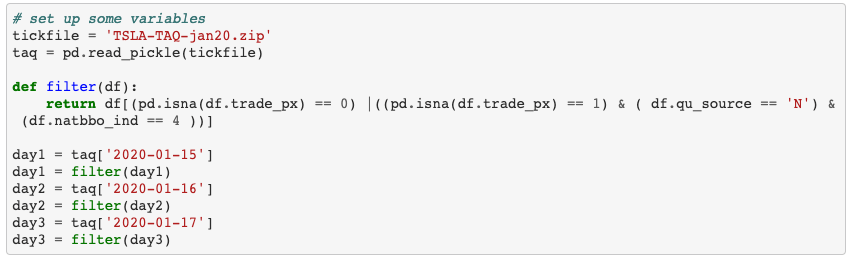
1. Optimizing the algo performance
   1. To accelerate the simulation, we cleaned the data by running the below function:

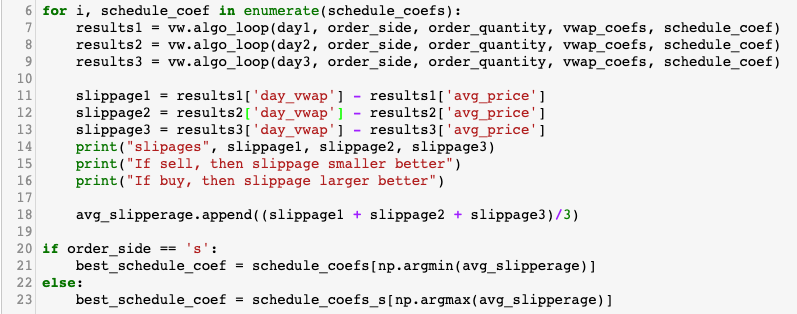


* 1. To select the best coefficients:
     1. Performance Measure:

The performance of a set of coefficients is based on the average slippage of the 3 days.

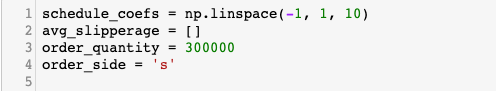
Notice that the for “sell”, the less the slippage, the better the performance.

Notice that the for “buy”, the larger the slippage, the better the performance.



* + 1. For schedule coefficients:

Set tick\_coefficient = 0, loop through schedule coefficients from -1 to 1 when the order side is “buy”, and loop through values from -1 to 0 when the order side is “sell”.



Then we select the best coefficient based on the performance measure mentioned in 1.2.1.

* + 1. For tick coefficients:

After we have got the best schedule coefficient for “buy” from 1.2.2, we assume it’s constant. Set schedule\_coefficient = best\_schedule\_coef, loop through tickor coefficients from -1 to 1 when the order side is “buy”.

For “sell”, the process is exactly the same.

Then we select the best coefficient based on the performance measure mentioned in 1.2.1.

* 1. Observation

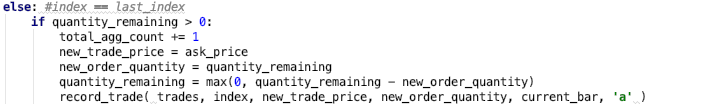
Considering the case where fair value is only determined by schedule factor. If the order side is “buy”, the schedule coefficient should be positive: if the current quantity is behind, the schedule factor is positive, so the fair value should be increased, which could be possible only if the coefficient is positive.

If the order side is “sell”, the schedule coefficient should be negative: if the current quantity is behind, the schedule factor is positive, so the fair value should be decreased to indicate the urgency to sell the current stock, which could be possible only if the coefficient is negative.

1. Bug Fix
   1. update quantity\_remaining after placing order passively
   2. update quantity\_behind after placing order aggressively
   3. Correct the calculation of quantity\_remaining from 'min(0, quantity\_remaining - new\_order\_quantity)' to 'max(0, quantity\_remaining - new\_order\_quantity)' to match the definition of this variable
   4. Revise the calc\_order\_quantity function, rounding down the order quantity to avoid overpurchase or oversale.
2. Force the order to complete exact number of shares

When the current index is the last index of the trading day, we buy or sell the remaining quantity aggressively.

When the order side is “buy”:



When the order side is “sell”:

