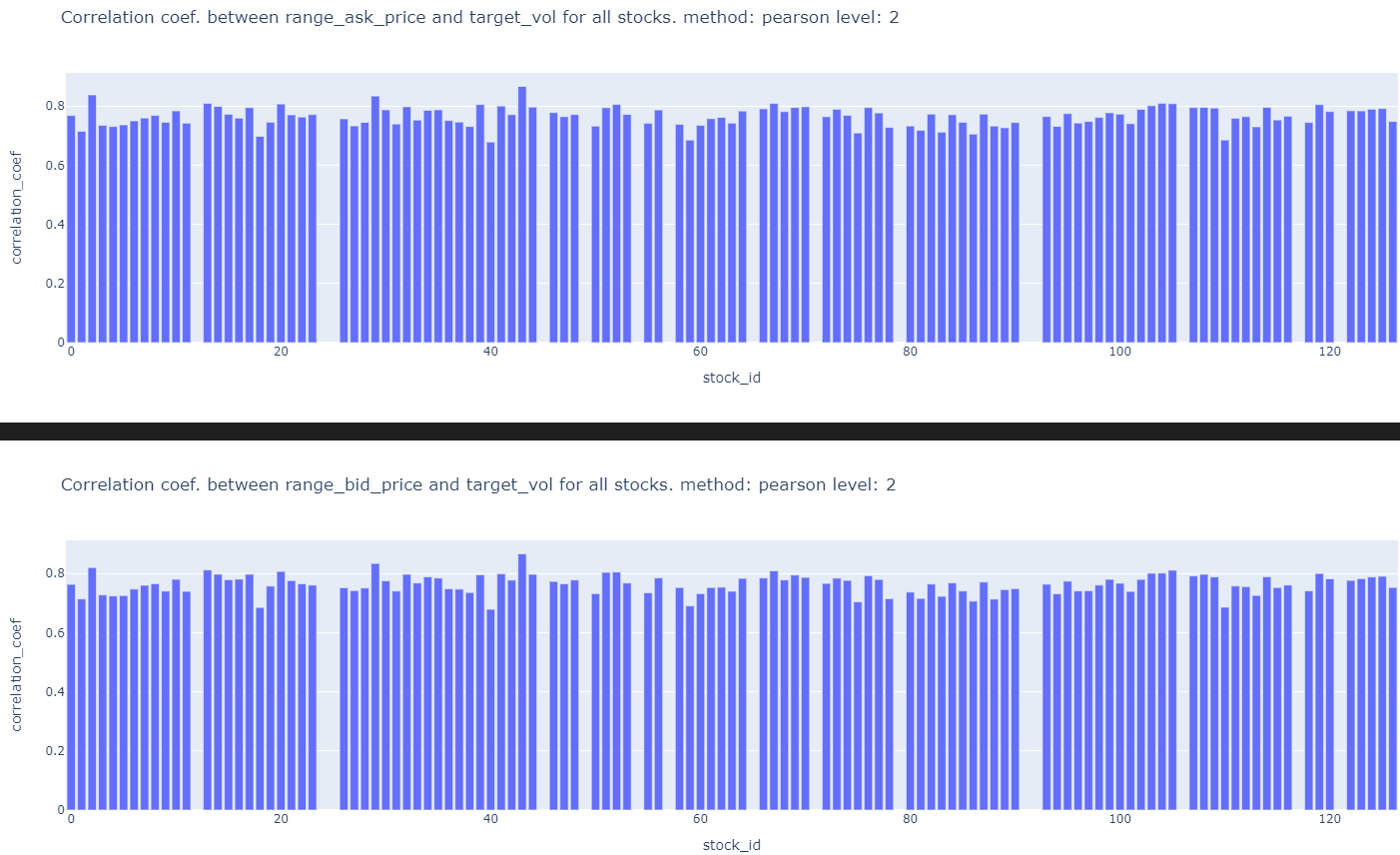
7j) Do the above for bid/ask\_price2, bid/ask\_size2 as well.

















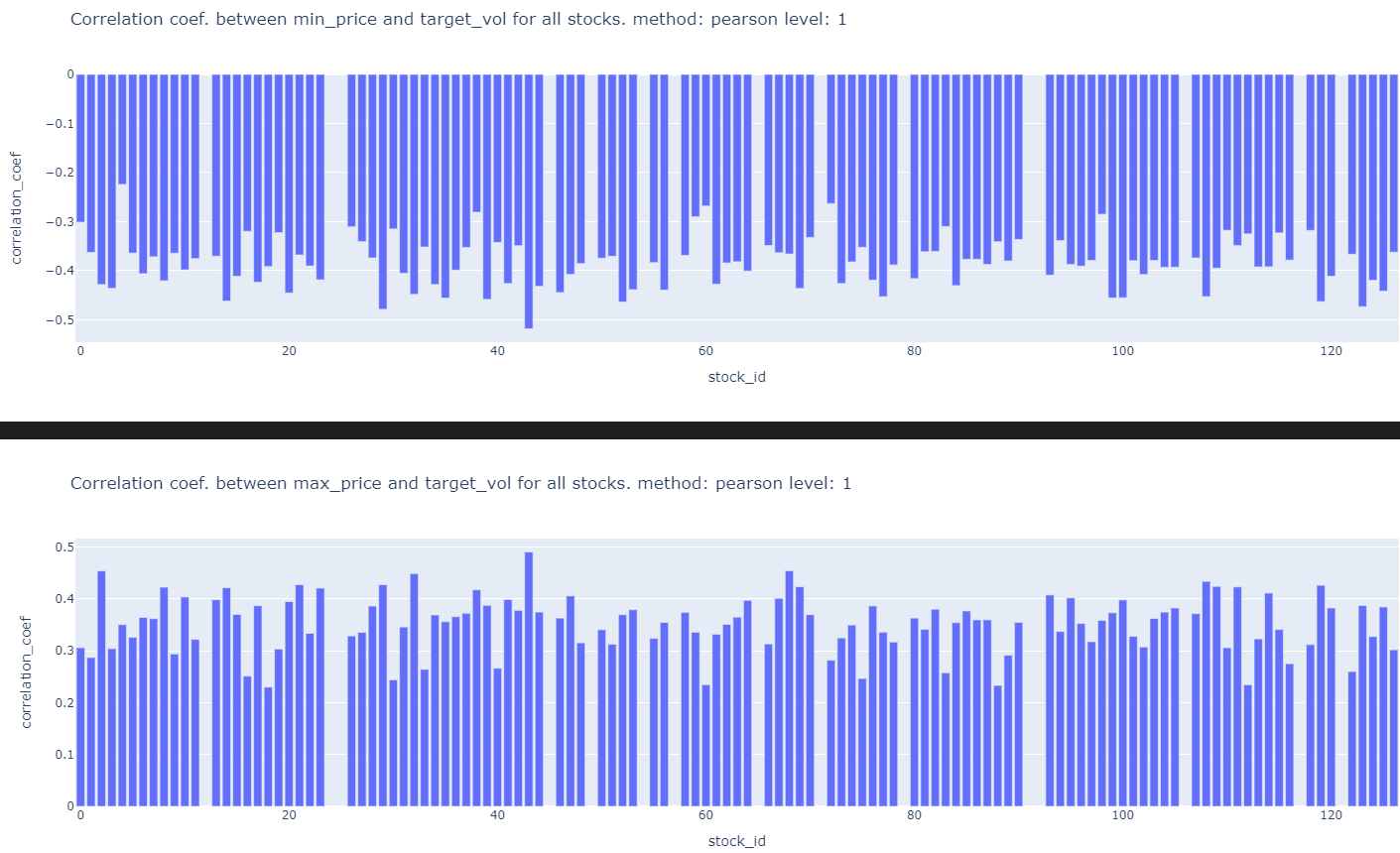


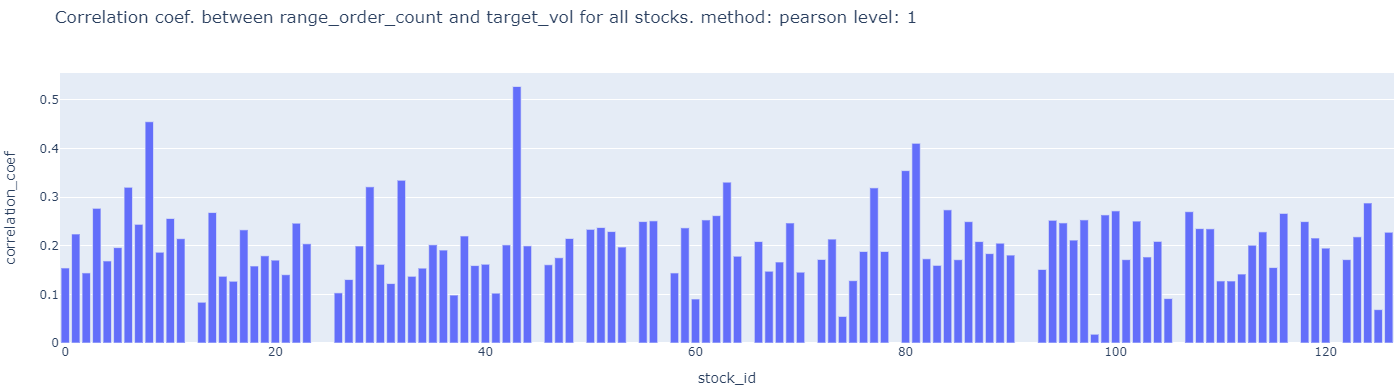
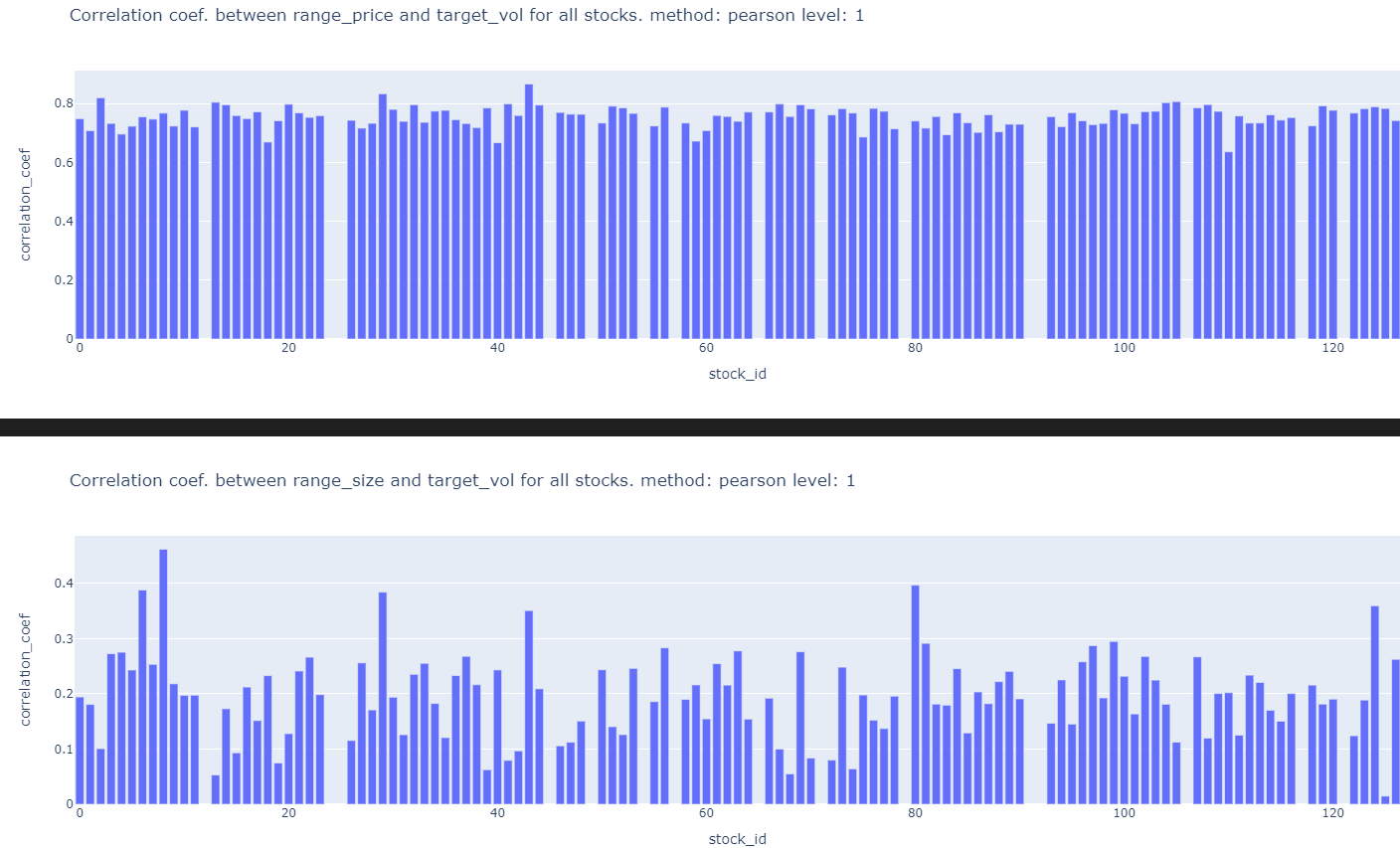


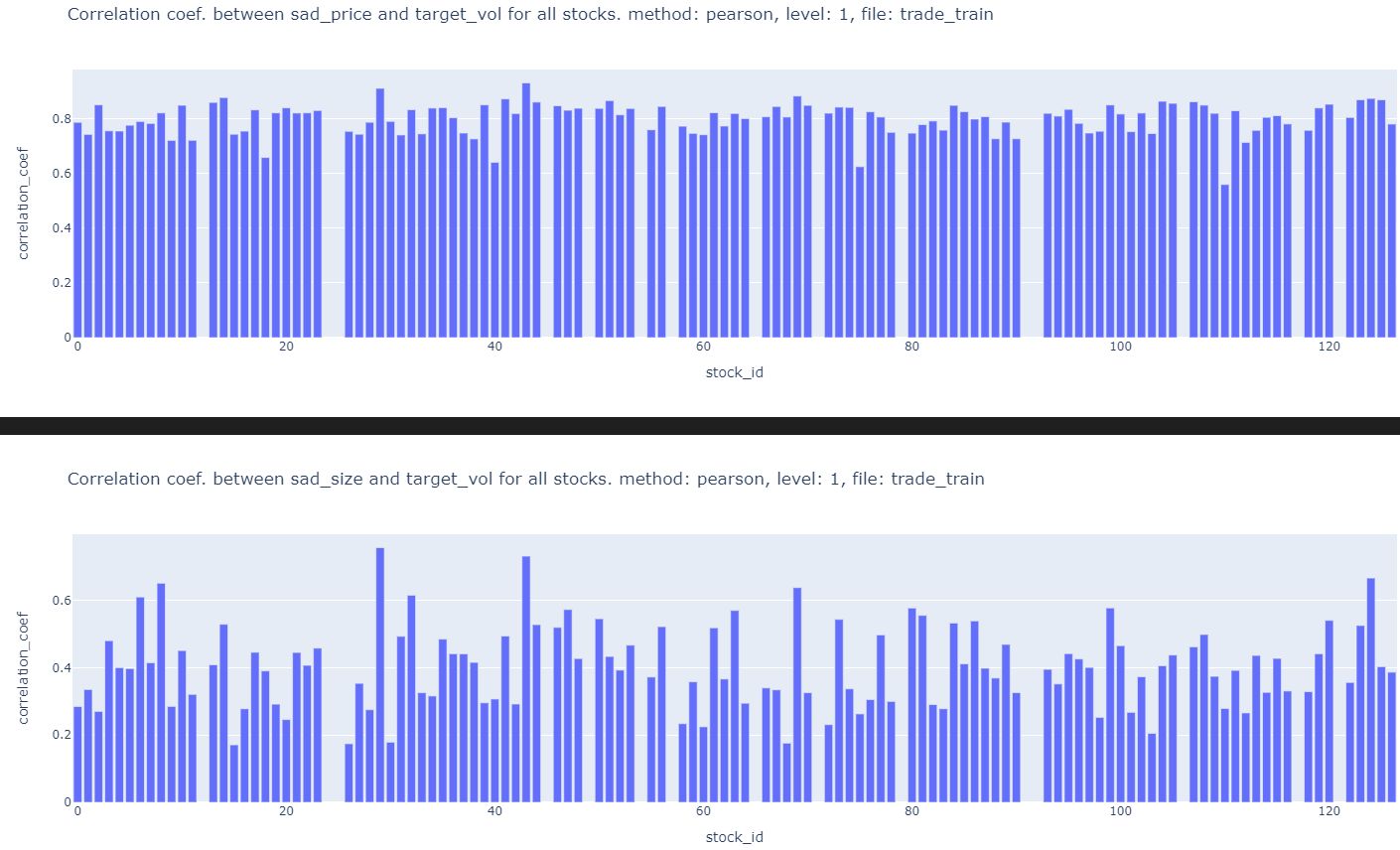


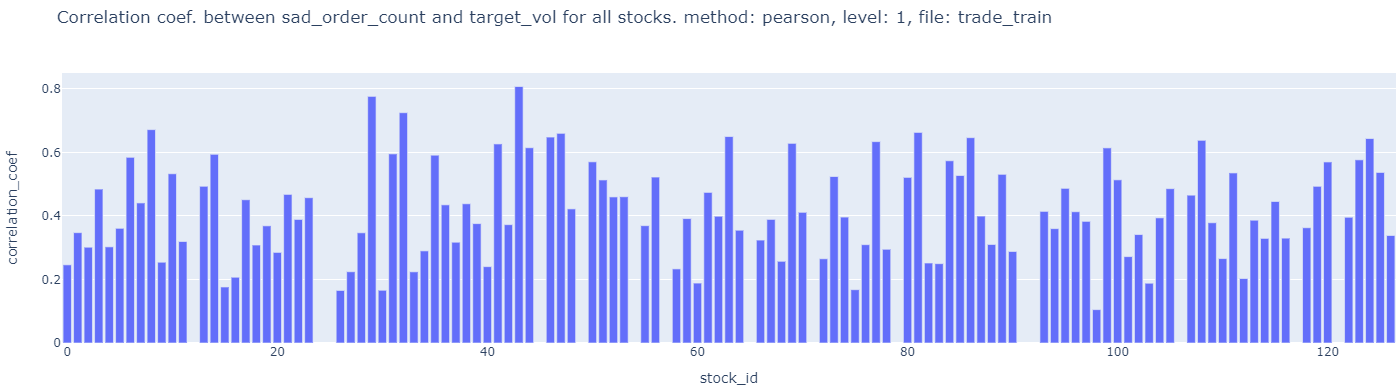
7k) Do the above for trade\_train.parquet

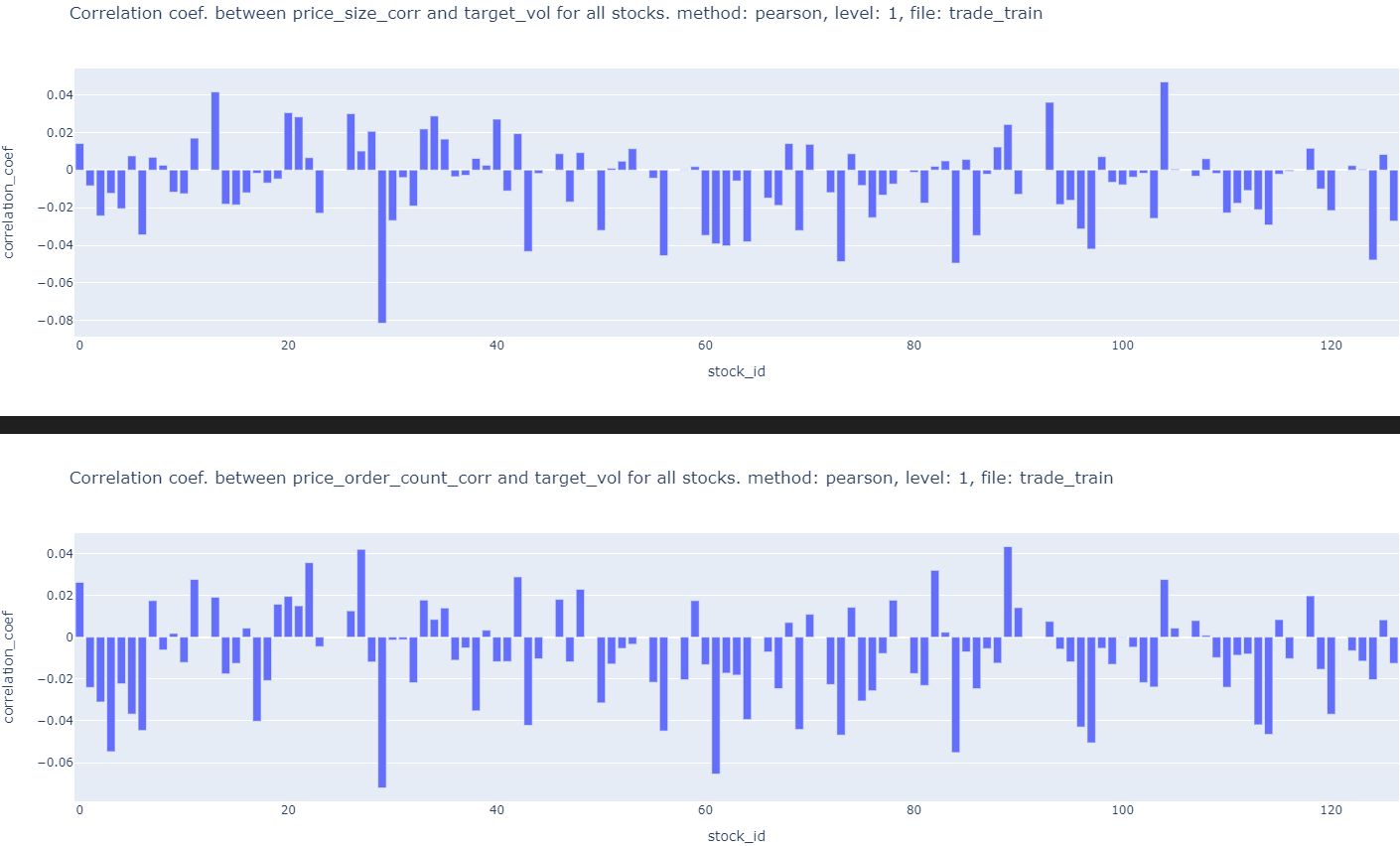
Slight code modification has to be done.









￼

This is because from the WAP formula we can see that there is large variation in wap time series caused by min and max values of bidprice1 and askprice1 bidsize1 and asksize1 respectively. This affects returns and volatility. ALL of suggestion 7 is done in features\_eda\_within\_stocks.ipynb

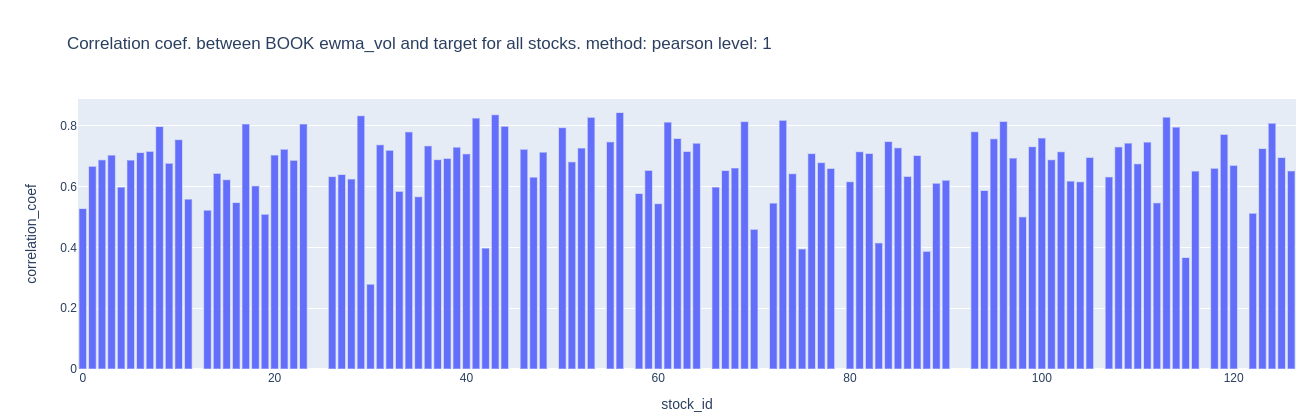
8) What features in the first 10 min that affect volatility in the next 10 mins?

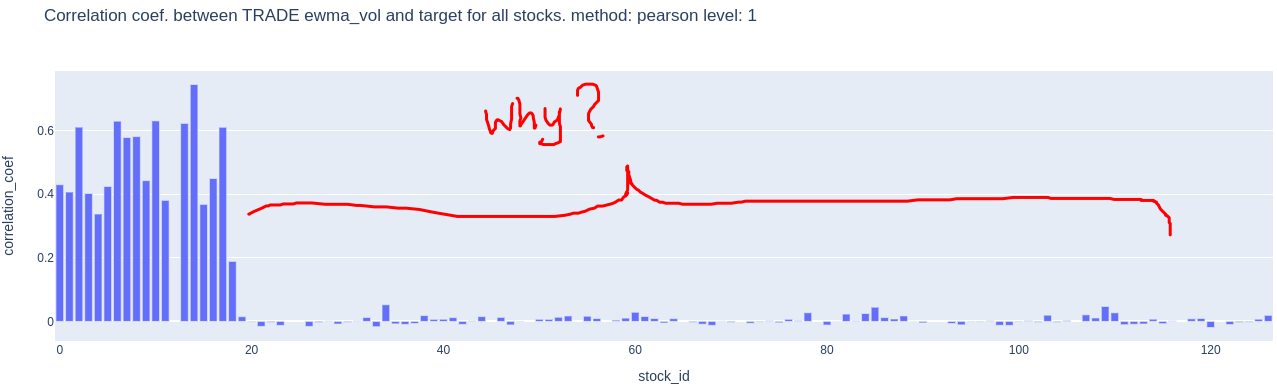
Feature that has Delayed effect on volatility?

Ans: Rolling average or moving average volatility can identify a lasting trend of high or low volatility. (peeterson)

This is not good because summed volatility is an increasing function

Exp. moving average of INstantaneous volatility (squared returns) can be used





9) Use a GARCH model on the time series available in the first 10 mins. to predict volatility into X (X<10) minutes of the second 10 minutes. Check correlation for different average x minutes realized volatility against the target realized volatility and choose the best x minutes of average volatility for each stock. (peeterson)

9a) The cons of a GARCH model is that the GARCH model does not distinguish between the impact of positive and negative negative returns. However, the leverage effect states negative returns have a greater impact on volatility and volatility clustering than positive returns.Therefore use GJR-GARCH or EGARCH.

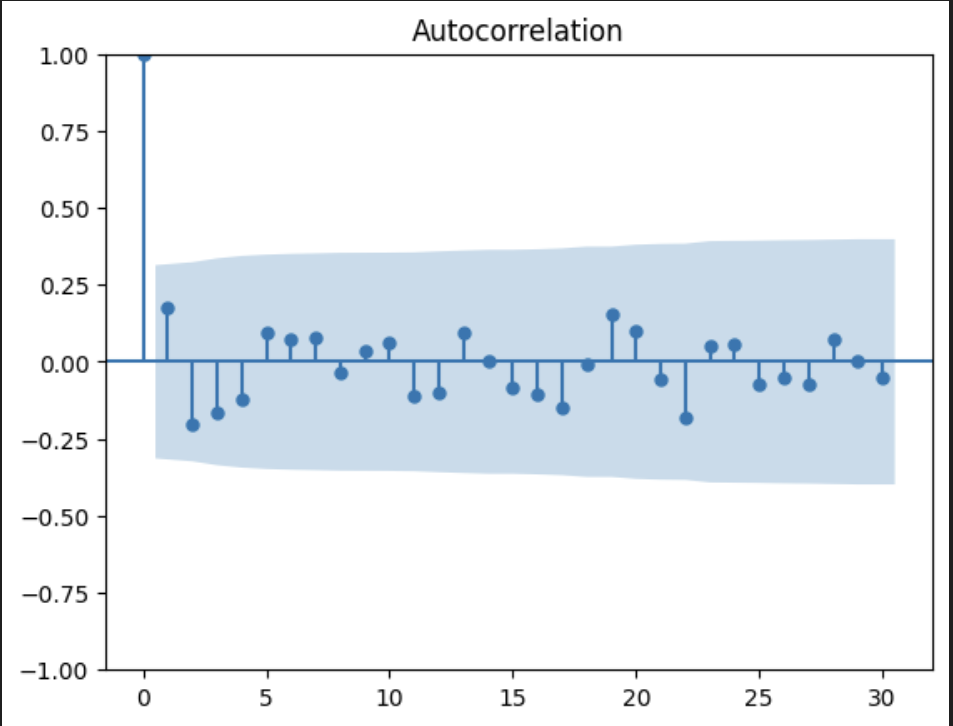
Use Apache spark to analyse 166 Million rows.

9b) use the clustering (of correlation matrix) done on pearson correlation of target in fitting a multivariate GARCH model.

9c) Can you reverse engineered time\_ids to create more contiguous trainings set and together with time\_id can be used in prediction.

https://www.kaggle.com/c/optiv￼er-realized-volatility-prediction/discussion/275825

10) Plot ACF of squared returns time series to leverage the Volatility clustering (persistence) phenomenon to predict/extend into second 10 minutes. Remember volatility is square root of sum of squared returns.(jx)



All autocorrelation values are under blue range so they are all close to zero

Series is random?

Yes, so it’s random or coming from a white noise distribution. We cannot predict instantaneous values but we can learn a probability distribution from these returns and predict things like averages, standard deviations and long term behaviours

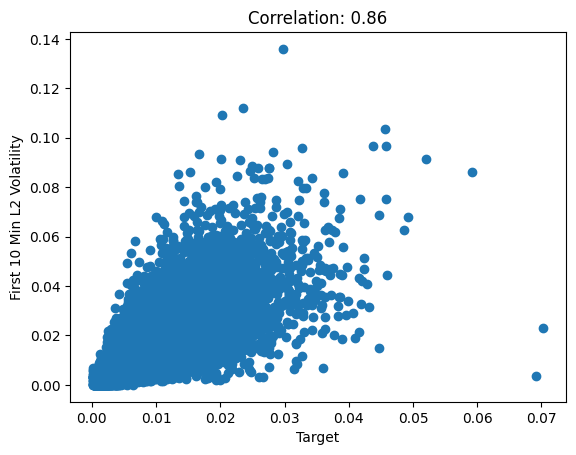
Ohh

~~11) WAP price trend up/down in the second 10 mins is negatively correlated with realized volatility in the 2nd 10 mins. This is called the leverage effect. Can we forecast trend in the 2nd 10 mins? Try using moving average of WAP of different window sizes and check if current price is above or below average. to predict short term trend in 2nd 10 mins.~~

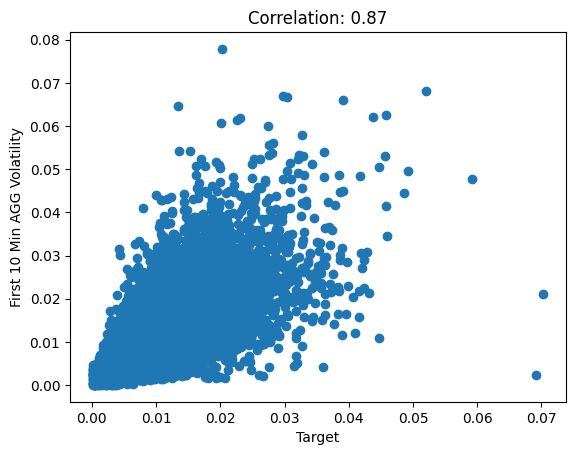
12) When calculating realized volatility did you first ffill for the missing book order seconds?

It does not make a difference because constant price/size/order\_count lead to zero volatility.

13) Compute Wap2 using ask/bid\_price/size 2.find the correlation of realized vol. Using level 2 wap with target and find the correlation of realized vol. Using average wap with target. Code is already available in features\_eda\_within\_stocks.ipynb in cells 1, 2, 225, 230, (jiaxu)

￼

Correlation for wap2 and target

￼

Correlation for aggregate wap and target

aggregated wap calculated using ((bidprice1 \* asksize1) + (bidprice2 \* asksize2) + (askprice1 \* bizsize1) + (askprice2 \* bizsize2)) / (bidsize1 + asksize1 + bidsize2 + asksize2)

14) Just use skew, kurtosis, min, max, std,and all other statistics for each bid ask wap etc… in each time id. (peeterson)

15) Check if Standard Deviations of Price Variables is POSITIVELY Highly Correlated with Target (peeterson)

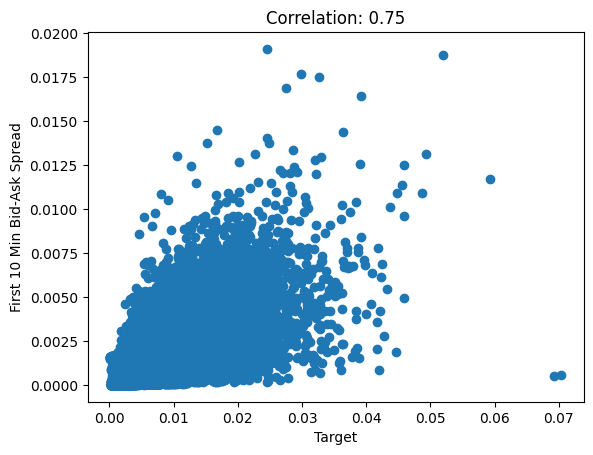
16)Check if Minimums of Prices Variables is NEGATIVELY Highly Correlated with Target; (peeterson)

17) the std of bid price was often more highly correlated with the target than the wap calculation. (peeterson)

18) Use bid ask spread. ask\_price1 / bid\_price1 - 1 (jiaxu)

(ask-bid)/bid

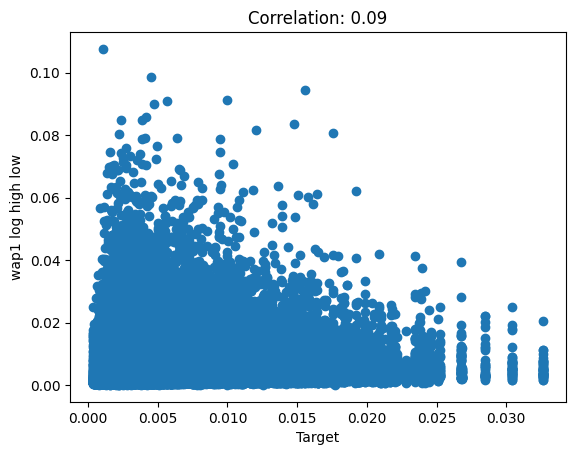
￼



Used mean to aggregate bid ask spread for each row\_id

19)wap1\_log\_high\_low - this is log(highest wap1 in interval) - log(lowest wap1 in interval)

Implement above feature (jiaxu)



20) Liquidity features:

Spread-related measures, Volume-based measures , Price-based measures

<https://www.kaggle.com/competitions/optiver-realized-volatility-prediction/discussion/276137> and Chatgpt 🙂

LIquidity describes how much of an asset can be purchased or sold immediately without moving the market (i.e. significant price changes). It is inversely proportional to volatility and bid-ask spread. It is directly proportional to trade volume/ size.

Generally, trades happen smoothly when the trade execution price (i.e. let’s call it “realized” WAP) is a compromise between the bid and ask price i.e. it lies in between to keep both the buying and selling party happy. The bid and ask price should approach this WAP price with time so that trades can happen at this price. This is like price discovery (equilibrium price). In this equation we are measuring how far away we are from this to happen. incorporating the order size also improves this information and this is defined as liquidity. Current time’s liquidity is an indicator of the future volatility to come. WAP is calculated all the time but “realized” WAP is when trade execution happens at that time.

First we need to find the wap (i.e. equilibrium price) for a given bid price, ask price, bid size and ask size. This can be found when we minimize the liquidity function. MInimizing liquidity does not mean liquidity is low value. E.g. 1000 ican be a minimum of a function similarly 10 can be a minimum of a function. Value of 10 is lower liquidity than 1000. The minimum is the correctly measured value of liquidity at that point in time. At different times we calculate different liquidity. The computed objective function (at the minimum) is computing liquidity based on parameters of the objective function. MInimizing this liquidity objective function helps us find the WAP that the bid\_price and ask\_price is likely to move towards in the future.

Given bid\_size, bid\_price, ask\_size and ask\_price, let us try to calculate the “realized” WAP at which trade execution is likely to happen in the future.

If WAP is far away from bid\_price/ask\_price then it indicating the bid ask spread is high. This is not ideal for trades or we can say it is ILLIQUID. Volatility has to come in the future and move prices in order to make it more LIQUID.

## **Defining Liquidity**

In finance, liquidity describes how much of an asset can be purchased or sold immediately without moving the price significantly. A reasonable numerical definition of liquidity then should start from the following assumptions:

liq\_1 = sum\_i[ bid\_size\_i/(wap\_1 - bid\_price\_i) + ask\_size\_i/(ask\_price\_i - wap1)]

or for wap\_2:

liq\_2 = sum\_i[ bid\_size\_i/(wap\_2 - bid\_price\_i)\*\*2 + ask\_size\_i/(ask\_price\_2 - wap1)\*\*2]

1. moving bids closer to asks always increases liquidity
   1. I.e. smaller (wap\_1 - bid\_price\_i)and smaller (ask\_price\_i - wap1)
2. increasing order sizes always increases liquidity
   1. I.e. Bigger bid\_size\_i and Bigger ask\_size\_i
3. adding additional orders to the book always increases liquidity
   1. I.e. sum\_i Refers to levels of the order book.

The above two behaviours are captured in the following equation as well.

f(x) = -bid\_size\*log(x-bid\_price) - ask\_size\*log(ask\_price-x) over the range [bid\_price, ask\_price].

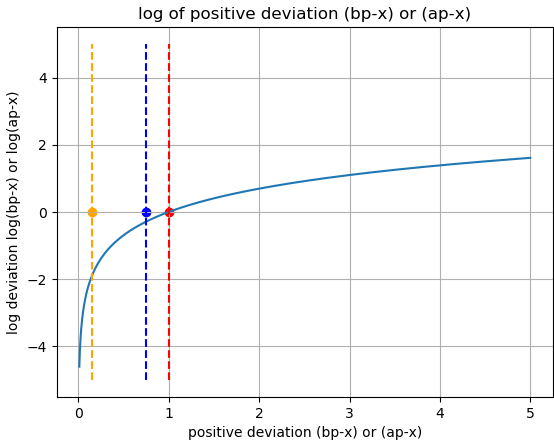
X-bid\_price = missing price on bid side. Ask\_price - x = missing price on ask side.

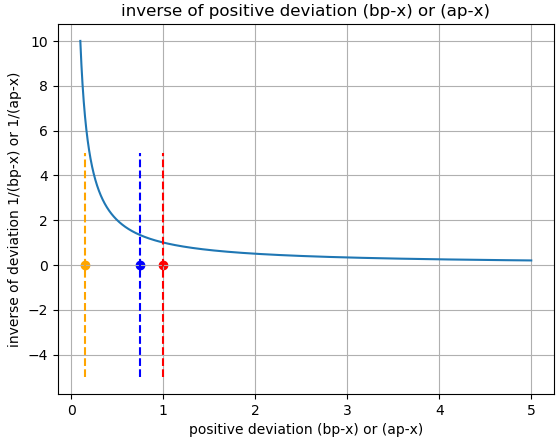
bid\_size\*log(x-bid\_price) = missing bid volume. ask\_size\*log(ask\_price-x) = missing ask volume.

Think of f(x) as the total missing volume. We want to find a price that minimizes the total missing volume. Missing volume means no trade can take place. When missing volume is minimized/less then trade can take place.

f(x) = bid\_size\*1/(x-bid\_price) + ask\_size\*1/(ask\_price-x) over the range [bid\_price, ask\_price].

In this equation we assume WAP i.e. x, x <= ask price or x >= bid price. AND bid price != ask price.





Instead of log we can also use other functions that can be computed faster. The function just has to have an asymptote at x=0. E.g. 1/x . Note: sign change in cost function.

Firstly, the deviation is highly unlikely to be greater than 1. Deviation closer to zero deviation incurs more cost.

In the code the wap (equilibrium price) is computed using the function full\_book\_wap\_bisect( df\_book, lvl=0) using a bisection algorithm iteratively and the wap moves towards the price (either bid or ask price) which has bigger bid or ask size and smaller . increasing lvl increases importance/weight of distance with respect to size. Higher lvl. Additionally, odd values of lvl make s negative and if bid size/price > ask size/price then p moves towards ask price. Similarly, opposite is true. For even values, always pushed towards ask price??

21) Explore the rest of the 46 features in clustering analysis to see if there are other feature that give a high score?

Ans: No need to try this because Jager must have tried this already. Actually we can try our features through this clustering method.

# **Feature SELECTION**

22) always check correlation between features to ensure no multicollinearity issues.

23) features target real vol

7 pc target real vol