# 

# Evaluate trading strategy in Python & gauge significance of testing results

Contents

[Evaluate trading strategy in Python & gauge significance of testing results 0](#_Toc144308608)

[1. Background 2](#_Toc144308609)

[2. Explanation of Code 4](#_Toc144308610)

[a) Setup 4](#_Toc144308611)

[b) Initial Data Exploration 6](#_Toc144308612)

[c) Graphical Visualization 11](#_Toc144308613)

[d) Implementing Technical Indicators 15](#_Toc144308614)

[i. TA-lib 15](#_Toc144308615)

[ii. MACD 16](#_Toc144308616)

[iii. ATR 18](#_Toc144308617)

[iv. Bollinger Bands 20](#_Toc144308618)

[v. Relative Strength Index (RSI) 21](#_Toc144308619)

[vi. ADX (Average Directional Index) 24](#_Toc144308620)

[vii. Renko chart 25](#_Toc144308621)

[e) Implementing Performance Measurements 27](#_Toc144308622)

[i. CAGR 27](#_Toc144308623)

[ii. Annualized Volatility 28](#_Toc144308624)

[iii. Sharpe Ratio 29](#_Toc144308625)

[iv. Sortino Ratio 29](#_Toc144308626)

[v. Maximum Drawdown 30](#_Toc144308627)

[vi. Calmar Ratio 31](#_Toc144308628)

[f) Implementing Trading Strategy 31](#_Toc144308629)

[i. Calmar Ratio 31](#_Toc144308630)

[ii. Resistance Breakout 34](#_Toc144308631)

# Background

A trading strategy consists of objective entry and exit criteria that determine when to buy or sell financial assets.

Strategies aim to generate profits by exploiting inefficiencies in markets. The components of a strategy include:

1. **Market selection** - What asset classes or instruments to trade like stocks, options, futures, or currencies. Market selection depends on factors like capital, risk tolerance, and knowledge of the asset class.
2. **Timeframe** - The duration between entering and exiting positions, like intraday, swing or position trading. Day trading involves opening and closing positions within the same day to profit off short-term fluctuations. Swing trading holds trades for days or weeks to capture larger moves and trends. Position trading is long-term with trades held for months or years based on major trend direction. Shorter timeframes require more frequent analysis and trading activity.
3. **Risk management** - The amount of capital exposed on each trade and trade sizing based on volatility. Also includes stop losses to limit downside.
4. **Entry rules** - The market conditions or technical indicators that trigger opening trades. Common entries use breakouts, support/resistance, moving averages etc. Entries aim to identify opportune points to enter winning trades.
5. **Exit rules** - When and how to close out positions for profit or loss. Exits may use take profit targets, trailing stops, technical indicators etc. Exit techniques include stop losses to limit downside, trailing stops to lock in profits as price moves favorably, and take profit targets based on risk/reward ratios. Strict exit rules are crucial for maximizing gains and minimizing losses.

Robust risk management is vital for trading success. Position sizing models like fixed fractional or percent of account balance size trades appropriately for portfolio risk parameters. Generally, it is good to limit the risk per trade to 1% - 2% of capital.

Backtesting is the process of testing a trading strategy on historical data to gauge its potential future performance. Backtesting enables traders to evaluate and refine trading systems before risking real capital. By applying a strategy to historical data, backtesting simulates how a strategy might have performed if it was traded live. This allows traders to identify flaws and optimize the strategy before implementation.

Several key metrics can be used to evaluate the performance of a trading strategy during backtesting. These include:

1. **Total return** - The overall profit or loss generated by the strategy over the backtest period. Higher total returns indicate better performance.
2. **Annualized return** - The average annual return or compound annual growth rate (CAGR) over the time period. Useful for comparing strategies over different time frames.
3. **Risk-adjusted return** - Returns normalized for the amount of risk taken. Common metrics include the Sharpe ratio and Sortino ratio. Higher ratios indicate better returns per unit of risk.
4. **Maximum drawdown** - The largest peak to trough decline in the strategy equity curve. Lower maximum drawdown suggests better risk management.
5. **Win rate** - The percentage of winning trades out of total trades. Higher win rates suggest a strategy is profitable more often than not.
6. **Profit factor** - The ratio of the average profit of winning trades to average loss of losing trades. Above 1.0 suggests a profitable strategy. Higher is better.

Statistical significance testing can verify if performance metrics like total return are due to skill or just luck. Hypothesis testing concepts like p-values and t-statistics are used to test the probability a strategy's performance is random. Strategies with low p-values and high t-stats are statistically significant.

By backtesting over longer time frames with diverse market conditions, a strategy can be robustly evaluated. Visualizations like equity curves, drawdown plots and monte carlo simulations also provide insights into performance.

By combining sound risk management, well-defined entry and exit rules, and strong technical research, traders can develop high-probability trading strategies with an edge over the markets. Meticulous backtesting and optimization is crucial before risking real capital. Following structured development procedures can create viable systems and prevent overfitting historical data.

# Explanation of Code

## Setup

A screenshot of a computer

Description automatically generated

We start our code by choosing relevant settings for displaying our code.

Firstly, we print the system version for diagnostic purposes. If you are trying to run this using jupyter notebook on another machine, it’s best if you run this using python version 3.11.4.

We also suppress all warnings that might show up, display any data frames for a maximum of 25 rows and 25 columns, and suppress any scientific notation that might be shown in our data frame.

Lastly, we also import relevant python modules as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Library used** | **Purpose of library** | **Link** |
| 1 | Datetime | datetime library defines the entire duration for which we would like to obtain financial data for. | <https://docs.python.org/3/library/datetime.html> |
| 2 | Yfinance | yfinance library is essential to importing financial data from yahoo finance. | <https://github.com/ranaroussi/yfinance> |
| 3 | Pandas | pandas library is used for manipulating our dataframes. | <https://pandas.pydata.org/docs/user_guide/index.html> |
| 4 | Matplotlib | matplotlib library is used for graphical visualization. | <https://matplotlib.org/stable/users/index> |
| 5 | Talib | talib library demonstrates usage of various technical indicators. | <https://ta-lib.github.io/ta-lib-python/> |
| 6 | Copy | copy library preserves data of original/input dataframe and make changes to copy only. | <https://docs.python.org/3/library/copy.html> |
| 7 | Time | time library is used to provide time-related functions. | <https://docs.python.org/3/library/time.html> |
| 8 | Stocktrends | stocktrends library calculate trends in stock market. We utilize the Renko chart function from this library. | <https://github.com/ChillarAnand/stocktrends> |
| 9 | Statsmodel.api | statsmodel library allows for advanced statistical testing and modelling. We are using this for OLS regression purposes. | <https://www.statsmodels.org/stable/api.html> |
| 10 | Numpy | Numpy is used for numerical computations and multidimensional array manipulation | <https://numpy.org/doc/stable/index.html> |

## Initial Data Exploration

A screenshot of a computer program

Description automatically generated

Before creating the trading strategies, it is always recommended to conduct a preliminary exploration and analysis of the dataset.

I have chosen the following portfolio of stocks, which will be utilized later on, while creating the trading strategies:

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Ticker** | **Full name of company** | **Link** |
| 1 | MSFT | Microsoft Corporation | <https://finance.yahoo.com/quote/MSFT?p=MSFT&.tsrc=fin-srch> |
| 2 | AAPL | Apple Inc. | <https://finance.yahoo.com/quote/AAPL?p=AAPL&.tsrc=fin-srch> |
| 3 | META | Meta Platforms, Inc. | <https://finance.yahoo.com/quote/META?p=META&.tsrc=fin-srch> |
| 4 | AMZN | Amazon.com, Inc. | <https://finance.yahoo.com/quote/AMZN?p=AMZN&.tsrc=fin-srch> |
| 5 | INTC | Intel Corporation, Inc. | <https://finance.yahoo.com/quote/INTC?p=INTC&.tsrc=fin-srch> |
| 6 | CSCO | Cisco Systems, Inc. | <https://finance.yahoo.com/quote/CSCO?p=CSCO&.tsrc=fin-srch> |
| 7 | VZ | Verizon Communications Inc. | <https://finance.yahoo.com/quote/VZ?p=VZ&.tsrc=fin-srch> |
| 8 | IBM | International Business Machines Corporation | <https://finance.yahoo.com/quote/IBM?p=IBM&.tsrc=fin-srch> |
| 9 | QCOM | QUALCOMM Incorporated | <https://finance.yahoo.com/quote/QCOM?p=QCOM&.tsrc=fin-srch> |
| 10 | TSLA | Tesla, Inc. | <https://sg.finance.yahoo.com/quote/TSLA?p=TSLA&.tsrc=fin-srch> |

To understand the wider picture first, the data for these stocks were fetched for a period of 3650 days, ranging all the way from 2013-09-03 to 2023-08-29.

This data was fetched using yfinance, a free library to obtain data from yahoo finance easily. It is important to note that while it fetches long-period of data reliably, it does have some limits on intraday data, as shown below, which poses a slight inconvenience as we will see later:

1. For 1 minute data, it is limited to a maximum of 7 days within the last 30 days.
2. For 60-minute data, it is limited to a maximum of the last 730 days.
3. For 90-minute data, it is limited to a maximum of the last 60 days.

A screenshot of a computer

Description automatically generated

Thereafter, as part of the data cleaning process, we drop any NaN values in the dataframe and do it in an in-place manner.

Additionally, we view a summary of each of the columns to give us a better understanding of each individual company that we have analyzed. For example, looking at Microsoft, we can notice that from 2013 to 2023, it had a minimum price of 26 and a maximum price of 358. Moreover, it had a standard deviation of 97.

Note: Each ticker should be viewed individually and cannot be compared with other tickers as we aren’t comparing daily returns.

A screenshot of a computer

Description automatically generated

Then, we calculate the daily % change using pandas.pct\_change function. This allows us to compare the results between the rest of the tickers. We also calculate the summary statistics for this daily return dataframe as well.

A white rectangular object with a black border

Description automatically generated

Thereafter, we also display the mean and standard deviation of the daily returns for all the tickers.

A screenshot of a computer

Description automatically generated

We also calculate the simple moving average using a rolling window of size 10, along with a rolling standard deviation, rolling maximum and rolling sum.

A basic skim of the SMA can help us understand the general trend, as well as smooth out any volatility in the daily value of the stock.

## Graphical Visualization

A graph of different colored lines

Description automatically generated

The above chart simply shows how the closing price of the stock has changed from 2013 to 2023. As we can see, we have top-performing stocks like MSFT, META, TSL, AAPL, as well as relatively low performing stocks like INTC, VZ and CSCO.

A graph of a stock market

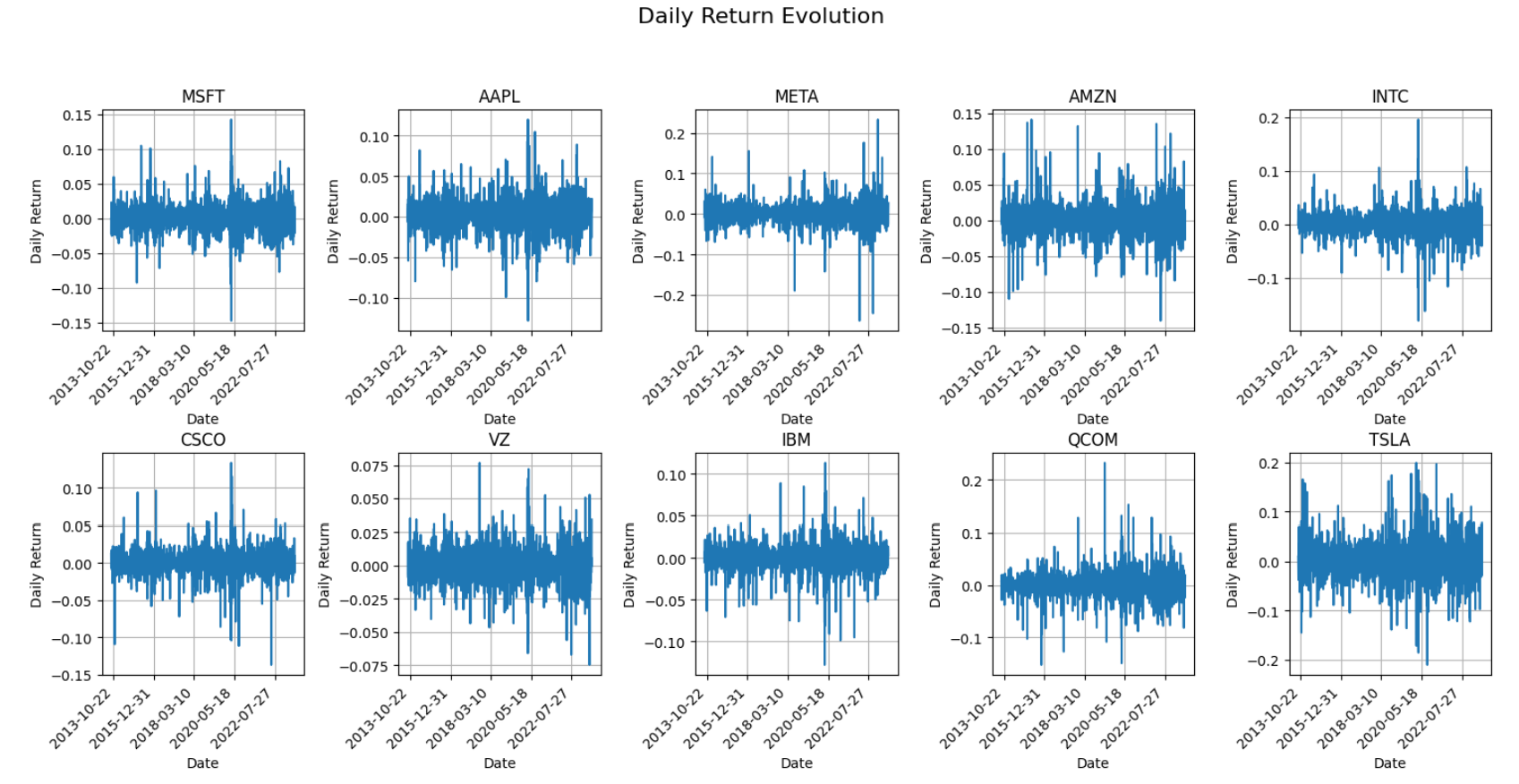
Description automatically generated with medium confidence

We have also created subplots to better visualize the trend for each of these stocks individually. Similar to the above observation, we notice a strong uptrend for stocks like MSFT, AAPL, AMZN, TSLA. We also notice a downtrend lately, in stocks like VZ and INTC.

A screen shot of a graph

Description automatically generated

Thereafter, we plot the daily returns from our portfolio and gain a pictorial view of the mean, standard deviation etc.

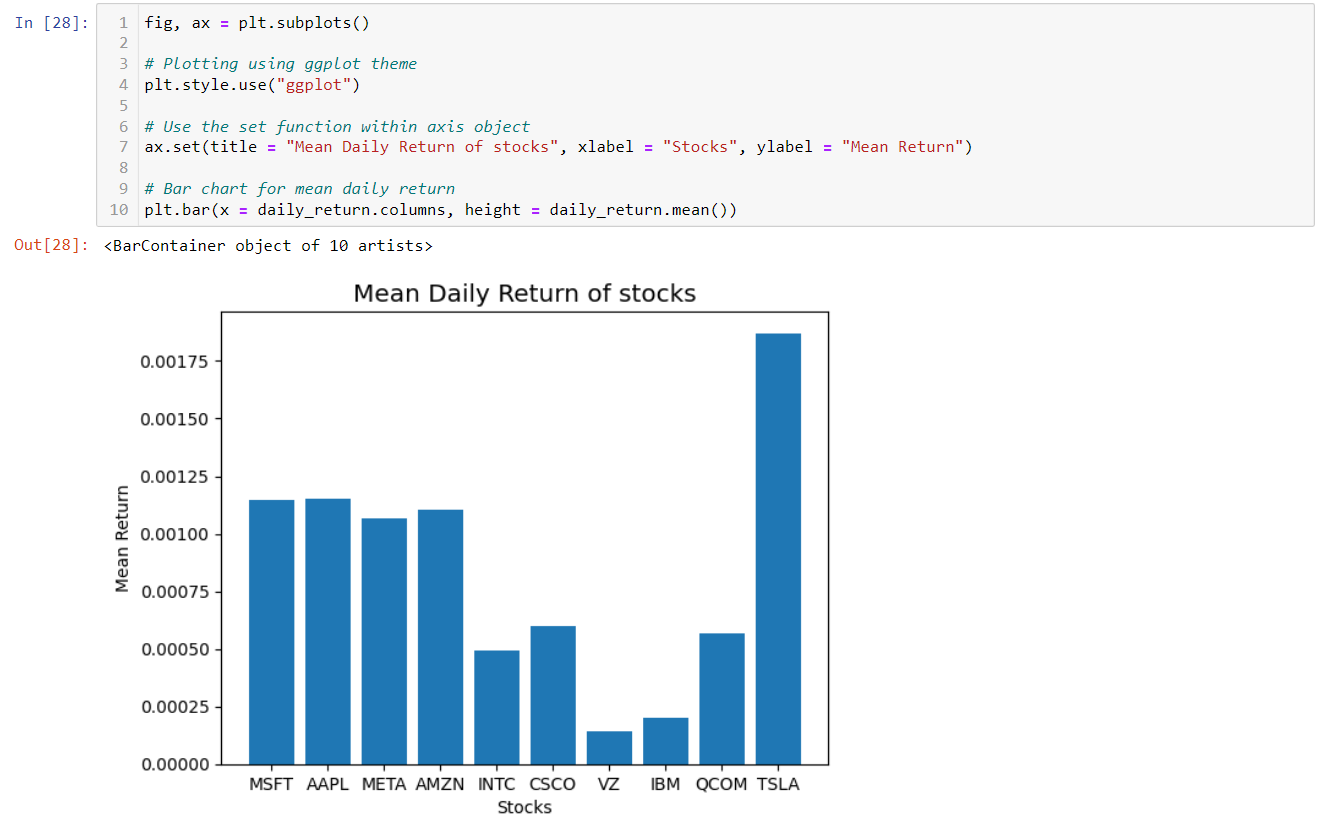


We also plot the daily return evolution for each of the stock to better understand the nuances in daily returns. For example, TSLA is a fairly volatile stock with multiple jumps in daily returns, both ways. Additionally, INTC also had a big spike around 2020, as we can observe from the chart.

A graph of stock market returns

Description automatically generated with medium confidence

Finally, we display a cumulative return for the portfolio, which essentially answers the question of “What would your return look like today, had you invested $1 back in 2013 in all these companies?”



We also plot the mean daily return of each of these stocks, and notice that TSLA has one of the highest mean daily returns, with MSFT, AAPL, META and AMZN, following thereafter.

A graph of a graph

Description automatically generated with medium confidence

We also plot the standard deviation daily return of each of these stocks to observe how much they have varied, i.e., their volatility. Again, we notice TSLA, META and surprisingly, QCOM, despite its low mean daily return.

## Implementing Technical Indicators

### TA-lib

We will implement technical indicators on a shorter timeframe, i.e., data for a period of 1 month with a 5-minute interval.

A screenshot of a computer program

Description automatically generated

One such library that helps in implementing these functions is Ta-lib.

We can not only calculate momentum indicators like ADX, MACD etc, but also chart patterns and statistical functions.

### MACD

MACD, also known as Moving Average Convergence Divergence, is a trend-following momentum indicator. It uses moving averages to identify changing momentum and trend direction.

MACD is typically calculated by taking the difference of two moving averages, a long moving average typically of 26 periods and a shorter moving average, typically of 12 periods. Moreover, a signal line, typically of 9 periods, is also calculated and is a moving average of the MACD line.

When the MACD line crosses above the signal line, it signifies upside momentum and marks a potential buy signal. On the other hand, when the MACD crosses below the signal line, downward momentum is signaled which could indicate a sell signal. The more separation or divergence between the MACD and signal lines, the stronger the trend.

The MACD also gauges overbought and oversold levels. When the MACD rises sharply to a high positive level, it implies overbought conditions characterized by excessive optimism. This may signify an impending correction or pullback. Conversely, falling deeply into negative territory implies oversold conditions indicating pessimism and a potential reversal back higher.

MACD can be employed with SMA, but many traders prefer EMA because it gives more weight to recent data points and makes it more responsive to latest price movements, which might closely parallel the upcoming future.

However, important points of MACD must be noted. Firstly, it is a lagging indicator and trails behind the actual price action. Moreover, this indicator generates many false positives which is exceedingly common in sideway markets. Therefore, traders typically use this MACD indicator, only in conjunction with other indicators.

A screenshot of a computer code

Description automatically generated

In the above code, we calculate the MACD indicator with the typical values:

1. Fast moving average line of MACD: 12 days
2. Slow moving average of MACD: 26 days
3. Signal line moving average: 9 days

We calculate the fast-moving average line and slow-moving average line by using the exponential moving window function (pandas.DataFrame.ewm).

We set our min\_periods so that it calculates moving average only when given that many rows of data.

Additionally, we use the span function, which is calculated as follows, according:

This span parameter is related to the decay factor used to weight past observations in the calculation of EMA. A smaller span value will give more weight to recent observations, while a larger span value will give less weight to recent observations.

Thereafter, we subtract the fast-moving average line with the slow-moving average line and calculate the MACD line.

To calculate the signal line, we calculate the exponential moving average on this MACD line.

### ATR

The Average True Range (ATR) is a technical analysis indicator that measures market volatility for the entire range of an asset price for that period. It focuses on total price movement and essentially conveys how wildly the market is swinging as it moves.

To calculate the ATR, we first need to find the True Range, which is the maximum of the following ranges:

1. Difference between High and Low of each period
2. Difference between current period High and previous period’s close.
3. Difference between current period Low and previous period’s close.

Thereafter, we take the exponential moving average of this true range over a 14-day period and calculate the ATR.

The ATR indicator can be used for many purposes. A primary use of ATR is to gauge the current level of volatility for risk management purposes. Higher ATR values signify increased volatility, while Lower ATR signifies calmer markets. Additionally, ATR is used to determine position sizing in markets with differing volatility levels. Larger ATR values suggest reducing position size to cut risk exposure. Conversely, smaller ATR values allow increased position size due to lower expected volatility.

ATR is also useful for establishing profit targets and stop losses. One approach is to set stop losses at some multiple of the ATR below entry price, such as 2x or 3x the ATR value which accommodates market fluctuations.

A screenshot of a computer code

Description automatically generated

We calculate each of the three ranges from the imported data from Yahoo Finance.

To calculate the previous periods close, we use the shift() function which moves each value one step back in time. Then, to calculate True Range, we use axis = 1, to calculate max values along the rows, rather than columns. Additionally, we use skipna, to discard that row in our maximum value calculation.

We calculate the Average True range by using exponential moving average. We set our min\_periods so that it calculates moving average only when given that many rows of data. Additionally, we use the span function, which is calculated as follows, according:

### Bollinger Bands

Bollinger Bands is another volatility-based indicator to measure market volatility and identify overbought or oversold conditions.

Bolinger bands comprise of 2 lines plotted n standard deviations from a m period SMA line.

Typically, the lower band and upper band use a value of 2 standard deviations. Moreover, the middle band is typically a 20 period, simple moving average line, which provides a baseline for the stock’s price action.

During periods of low volatility, the Bollinger Bands contract as the bands move closer together. This signals uncertainty in the market direction. During high volatility, the Bollinger Bands expand as the bands widen. This indicates accelerated market movements.

Bollinger Bands also indicates overbought or oversold levels. The closer the price moves to the upper bands, the more overbought a security becomes. Additionally, the closer the price moves to the lower bands, the oversold the security has become. Reversion back to the mean towards the middle band often occurs after reaching these extremes.

Strategies by trading using Bollinger Bands include:

1. Fading the bands - Selling near the upper band and buying near the lower band in expectation of a reversion.
2. Riding the bands - Going long near the lower band and closing trades near the upper band to capture an uptrend.
3. Band squeezes - When the bands narrow significantly, a breakout in either direction may occur soon.
4. Walking the bands - Using the bands as dynamic support and resistance levels.

A screenshot of a computer program

Description automatically generated

Here, we simply calculate the SMA for the middle band using the pandas.DataFrame.rolling().mean() function. We do this for a period of 20 days.

Then, we calculate the upper and lower band using their formula for standard deviation, along with no delta degrees of freedom, as this is calculated for the population and not the sample.

We also calculate the Band width, the difference between the upper band and lower band to see how wide they are.

### Relative Strength Index (RSI)

Relative Strength Index (RSI) is a momentum oscillator that measures the speed and rate of price changes. It compares the security's strength on days when prices go up to its strength on days when prices go down.

The formula for RSI is as shown below:

The average gain or loss used in this calculation is the average percentage gain or loss during a look-back period, typically 14 days.

Note: This formula uses a positive value for average loss. And, when calculating the average gain, any periods with price declines are excluded and treated as zero gains. Similarly, when determining the average loss, any periods with price rises are not counted and are considered to have zero losses.

The value of RSI oscillates between 0 and 100. Values above 70 are conventionally considered to be overbought levels, while values below 30 are considered to be oversold levels. Higher RSI levels suggest bullish momentum as there is a greater ratio of average ups than downs. While, lower RSI levels indicate bearish momentum is increasing as downside outpaces upside.

RSI should always be combined with other technical analysis tools for confirmation before acting out on a trade. This is because overbought and oversold levels can persist for long duration of time in strong uptrends and downtrends.

A screenshot of a computer code

Description automatically generated

In our RSI function, we have first calculated the change between the closing price in the current period and previous period.

Thereafter, we compute the gain and loss, using the np.where function. If the change is positive, we factor that in for gain, and if the change is negative, we factor that in for our loss.

Then, we calculate our average gain and average loss using an exponential weighted moving average function. Here, we have defined our decay to be alpha:

Then, we calculate our relative strength using average gain and average loss, which is used in our relative strength index calculation.

### ADX (Average Directional Index)

The Average Directional Index (ADX) is a technical indicator to measure the strength and momentum of a trend. The ADX simply tells traders about the strength of a trend, that is, whether a market is trending or consolidating. It does not make any inference on the direction of the trned.

The ADX is derived from the Directional Movement Index (DMI) which consists of the +DI and -DI lines. The +DI measures upward trend momentum by comparing successive highs while the -DI measures downward trend momentum by comparing successive lows. High +DI indicates strong uptrend while high -DI signals strong downtrend.

The values of ADX ranges from 0 to 100 and quantifies the strength of a trend. Values below 25 typically have a weak trend, while those above 75 have an extremely strong trend. Values between 25 and 75 will have a strong trend.

A screenshot of a computer program

Description automatically generated

Here, we first calculate the upmoves and downmoves by looking at successive values between rows, leveraging the shift command.

Then, we calculate the positive directional movement, only if the upmove is greater than downmove and that the upmove is positive. Likewise, we calculate the negative directional movement, only if the downmove is greater than upmove and that the downmove is positive.

Lastly, we calculate the ADX value using the below equation:

### Renko chart

Renko charts are a type of financial chart that plots price movements in fixed increments called "bricks" instead of time-based increments like minutes or hours. Renko charts remove noise and can highlight trends more clearly.

Renko bricks are always equal in size. A new brick is plotted when the price moves by the predefined brick size. For example, in a 100 point Renko chart, a new brick is added every time the price moves ±100 points.

Uptrends are shown as green bricks when the price rises by the brick size. Downtrends plot red bricks when the price falls by the brick size. Consolidation and small counter-trend moves that do not reach the brick size are filtered out.

Without time on the X-axis, Renko charts reflect pure price action and trends. Since the bricks are uniform in size, the intensity and momentum of trends become visually apparent based on the angle of ascent or descent.

A screenshot of a computer program

Description automatically generated

We will collect both 5-minute data over 1 month period and hourly data over 1 year period and store it in ohlcv\_data and hour\_data dictionary. We will also create another dictionary, renko\_data, which is empty for now, but will be used to store out results.

Since we are using the stocktrends library, we rename “Adjusted Close” to “Close” to match its specifications and create a Renko object called df2.

The brick size for our renko chart will be 3 times the ATR of hourly data. This is not too small that the chart will be choppy and prone to whipsaws, nor is it to large to lag the trend and filter important information.

## e) Implementing Performance Measurements

Before implementing strategies, we should backtest strategies and measure expected performance of any trading strategy by testing it on historical data and trying to mimic actual trading conditions.

We measure various key performance indicators that measure both risk and return characteristic of the strategy. We have implemented the following measures:

* Cumulative Annual Growth Rate
* Annualized Volatility (Standard Deviation)
* Sharpe Ratio
* Sortino Ratio
* Maximum Drawdown
* Calmar Ratio

### CAGR

CAGR is the annual rate of return realized by an asset or portfolio to reach its current market value from its initial value. This helps us compare different trading strategy.

The formula is as shown below:

A screenshot of a computer program

Description automatically generated

### Annualized Volatility

In finance, the volatility of a strategy is represented by standard deviation of the returns. We try to capture the variability around this mean daily return.

We should standardize this volatility so we can compare across different timeframes. This is done by multiplying it with an annualization factor:

1. Daily volatility is annualized by multiplying with (number of trading days in a year)
2. Weekly volatility is annualized by multiplying with (number of trading weeks in a year)
3. Monthly volatility is annualized by multiplying with (number of trading months in a year)

However, it is to be noted that this assumes returns are normally distributed and does not capture tail risk:

A screenshot of a computer code

Description automatically generated

### Sharpe Ratio

Sharpe ratio measures the average return obtained in excess of riskfree rate of return per unit of volatility. This is widely used as the risk-adjusted return.

It is computed as follows:

The higher the Sharpe ratio, the better the risk-adjusted performance. A higher return with lower volatility results in a greater Sharpe ratio which is desirable. A negative Sharpe ratio occurs when the risk-free rate exceeds the portfolio's return indicating underperformance on a risk-adjusted basis.

Typically, a Sharpe ratio greater than 1 is considered good. A Sharpe ratio greater than 2 is considered very good and those greater than 3 is excellent.

### Sortino Ratio

This is similar to the Sharpe ratio, which takes into account standard deviation of negative returns because it should consider only “harmful volatility”. The equation is same as Sharpe ratio, but simply the denominator is replaced.

A screenshot of a computer program

Description automatically generated

### Maximum Drawdown

Maximum drawdown is the largest percentage drop measured in asset price. It is simply the distance between peak and trough in the line curve of the asset. Maximum drawdown tells you whether you remain solvent or not with a given strategy. This is particularly important if you are using leverage.

Important thing to note is that you should compare maximum drawdown of two strategies only over the same time horizon, as the drawdown changes over a longer back testing period.

A computer code with many colorful text

Description automatically generated with medium confidence

### Calmar Ratio

Calmar ratio is the ratio of the CAGR and Max drawdown and is another measure of risk-adjusted return.

The higher the Calmar ratio, the better since higher returns are generated relative to drawdown risk taken. Generally, ratios above 1 are considered acceptable. Ratios above 2 are very good. Ratios above 5 are seen as exceptional.

A screen shot of a computer code

Description automatically generated

## f) Implementing Trading Strategy

### Calmar Ratio

In this strategy, we choose a portfolio of stocks based on the stock universe. We build a long-only portfolio with fixed individual position sizes of m number of stocks. Our main criteria is picking m number of stocks based on monthly returns. We rebalance our portfolio every month by removing the worse x stocks and replace them with top x stocks from stock universe. Then, we backtest the strategy and compare the key metrics with a buy and hold strategy.

Note: We can select existing stocks to replace worse performing stocks should they have high returns.

A screenshot of a computer

Description automatically generated

We have chosen the DJI constituent stocks as our stock universe. We select those stocks in 2013 to eliminate any possibility of survivorship bias. We will download data over the period of 10 years, for an interval of 1 month.

We also drop any rows with NA values.

A screenshot of a computer code

Description automatically generated

We first calculate the monthly returns of our stock universe and store it in the return\_df dataframe.

Then, we define our portfolio of stocks, namely:

1. 3M
2. American Express
3. AT&T
4. The Boeing Company
5. Caterpillar Inc.
6. Cisco Systems, Inc.

A screenshot of a computer program

Description automatically generated

We then calculate our portfolio iteratively. We store an empty list as our portfolio at first and a monthly return list, initialized as o.

We calculate the mean monthly return of our portfolio and append it to the list. Then, we calculate the bad stocks in our portfolio based on their monthly return and remove them from our portfolio. Then, we pick new stocks from the stock universe and add it back to our portfolio.

For our portfolio, we obtain the following results:

CAGR: 0.111

Sharpe ratio: 0.524

Maximum Drawdown: 0.270

For our index, we obtain the following results:

CAGR: 0.087

Sharpe ratio: 0.4158

Maximum Drawdown: 0.232

This is the final graph comparing the index and our trading strategy:

A graph of a graph of a return and a strategy

Description automatically generated with medium confidence

### Resistance Breakout

In this strategy, we are looking at a resistance breakout:

Resistance breakout is a well-known strategy where we see the price of the stock breaching a previous resistance level. Our breakout rule in this case will be a price breaching 20 period rolling max/min price used

in conjunction with a volume breaching rolling maximum volume. Using this, we go long/short stocks based on the signal. Additionally, we define exit/stop loss signal with a previous price plus/minus 20 period ATR as the rolling stop loss price. Finally, we backtest the strategy and calculate the cumulative return for each stock.