**Trading Strategy Risk Management Rules**

**1.Position Assignment**

**Explanation:** The position is assigned based on the trading signal generated by the strategy. A Buy signal is represented by 1, a Sell signal by -1, and hold by 0.

**Relevance:** This rule ensures clear and straightforward identification of the current market stance.

**2. Stop Loss Rule**

**Condition:**

**Explanation:** This rule aims to limit potential losses. If the closing price falls below a certain percentage of the rolling maximum closing price, it triggers a sell (Position = -1).

**Relevance:** By implementing a stop-loss, the strategy aims to protect capital from excessive declines during unfavourable market conditions.

**Action:**

**Explanation:** If the Stop Loss condition is met, the position is set to Sell (-1), indicating a protective move to limit losses.

**Perks:** Mitigating downside risk enhances the overall risk-adjusted performance of the strategy.

**Example:**

stop\_loss\_condition = df\_resampled['close'] < df\_resampled['close'].rolling(autocorr\_length\_close).max() \* (1 - max\_risk\_per\_trade)

df\_resampled.loc[stop\_loss\_condition, 'Position'] = -1

**3. Doubling Investment after Loss**

**Condition:**

**Explanation:** This rule aims to recover losses. If the strategy faces a loss (Position = -1), it doubles the investment amount for the next trading round.

**Relevance:** Doubling down after a loss is a high-risk, high-reward strategy designed to accelerate capital recovery.

**Action:**

**Explanation:** After facing a loss, the investment amount is multiplied by 2, increasing the capital to recover previous losses.

**Perks:** If successful, this strategy can expedite the recovery process and potentially lead to higher overall returns.

**Example:**

loss\_condition = df\_resampled['Position'] == -1

df\_resampled.loc[loss\_condition, 'Investment'] \*= 2

**4. Mean Reversion Strategy**

**Condition:**

**Explanation:**

This strategy aims to identify mean-reverting conditions based on the autocorrelation length of the closing prices.

If the autocorrelation falls below a specified threshold, it signals a mean-reversion condition

(Mean\_Reversion\_Signal = -1).

**Relevance:** Mean reversion provides an opportunity to capitalize on price reversals after extended trends.

**Action:**

**Explanation:**

When the Mean Reversion condition is met, it triggers a sell signal, positioning the strategy for potential profit from price reversals.

**Perks:** Capitalizing on mean-reverting conditions can enhance overall profitability.

**Example:**

mean\_reversion\_condition = df\_resampled['Autocorrelation\_Lag\_' + str(autocorr\_length\_close)] < autocorrelation\_threshold

df\_resampled.loc[mean\_reversion\_condition, 'Mean\_Reversion\_Signal'] = -1

**5. Transaction Cost**

**Percentage:** 0.15%

**Explanation:** Transaction costs are incurred for each buy or sell transaction. This percentage is deducted from the investment amount.

**Relevance:** Factoring in transaction costs provides a realistic representation of the impact of trading expenses on overall profitability.

**6. Maximum Risk per Trade**

**Percentage:** 2% (Max)

**Explanation:** This rule sets a limit on the maximum risk per trade. If the calculated risk exceeds this percentage, positions are adjusted to meet the set maximum risk.

**Relevance:** Restricting the maximum risk per trade helps maintain a balanced and controlled approach to trading, preventing large drawdowns.

**Example:**

max\_risk\_per\_trade = 0.02

df\_resampled.loc[df\_resampled['close'] < df\_resampled['close'].rolling(autocorr\_length\_close).max() \* (1 - max\_risk\_per\_trade), 'Position'] = -1