Jan 28th today, Jan 30th: Try more code structure design on the cancelling order, trade on seperate only one or two ticks, identify the price trend and stop loss on inventory, for loop while loop for cancel and use ⅔ sleep time

You can modify the code during the trade,

Live Trading taken away- ***MARKET CANCELLATION:*** As you cannot close ALL your current position on trade quantity, “CANCEL” allows cancellations on multiple types: (cancel all continuing orders at once, cancel all buy orders or sell orders, cancel the orders over a price level , cancel the orders on the dynamics trading volumes) this used to hedge risks, stop loss, conduct fast algo-manipulation.

(the $3 dollar Risk) ***INVENTORY ACCUMULATION RISK:*** - the other side of the market, aka. accumulating the wrong position in the wrong time and place.

THE SLEEP PROBLEM: Change the cancellation and while loop and for loop to use of the ⅔ of time out of trading because of the sleep time 0.5.

1.1 The Profit Function of Algorithmic Trading (the Price Market Move and the $3 dollar Risk):

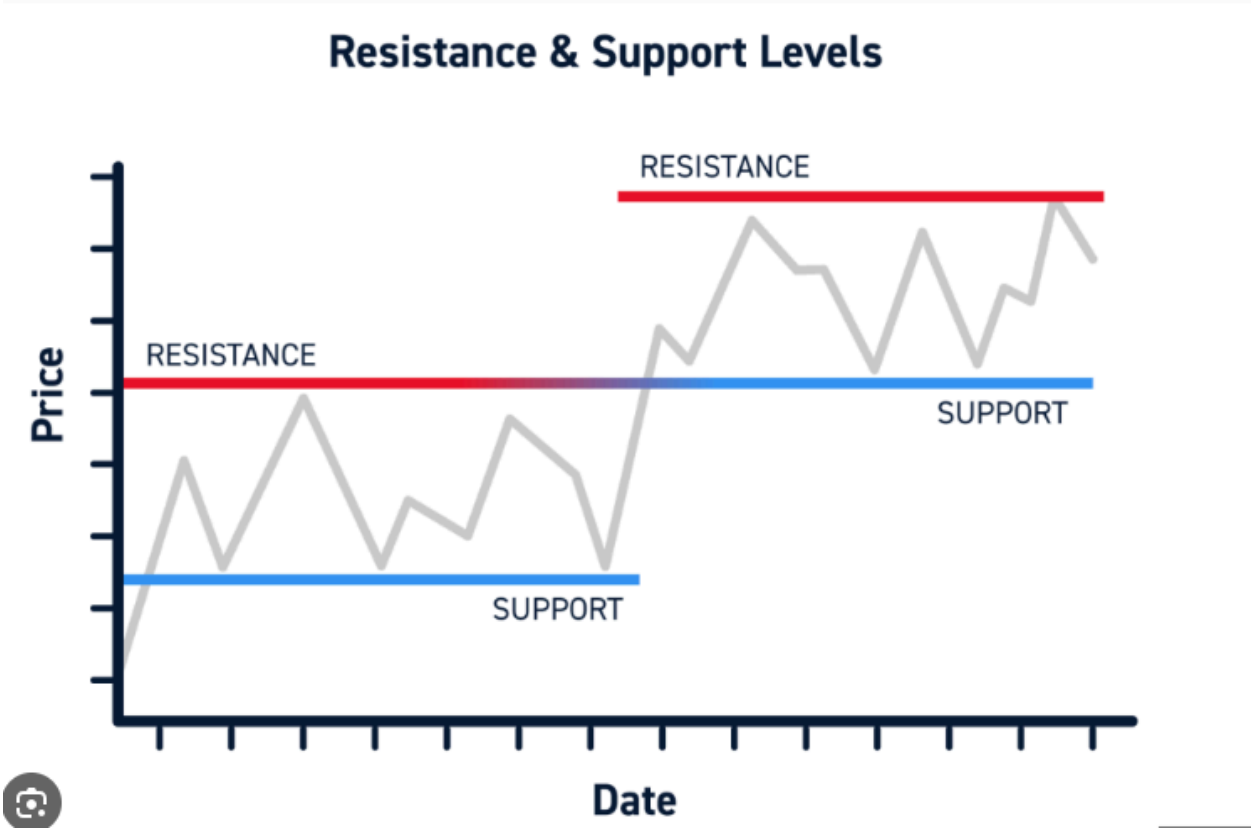
You are the Market Marker: Think about your profit function, or at least your revenue function is :**On the one side is the Bid-ask Spread: Simply the difference between the buying and selling prices multiplied by the number of shares that you can trade. Spread bid ask spread narrowing down because of a dynamic market, the rewards will shrink.**

Another side of the equation is the market move: Market moves against :(you, you are long and stock price drops, you are short and the stock price increases…

What we called inventory risk? - you are long and stock price drops:(, you are short and the stock price increases:(, you are in the wrong position at the wrong time ! that is inventory risk on you.
That just always happens when you loop for [buy sell] when price is decreasing only buy or only sell can avoid loss position.

Example: (the $3 dollar risk)

When I start the case I look at The Bid-ask spread , The bid-ask spread is $0.10, $0.20, that is your profits, your earnings. So you have a reward on $0.20 for every share that you can buy and sell.
And your risk is $3, when you look at the CNR price stock chart , CNR starts from $166, down to $163, closed around $160. So from the price trend chart, you are in the wrong position, as the $3 downward movement is a terrible risk-forward trade-off。



Terrible very terrible , as if you are right you earn $0.10, if you are wrong you lose $3.0. This is a 1 to 30 ratio. How do we make money then?

You want to buy and sell buy and sell and , buy and sell, you want to hold your position as short as you can. You don't want to get 2000 quantity , then just wait.

So when you are thinking about designing your algo, your trading strategies, you might consider something affected by *time and risk*, for example, a long position I have is ten seconds old, or twenty seconds… static on my hands, I don't need long position static in my hand.

1.2 Market Static & How to do Something to Reverse the Situation on Inventory Risk (the $3 dollar risks) Price Risk

When you trading, your are placing bid and asks on the market: now when the market trend is market price decreasing, the market is now on the bid side, your bids orders will be filled much quickly , which means if you are placing bids orders, whenever excute buy. You are only wasting buying when others willing to sells.

So Now, as a market maker, you are always accumulating the wrong position at the wrong time! If you are rebating, you are only wasting buying when others are willing to sell.

Timing Statics and Wrong Price Position both simultaneously happens as: *time and risk*,
And you cannot sell out, as the price trend chart is downward reducing pricing, almost everyone is willing to sell.

You are always in the wrong position if you only use the sample starting code(Simple Looping trade on the bid-ask spreads.) How do you modify the code to try to avoid that ?

SOLUTION：So what could I do? Use Cancel ? how to deal with the three dollar risk,

→ I was thinking about maybe design the code to let the computer diagnose the price trend by itself ??? I don't want a wrong position in BUY or SELL.
Class Four 11 a.m. Professor's Code-: Check Momentum thirteen functions Added one function "Momentum" , edit main functions in While loop for loop, execute the responding code for cancel orders, Simple and Clear.

```
        return order['status']

def check_momentum(ticker):
    payload = {'ticker': ticker}
    resp = s.get('http://localhost:9999/v1/securities/book', params = payload)
    if resp.ok:
        trend = None
        book = resp.json()
        bid_side_book = book['bids']
        ask_side_book = book['asks']
        if len(bid_side_book) > len(ask_side_book) + 10:
            trend = 'Up'
        elif len (bid_side_book) + 10 < len(ask_side_book):
            trend = 'Down'
    return trend
```

Cancel all buy orders on specific ticks when the price trending is decreasing.

Pricing, if stocks appear to be price trendings, that might send a different signal to your aglo.

Price Trends: Pricing in different conditions.Add Price Premium or discount.
→ I was thinking maybe set the price premium and price discounts on existing prices to reach the threshold to stop and continue loops on the buy and sell.

- **Decreasing the net_limite Position , will reduce the trading volume, will shrink the profit as well as shrink the loss.**
-

ALGO run 2: Hint is to edit your code, I know it is tough, but you think about some of the easier things that we could like changing the prices, Put in a premium or a discount on the stock prices, see how that can affect your profitability and try it out.

ALGO run 3, 4, 5: Your algo trading performance sometimes affects the market conditions, which after running, you find some RY stock prices don't move, some AC price went down and up. CNR price a large raise/decrease. When you run on the case soon a little, you will see a consistent price trend pattern.

1.3 Decision Making:

When you see the price movements, the trading volume, you can think a little bit on having a strategy tailored to each one.

You don't necessarily need to trade all three stocks, could trade one stock, could trade two stocks. Only requirement is trade 25,000 shares at least,

In this case. Decisions include: 1.ALL RY, 2.ALL AC, 3.AC&RY, or ALL three ticks, in your algo decision making a little bit about the rewards versus the risks of each one of the stocks. → Separate functions circulating For Loop and While Loop for three individual stocks. Only trade one or two stocks sometimes.
Code sample for only trading AC, CNR

```
def main():
    tick, status = get_tick()
    ticker_list = ['CNR', 'AC']

    while status == 'ACTIVE':

        for i in range(2):

            ticker_symbol = ticker_list[i]
            net_position = get_net_position() # modify to include net positions
            gross_position = get_gross_position()
            position = get_ticker_position(ticker_symbol)[0]['position']
            best_bid_price, best_ask_price = get_bid_ask(ticker_symbol)
```

- **In this case , CNR has more price risks than RY, you might only consider only trade CNR if you can capture a big bid-ask spread.**

- **In this case, RY has less price risks than CNR, so you might consider trading larger quantities and volumes of RY.**

the price movements→ **I was thinking about only stimulating sales on CNR maybe? , this also take advantage and make profit by capturing a big bid-ask spread**

the trading volume→ **I was thinking about stimulate larger Order_LIMIT quantity on RY maybe? , this also take advantage and make profit by expanding exchanged large quantity for certainty.**

1.4 Last: The Sleep Problem:Agglo has two thirds of time sleeping, out of trade. Only use one while loop and for loop for all three stocks, only one-third of quantity trade, only one-third of profit made.

the loop sleep time: Sleep time (0.5)

1.5 seconds per loop
After While loop start, then we go into for loop,

Only one while loop and for loop conduct by sequence for AC RY and CNR,

Sequence AC first->0.5 s , RYthen-> 0.5s, CNRend -> 0.5s

For AC, RY , CNR , The time algo trading out of market is ⅔ of time.

300 seconds out of the market per trade case, algo trade 100 out of 300 ticks for CNR in one trade case, for RY either. So this algo spends more time not trading than trading. This is inherent to your PNO.So how can I change the algo?

SOLUTION: 1. → Run three loops simultaneously.

2. → Move the cancel outside of the for loop. We will put orders on all three stocks.This will cancel all.For one stock trade, like a KILL ALL, this means that we could only trade one stock in this for loop, we won't wait on the other two.

Pause the code to show you something will break all of your LOOPS. Upon a market order on the screen for CNR, when it was in the for loop , it went through the for loop. The ticker symbol was CNR , it cancels all the CNR orders.Because this line of code is now outside the for loop. It will not be going through the ticker symbol list.

```
            if position < 0: # Checking for short position and buying to neutralize position
                resp = s.post('http://localhost:9999/v1/orders', params = {'ticker': ticker_symbol, 'type': 'LIMIT', 'quantity': abs(position), 'price': best_bid_price, 'action': 'BUY'}
            sleep(0.25)


            best_bid_price, best_ask_price = get_bid_ask(ticker_symbol)

            if gross_position < (MAX_LONG_EXPOSURE - ORDER_LIMIT) and net_position < 20000: # testing against net limits, but gross limit hit first, check against gross limit
                # Check NET positions; is less than 21000
                resp = s.post('http://localhost:9999/v1/orders', params = {'ticker': ticker_symbol, 'type': 'LIMIT', 'quantity': ORDER_LIMIT, 'price': best_bid_price - 0.01, 'action': '

            if gross_position < (MAX_SHORT_EXPOSURE - ORDER_LIMIT) and net_position > -20000: # comparing gross position to gross limit; change greater than sign to less than sign
                resp = s.post('http://localhost:9999/v1/orders', params = {'ticker': ticker_symbol, 'type': 'LIMIT', 'quantity': ORDER_LIMIT, 'price': best_ask_price + 0.01, 'action': '




            sleep(0.25)

            s.post('http://localhost:9999/v1/commands/cancel', params = {'ticker': ticker_symbol})

        tick, status = get_tick()

if __name__ == '__main__':
    main()
```

This will efficiently increase the amount algo trading and maximize your profit.

I have uploaded the ALGO2e renewed code for 1) Check the price momentum and cancel buy in orders when trend down 2) Optimize the sleep.3) Consider only trade AC and CNR, as RY price is uncertain, the most inventory risk.