Algorithmic Trading (Equal Weights S&P500)

1. Introduction

This is my first personal project on algorithmic trading.

The strategy for this project is assigning equal weights to all the companies in the S&P500 (a static list).

The hypothesis for this strategy is that smaller-cap stocks may have more room for growth or are more likely to be mispriced, compared to the mega-cap stocks that dominate the S&P 500's market-cap-weighted index.

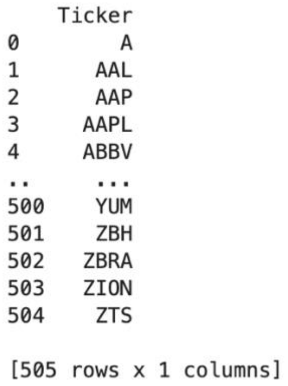
How this hypothesis may work:

1. Diversification of Risk: Equal-weighting reduces the concentration of risk by spreading it evenly across all 500 companies, limiting the dominance of large-cap stocks.
2. Smaller-Cap Exposure: The strategy increases exposure to mid- and small-cap stocks, potentially leading to higher returns compared to large-cap stocks over certain periods.

The output of this project is a excel sheet recommending the number of shares to buy for each S&P500 company.

2. Methodology

- Data processing
    - Loaded a csv file of 505 S&P500 companies

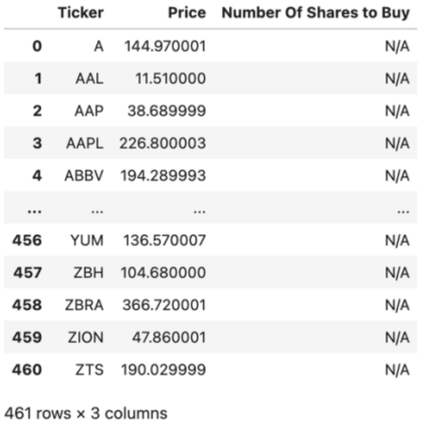


```
    Ticker
0        A
1      AAL
2      AAP
3     AAPL
4     ABBV
..     ...
500    YUM
501    ZBH
502   ZBRA
503   ZION
504    ZTS

[505 rows x 1 columns]
```

Figure 1: Static List of S&P500 Companies

-
    - Obtain price quotes of each company using yfinance library and created a DataFrame to store the data

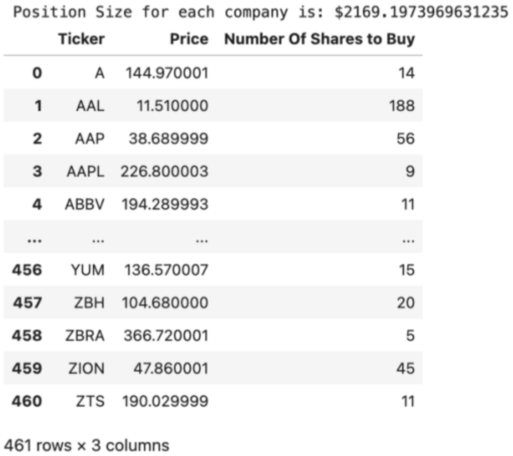


	Ticker	Price	Number Of Shares to Buy
0	A	144.970001	N/A
1	AAL	11.510000	N/A
2	AAP	38.689999	N/A
3	AAPL	226.800003	N/A
4	ABBV	194.289993	N/A
...	...	...	...
456	YUM	136.570007	N/A
457	ZBH	104.680000	N/A
458	ZBRA	366.720001	N/A
459	ZION	47.860001	N/A
460	ZTS	190.029999	N/A

461 rows × 3 columns

Figure 2: Data frame of Tickers and their last quoted price

- User Interface
    - Using input function of python, users can input their portfolio size
    - Calculated position size by dividing portfolio size with number of companies
    - Calculated number of shares to buy of respective companies by dividing position size with stock price



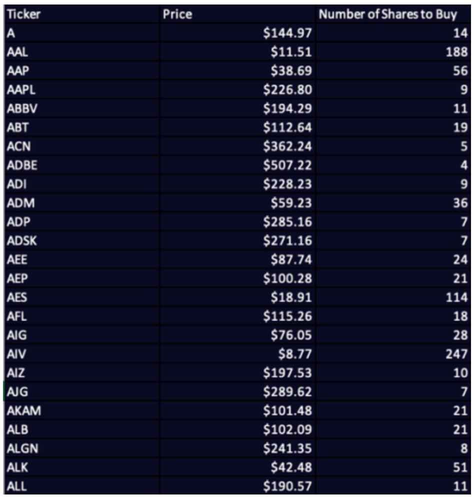
```
Position Size for each company is: $2169.1973969631235
```

	Ticker	Price	Number Of Shares to Buy
0	A	144.970001	14
1	AAL	11.510000	188
2	AAP	38.689999	56
3	AAPL	226.800003	9
4	ABBV	194.289993	11
...	...	...	...
456	YUM	136.570007	15
457	ZBH	104.680000	20
458	ZBRA	366.720001	5
459	ZION	47.860001	45
460	ZTS	190.029999	11

461 rows × 3 columns

Figure 5: Data Frame with updated data on number of shares to buy

- Saving output into an excel file
    - By using xlsxwriter library in python, an excel sheet with the recommended trades will be created and saved



Ticker	Price	Number of Shares to Buy
A	$144.97	14
AAL	$11.51	188
AAP	$38.69	56
AAPL	$226.80	9
ABBV	$194.29	11
ABT	$112.64	19
ACN	$362.24	5
ADBE	$507.22	4
ADI	$228.23	9
ADM	$59.23	36
ADP	$285.16	7
ADSK	$271.16	7
AEE	$87.74	24
AEP	$100.28	21
AES	$18.91	114
AFL	$115.26	18
AIG	$76.05	28
AIV	$8.77	247
AIZ	$197.53	10
AJG	$289.62	7
AKAM	$101.48	21
ALB	$102.09	21
ALGN	$241.35	8
ALK	$42.48	51
ALL	$190.57	11

Figure 6: Excel Sheet of recommended trades

- Backtesting Strategy
    - Using Backtrader Library, I backtested my ‘Equal Weights S&P500’ strategy using 10 years of historical data from yfinance.
    - The strategy works by rebalancing the portfolio (static S&P500 companies) every 60 days
        - 1st day, buy equal weights of all S&P500 companies
        - 60th day, rebalance portfolio by selling and re-buying the S&P500 companies with updated weights
        - Repeat for 10 years of historical data from 2014-10-01 to 2024-10-01
    - Performance of Strategy:
        - Initial Portfolio Value: 1000000
        - Final Portfolio Value: 2709823.0699328184
        - Sharpe Ratio: 0.7308558219583783
        - Max Drawdown: 37.88724372604305%
        - Annual Return: 10.103517528253336%
        - Total Trades: 19802
        - Winning Trades: 11530
        - Losing Trades: 7812
    - Key Takeaways and Recommendations:
        - **Strong Performance**: The strategy delivers solid performance with a decent annual return (10.1%) and a Sharpe ratio of 0.73. This indicates that the strategy is generally risk-efficient and profitable.
        - **High Drawdown**: The max drawdown of nearly 38% is something to be cautious about. Reducing drawdowns can make the strategy more appealing to investors with lower risk tolerance.
        - **High Number of Trades**: With over 19,000 trades, transaction costs could significantly impact real-world performance, especially if they weren't factored in. Look into optimizing this to reduce costs without hurting returns.

        - **Solid Win Rate**: A 58.2% win rate means that most of the time, the strategy is profitable. The next step would be to examine the profit-to-loss ratio to ensure we are making more on winners than losing on losers.
    - Areas for Improvement:
        - **Risk Management**: Consider using tighter stop losses or hedging techniques to manage drawdowns and reduce risk.
        - **Transaction Costs**: If not already included, model the transaction costs to see how they affect the strategy's profitability.
        - **Rebalance Frequency**: Check if rebalancing less frequently could still maintain returns while reducing costs and drawdowns.

3. Learning Summary

- Algorithmic Trading Process is broken down generally into these steps:
    - Collecting Data
    - Developing a hypothesis for a strategy
    - Backtesting the strategy
    - Implement strategy in production* (did not do for this project)
- **How I can improve this project:**
    - Purchasing an API to get real-time list of S&P500 companies
    - Purchasing an API that allows for batch calls to obtain last quoted price(optimises the speed at which the data is collected)