**Project Report**

**Initial Comments:**

Jupyter will be used as the coding environment, as it will allow debugging chunks of code without hassle. Libraries such as Backtrader exist which might make the process simpler, but I would prefer to go forward with standard NumPy and Pandas. For one, I am familiar with these libraries and won’t have to rely on LLMs (hello GPT) as much to guide me. Additionally, the strategy seems simple enough to hardcode. Hardcoding the strategy will allow me to truly comprehend and understand the nitty-gritty involved with backtesting.

**Process to follow**:

1. Data cleaning: Eliminate columns for Monthly contract, OI, Volume, Open, High, and Close. Also, keep only rows with contract\_weekly = I. From Futures data, keep only rows with contract = I. Again, eliminate columns for Open, High, Low, Volume, and OI.
2. On every trading day at 10.30 am, extract the spot price from the futures data. This will serve as ATM reference.
3. Next, enter a short position on N qty each of call and put options which are ATM. This is called making an ATM Straddle (suppose premiums are SC, SP respectively).
4. As both call and put short positions are entered, the trade stands to benefit if price moves in either direction (without accounting for premium paid).
5. To hedge your position (i.e., price moves but not enough to cover up for total premium made), enter a long position on N qty each of out-of-the-money call and put options, with strike at +/- 2% of the spot price. (suppose premiums are LC, LP).
6. If the spot price moves above the strike, the short call option is OTM, short put position is ITM. If the spot price stays within 2%, the long-put position is ITM and the long call is OTM. If it rises above 2%, long call is ITM. This overall process ensures a non-zero payoff no matter where the price moves (the risk of loss is still there, depending on the premiums charged).
7. If current premiums are SC’, SP’, LC’, LP’, then

• 30% stop loss hit for short positions if (SC’+SP’) = 1.3(SC+SP), and 30% stop loss hit for long positions if (LC’+LP’) = 0.7(LC+LP). ATM (or near) strike prices charge higher premiums, which decrease as the strike moves OTM.

• Exit short position when SC’=1.3SC, SP’=1.3SP and exit long position when LC’=0.7LC, LP’=0.7LP

• The same argument can be applied to the target of 80% as well. Therefore, target is hit if SC’=0.2SC, SP’=0.2SP for short position and LC’=1.8LC, LP’=1.8LP for long position.

1. If neither condition is met, exit all trades at 3.20 pm of the same date.
2. In any case, the return for that day will be N\*lot\_size\*[(SC+SP)-(SC’+SP’) + (LC’+LP’)-(LC+LP)]
3. The final revenue obtained on each day will be used as starting capital for the next day (i.e., compounded returns).
4. To track the overall portfolio movement, consider the movement of Pfolio(V) = -SC’-SP’+LC’+LP’. Use this movement to compute the drawdown for each day.
5. Keep track of Pfolio(V) for each day, and use that movement for plotting and for computing overall max drawdown.
6. Also, keep track of the final revenue on each day. Use this for plotting the equity curve.

**Assumptions Made:**

1. The lot size is taken to be 25, as was the case in 2017. It was only recently changed to 15.
2. Quantity traded is taken to be 3 for all days. The average value for quantity traded in the sample report was 2.7, so I took the nearest integer.
3. The continuous movement of equity value between the close of one day and open of the next day.
4. We do not need to consider High and Low prices for calculating drawdown (firstly, it is an imperfect idea, additionally, it won’t affect the outcome appreciably).
5. The risk-free rate (Rf) was taken to be 6.9% (which is equal to the 3-year bond return rate in 2017). Source: <https://www.worldgovernmentbonds.com/bond-historical-data/india/3-years/>

**Philosophy behind functions used:**

1. Make a function for finding the closest value to a given value in an array
2. Add another column in dataframe ‘Status’
3. For any given day, extract the future strike price from futures.csv and use the closest\_val function to find the ATM strike. Also, use it for +/- 2% strike.
4. Mark the status column of ATM with S\_En (for short enter) and +2% with L\_En (for long\_enter). Take the ticker values and store it. Filter the data for that day by taking only those two ticker values, and make two separate dataframes, one for the ticker corresponding to S\_En and one for ticker corresponding to L\_En. Also store the ATM price and OTM price.
5. In the filtered dataframe for S\_En, loop over every row. Calculate ratio between the closing price and the ATM price identified above. If it is greater than or equal to 1.3, or less than or equal to 0.2, mark the corresponding status as S\_Ex (for short exit), and exit the loop whenever this condition is satisfied for the first time. If S\_Ex marked because of ratio exceeding 1.3, then increment counter for loss\_trade. If S\_Ex marked because of ratio dropping below 0.2, increment counter for prof\_trade.
6. Similarly, if the dataframe corresponds to ticker of L\_En, then calculate ratio between the closing price and the OTM price identified above. If it is less than or equal to 0.7, or greater than or equal to 1.8, mark the corresponding status as L\_Ex (for long exit), whenever this condition satisfied for the first time. If L\_Ex marked due to ratio < 0.7 then increment counter for loss\_trade, else increment counter for prof\_trade
7. If any of the above conditions aren’t satisfied, then mark S\_Ex and L\_Ex for the corresponding tickers at timestamp 15:20. In this case, if dataframe is for S\_En, then calculate ratio of current close price and ATM price. If ratio greater than 1, increment counter for loss\_trade, else increment counter for prof\_trade.
8. Similarly, if dataframe corresponds go L\_En, then calculate ratio of current close price and OTM price. If ratio less than 1, increment counter for loss\_trade, else increment counter for prof\_trade

**Metrics Calculated:**

1. Hit Ratio: Profitable Trades / Total Trades
2. Risk Reward: Ratio of max loss to max profit. If expressed in form of 01:x, I need to invest 1 unit to get a reward (profit) of x units
3. Calmar Ratio: (Rp-Rf)/Max drawdown. Rp is 3 3-year portfolio return and Rf is risk-free rate. A higher ratio points to a better investment.
4. Avg Return per trade
5. Avg Profit per trade
6. Avg Loss per trade
7. Compounded Annual Growth Rate
8. Max Drawdown