1622 Assignment 1

Angela Hsu

1003328874

Feb 15

Angela Hsu
1003328874

1622 Assignment

The initial portfolio value is 1000070.06 USD and the initial cash account is zero USD. In the initial portfolio, there are 980 shares of IBM and 20000 shares of BK.

Four portfolio re-balancing strategies are implement as following:

(1) **"Buy and hold" strategy**: hold initial portfolio for the entire investment horizon of 2 years. 980 shares of IBM and 20000 shares of BK are held from the beginning to the end.

```
def strat_buy_and_hold(x_init, cash_init, mu, Q, cur_prices):
    x_optimal = x_init
    cash_optimal = cash_init
    return x_optimal, cash_optimal
```

(2) **"Equally weighted" (also known as "1/n") portfolio strategy**: assets weights are equal and remain as the following.

$$w_i^t = \frac{1}{20}, \text{ where } w_i^t = \frac{v_i^t \times x_i^t}{v^t}$$

```
def strat_equally_weighted(x_init, cash_init, mu, Q, cur_prices):
    # Value of the portfolio
    portfolio_value = np.dot(cur_prices,x_init) + cash_init
    # Value of each asset
    asset_value = np.ones((20)) * portfolio_value / 20
    # Optimal shares of each asset
    x_optimal = np.floor(asset_value / cur_prices)
    # 0.5% of the value get from selling or buying
    transaction_cost = np.dot(cur_prices , abs(x_optimal-x_init)) * 0.005
    cash_optimal = portfolio_value - np.dot(cur_prices,x_optimal) - transaction_cost
    return x_optimal, cash_optimal
```

Since portfolio value is re-evaluated every two-month, it is calculated as the summation of current price of each stock multiplied by its optimal shares determined by last period.

$$\text{portfolio value}^t = \sum_{i=1}^{20} x_i^{t-1} \times \text{cur\_prices}_i^t + \text{cash}^t, \text{ where } t = 1,2,\ldots,12$$

The money allocated to each asset is determined by the current portfolio value divided by the number of assets which is 20.

$$\text{asset value}^t = \text{portfolio value}^t \div 20, \text{ where } t = 1,2,\ldots,12$$

The new optimal shares for each stock is determined by the money allocation divided by current stock price for each asset, the result of which is rounded down to an integer.

$$x_i^t = \text{round down } (\text{asset value}_i^t \div \text{cur\_prices}_i^t) \text{ for } i = 1,2,\ldots,20$$

Angela Hsu
1003328874

(3) **"Minimum variance" portfolio strategy**: minimizing the total risk of portfolio which is determined by variance.

$$\text{Minimize: } Z = \sum_{i,j=1}^{20} x_i x_j \sigma_{ij} = x^T Q x$$

$$\text{Subject to: } \sum_{i=1}^{20} \mu_i x_i \geq \varepsilon_R$$

$$\sum_{i=1}^{20} x_i = 1$$

$$x_i \geq 0 \quad \text{for } i = 1, 2, 3 \ldots, 20 \text{ (without short selling)}$$

The objective is to minimize the variance determined by x^TQx, in which x represents weights and Q represents variance-covariance matrix for 20 stocks. The constraint is that sum of weights equal to 1 and no negative weights allowed.

```
def strat_min_variance(x_init, cash_init, mu, Q, cur_prices):
    # initialize the CPLEX object
    cpx = cplex.Cplex()
    cpx.objective.set_sense(cpx.objective.sense.minimize)
    # No linear part of objective function
    c = np.zeros((20))
    # Define bounds on variables (lower bounds default as zeros)
    ub = np.ones((20))
    # Define constraint matrix A
    # Sum of weights: w1+w2+w3+....+w20 = 1
    # No shorting: wi >= 0
    A = []
    n = 20
    for i in range(n):
        A.append([[0,1],[1,0]])
    # Add constraints to CPLEX model
    # Sum of weights equal to one and no shorting allowed
    cpx.linear_constraints.add(rhs=[1.0,0],senses='EG')
    names = ['x_%s'% i for i in range(1,21)]
    cpx.variables.add(obj=c,ub=ub,columns=A,names=names)
    # Define and add quadratic part of objective function
    qmat = [[list(range(n)),list(2*Q[k,:])] for k in range(n)] # Q covariance matrix
    cpx.objective.set_quadratic(qmat)
    # No output from CPLEX
    cpx.set_results_stream(None)
    cpx.set_log_stream(None)
    cpx.set_warning_stream(None)
    cpx.set_error_stream(None)
    cpx.solve()
    # Store the solutions into weights
    weights = np.array(cpx.solution.get_values())
    portfolio_value = np.dot(cur_prices,x_init) + cash_init # Value of the portfolio
    asset_value = weights * portfolio_value # Value of each asset
    x_optimal = np.floor(asset_value / cur_prices) # Optimal shares of each asset
    transaction_cost = np.dot(cur_prices , abs(x_optimal-x_init)) * 0.005 # 0.5% of the value of net shares
    cash_optimal = portfolio_value - np.dot(cur_prices,x_optimal) - transaction_cost
    return x_optimal, cash_optimal
```

After weights are determined by CPLEX, multiplying weights by portfolio value results in assets value.

$$\text{asset value}_i^t = \text{portfolio value}^t \times w_i^t, \text{ where } t = 1,2,\ldots,12$$

The new optimal shares for each stock is determined by the money allocation divided by current stock price for each asset, the result of which is rounded down to an integer.

$$x_i^t = \text{round down } (\text{asset value}_i^t \div \text{cur\_prices}_i^t) \text{ for } i = 1,2,\ldots,20$$

(4) **"Maximum Sharpe ratio" portfolio strategy**: maximizing Sharpe ratio for each period.

$$\text{Minimize: } Z = \sum_{i,j=1}^{20} y_i y_j \sigma_{ij} = y^T Q y$$

$$\text{Subject to: } \sum_{i=1}^{20} (\mu_i - rf) y_i = 1$$

$$\sum_{i=1}^{20} y_i = k$$

$$lk \leq Ay \leq uk$$

$$k \geq 0$$

To be specific, the objective is to minimize $y^T Q y$, in which variable y with dimension 20×1 and variance-covariance matrix Q with dimension 21×21. The constraint is that the summation of the difference between expected return of each stock and risk-free rate multiplied by variable yi equal to 1. Moreover, sum of yi equal to k and variable k greater or equal to zero.

The result computed by CPLEX is variable y with n dimension and variable k. Weights for each stock is computed by the following equation.

$$w_i^t = y_i^t \div k$$

After weights are calculated, multiplying weights by portfolio value results in assets value.

$$\text{asset value}_i^t = \text{portfolio value}^t \times w_i^t$$

The new optimal shares for each stock is determined by the money allocation divided by current stock price for each asset, the result of which is rounded down to an integer.

$$x_i^t = \text{round down } (\text{asset value}_i^t \div \text{cur\_prices}_i^t) \text{ for } i = 1,2,\ldots,20$$

Angela Hsu
1003328874

```
def strat_max_Sharpe(x_init, cash_init, mu, Q, cur_prices):
    n = 21 # 21 variables
    r_rf = 0.025 # risk-free rate
    daily_rf = 0.025 / 252 # daily risk-free rate
    # Matrix Q is 21x21 dimension
    Q1 = np.append(Q,np.zeros((20,1)),axis=1)
    new_row = np.zeros((21))
    Q2 = np.vstack([Q1,new_row]) #add new column and new row for a new variable k
    diff = mu-daily_rf * np.ones(20) # difference between expected return and rf rate
    # Define constraint matrix A
    # Summation of (mu-rf)yi = 1
    # Summation of yi - k = 0
    A = []
    for i in range(n-1):
        A.append([[0,1],[diff[i],1.0]])
    A.append([[0,1],[0,-1.0]])

    # initialize the CPLEX object
    cpx = cplex.Cplex()
    cpx.objective.set_sense(cpx.objective.sense.minimize)

    # Add objective function, bounds on variables and constraints to CPLEX model
    # Sum of (mu-daily_rf)*yi equal to one and Sum of weights equal to one
    c = np.zeros((n))
    ub = [np.inf]*n
    cpx.linear_constraints.add(rhs=[1.0,0],senses='EE')
    names = ['y_%s'% i for i in range(1,n+1)]
    cpx.variables.add(obj=c,ub=ub,columns=A,names=names)
    qmat = [[list(range(n)),list(2*Q2[k,:])] for k in range(n)] # Q variance-covariance matrix
    cpx.objective.set_quadratic(qmat)

    # No output from CPLEX
    cpx.set_results_stream(None)
    cpx.set_log_stream(None)
    cpx.set_warning_stream(None)
    cpx.set_error_stream(None)
    cpx.solve()
    result = np.array(cpx.solution.get_values())
    weights = result[0:20]/result[20] # wi = yi/k
```

```
    portfolio_value = np.dot(cur_prices,x_init) + cash_init # Value of the portfolio
    asset_value = weights * portfolio_value # Value of each asset
    x_optimal = np.floor(asset_value / cur_prices) # Optimal shares of each asset
    transaction_cost = np.dot(cur_prices , abs(x_optimal-x_init)) * 0.005 # 0.5% of the
    cash_optimal = portfolio_value - np.dot(cur_prices,x_optimal) - transaction_cost
    return x_optimal, cash_optimal
```

For all strategies, transaction cost for each rebalancing need to be calculated. To be specific, the transaction cost is calculated as 0.5% of the difference between change of shares for each stock multiplying the stock price. The equation is as below.

$$\text{transaction cost}_i^t = (\text{stock price}_i^t \times |x_i^t - x_i^{t-1}|) * 0.005$$

Moreover, the optimal cash for each period is calculated as the remain of portfolio value with the optimal shares for last period minus the portfolio value with new optimal shares minus transaction cost.

$$\text{optimal cash}_i^t = \text{portfolio value}^{t-1} - \text{portfolio value}^t - \sum_{i=1}^{20} \text{transaction cost}_i$$

Angela Hsu
1003328874

For the body function about checking non-negative cash, if cash is negative, optimal shares should be re-evaluated. Firstly, the portion for each stock multiplied the negative cash results in the cash that need to save for each stock. Dividing the cash by stock price results in the shares that need to be sold, which is rounded up to integers. New optimal shares is calculated by minus the original shares by the shares that need to be sold. And then the transaction cost is calculated as 0.5% of the value of net shares for each stock. New cash is calculated by minus the original portfolio value by current portfolio value and transaction cost.

```
###################### Insert your code here ############################
if cash[strategy, period-1] < 0:
    # Check how much value the current portfolio is
    portfolio_value = np.dot(cur_prices,curr_positions) + curr_cash
    # The portion for each stock
    portion = x[strategy, period-1]/sum(x[strategy, period-1])
    # The cash that need to save for each stock
    cash = abs(cash[strategy, period-1])*portion
    # The shares that need to sell for each stock, which is rounded up to integers
    shares = np.ceil(cash/cur_prices)
    # The new optimal shares for each stock is the difference between original shares and the shares
    # that need to be sold
    x[strategy, period-1] = x[strategy, period-1] - shares
    # Transaction cost is 0.5% of the value of net shares for each stock
    new_tran_cost = np.dot(cur_prices , abs(x[strategy, period-1]-curr_positions)) * 0.005
    # The new optimal cash is the remain by minus original portfolio value
    # by current portfolio value and transaction cost
    cash[strategy, period-1] = portfolio_value - np.dot(cur_prices,x[strategy, period-1]) - new_tran_cost
```

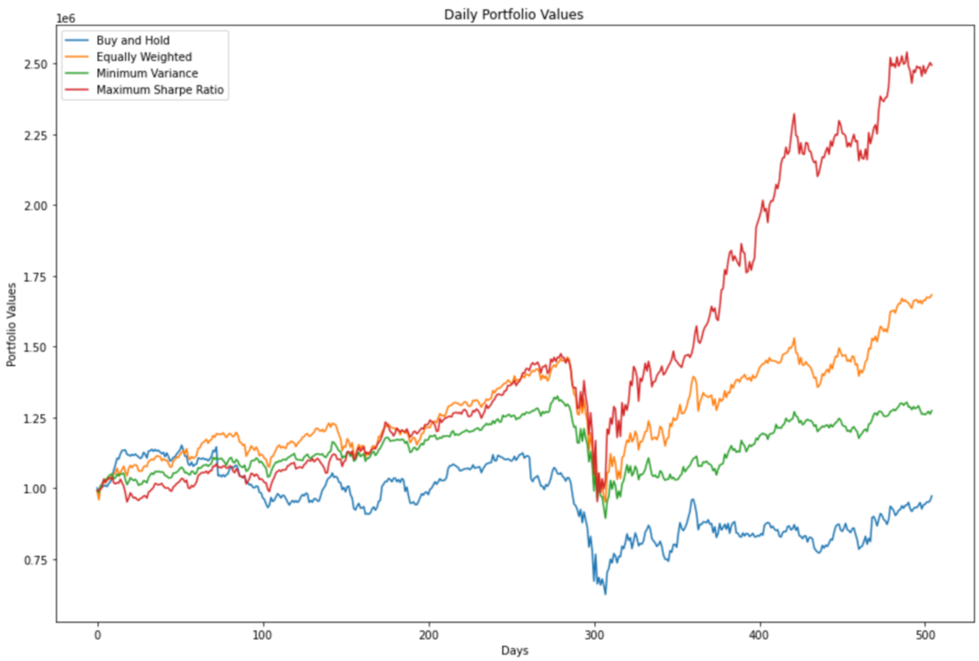
Angela Hsu
1003328874

Output:

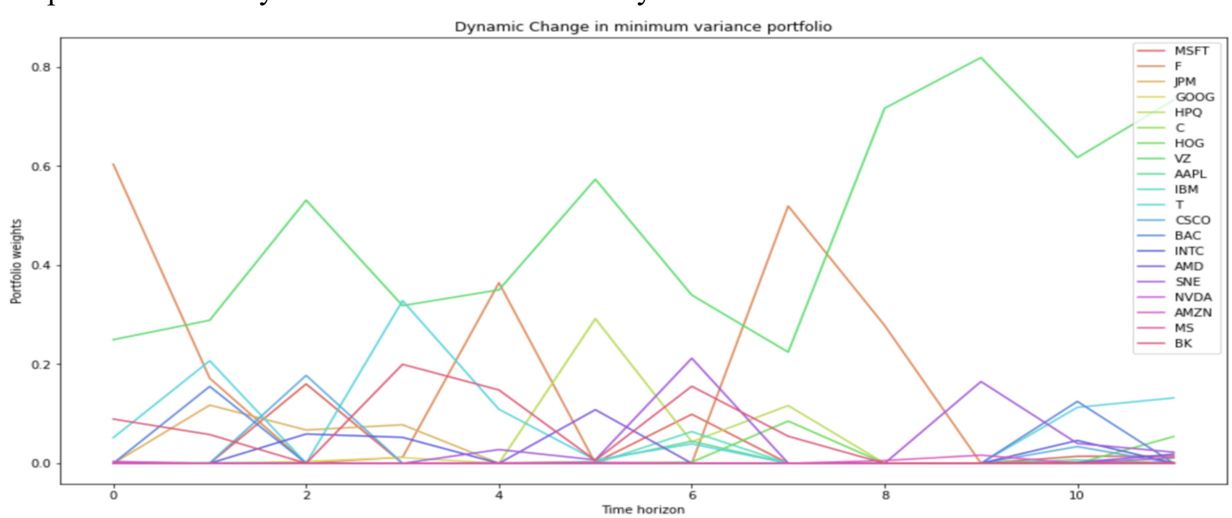
```
Initial portfolio value = $ 1000070.06


Period 1: start date 01/02/2019, end date 02/28/2019
  Strategy "Buy and Hold", value begin = $ 1000070.06, value end = $ 1121179.83
  Strategy "Equally Weighted Portfolio", value begin = $ 991124.38, value end = $ 1097031.81
  Strategy "Mininum Variance Portfolio", value begin = $ 991702.28, value end = $ 1057440.44
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 990119.39, value end = $ 1016524.41

Period 2: start date 03/01/2019, end date 04/30/2019
  Strategy "Buy and Hold", value begin = $ 1126131.27, value end = $ 1075001.89
  Strategy "Equally Weighted Portfolio", value begin = $ 1103260.47, value end = $ 1188731.33
  Strategy "Mininum Variance Portfolio", value begin = $ 1055378.90, value end = $ 1107930.67
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1007118.10, value end = $ 1076636.20

Period 3: start date 05/01/2019, end date 06/28/2019
  Strategy "Buy and Hold", value begin = $ 1070867.54, value end = $ 969057.81
  Strategy "Equally Weighted Portfolio", value begin = $ 1181234.03, value end = $ 1169139.09
  Strategy "Mininum Variance Portfolio", value begin = $ 1091907.85, value end = $ 1099494.27
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1060338.50, value end = $ 1073404.96

Period 4: start date 07/01/2019, end date 08/30/2019
  Strategy "Buy and Hold", value begin = $ 976973.31, value end = $ 933721.61
  Strategy "Equally Weighted Portfolio", value begin = $ 1179634.22, value end = $ 1149869.96
  Strategy "Mininum Variance Portfolio", value begin = $ 1097336.69, value end = $ 1129311.06
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1071286.10, value end = $ 1140344.20

Period 5: start date 09/03/2019, end date 10/31/2019
  Strategy "Buy and Hold", value begin = $ 922211.42, value end = $ 1028337.74
  Strategy "Equally Weighted Portfolio", value begin = $ 1138167.02, value end = $ 1252745.95
  Strategy "Mininum Variance Portfolio", value begin = $ 1115582.54, value end = $ 1182495.62
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1139157.61, value end = $ 1246555.84

Period 6: start date 11/01/2019, end date 12/31/2019
  Strategy "Buy and Hold", value begin = $ 1037933.42, value end = $ 1099403.03
  Strategy "Equally Weighted Portfolio", value begin = $ 1270461.87, value end = $ 1373479.86
  Strategy "Mininum Variance Portfolio", value begin = $ 1184511.14, value end = $ 1255872.45
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1248630.54, value end = $ 1370124.66


Period 7: start date 01/02/2020, end date 02/28/2020
  Strategy "Buy and Hold", value begin = $ 1112112.69, value end = $ 900207.54
  Strategy "Equally Weighted Portfolio", value begin = $ 1396296.22, value end = $ 1258330.19
  Strategy "Mininum Variance Portfolio", value begin = $ 1256164.22, value end = $ 1159346.53
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1379534.38, value end = $ 1284831.08

Period 8: start date 03/02/2020, end date 04/30/2020
  Strategy "Buy and Hold", value begin = $ 924774.25, value end = $ 856285.51
  Strategy "Equally Weighted Portfolio", value begin = $ 1312225.31, value end = $ 1215208.23
  Strategy "Mininum Variance Portfolio", value begin = $ 1209653.03, value end = $ 1077520.12
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1340554.90, value end = $ 1417410.10

Period 9: start date 05/01/2020, end date 06/30/2020
  Strategy "Buy and Hold", value begin = $ 822532.65, value end = $ 875128.45
  Strategy "Equally Weighted Portfolio", value begin = $ 1171040.04, value end = $ 1316082.51
  Strategy "Mininum Variance Portfolio", value begin = $ 1046783.13, value end = $ 1081097.46
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1357711.42, value end = $ 1638069.98

Period 10: start date 07/01/2020, end date 08/31/2020
  Strategy "Buy and Hold", value begin = $ 852159.31, value end = $ 852474.32
  Strategy "Equally Weighted Portfolio", value begin = $ 1307022.80, value end = $ 1493983.95
  Strategy "Mininum Variance Portfolio", value begin = $ 1084116.26, value end = $ 1242140.70
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1699166.66, value end = $ 2238757.92

Period 11: start date 09/01/2020, end date 10/30/2020
  Strategy "Buy and Hold", value begin = $ 857122.42, value end = $ 795062.75
  Strategy "Equally Weighted Portfolio", value begin = $ 1504676.72, value end = $ 1407362.52
  Strategy "Mininum Variance Portfolio", value begin = $ 1244341.51, value end = $ 1193122.73
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 2286127.06, value end = $ 2163972.69

Period 12: start date 11/02/2020, end date 12/31/2020
  Strategy "Buy and Hold", value begin = $ 811070.20, value end = $ 972162.37
  Strategy "Equally Weighted Portfolio", value begin = $ 1419803.51, value end = $ 1682239.09
  Strategy "Mininum Variance Portfolio", value begin = $ 1203779.19, value end = $ 1273097.92
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 2162013.24, value end = $ 2492775.63
```

Angela Hsu
1003328874

According to the chart following, “Maximum Sharpe Ratio” portfolio produces best results and “Equally Weighted” portfolio the next. The former one has doubled portfolio value in two years while the latter one increases portfolio value by about 150%. On the contrary, “Minimum Variance” portfolio remains almost the same in two years while “Buy and Hold” portfolio has depreciated portfolio value. All four portfolio has a common performance during 300th day when their portfolio value pull down a lot.

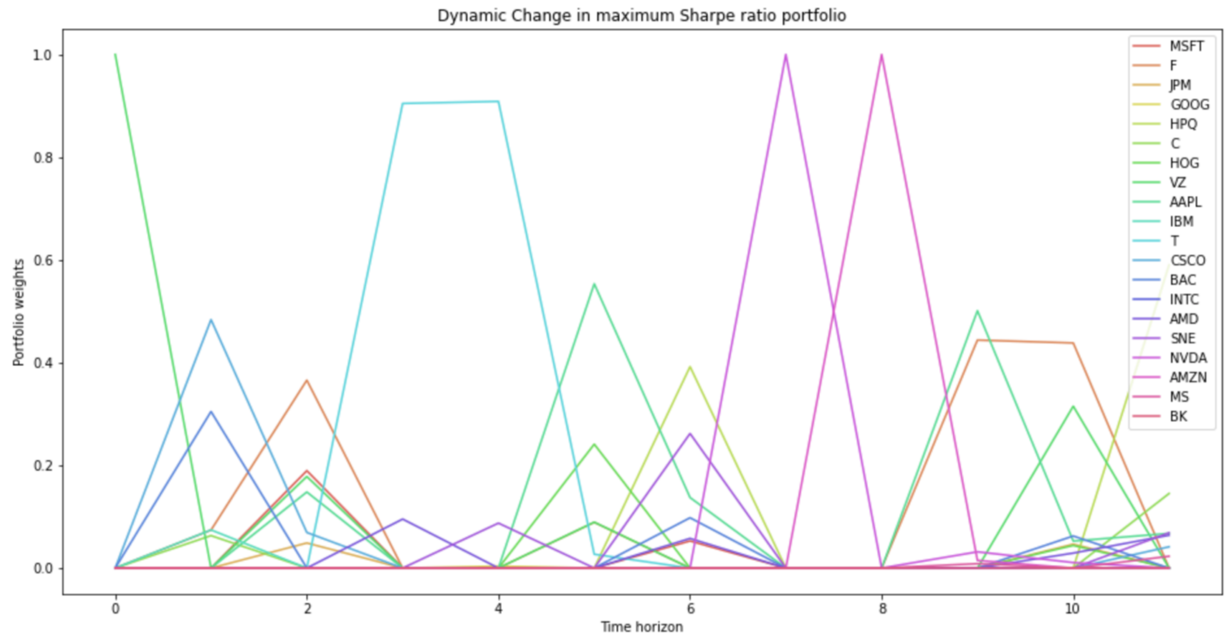


Dynamic changes in “Minimum Variance” portfolio is shown as below. Obviously, HOG has a big portfolio weight (≥ 20%) constantly while F only occupies big portfolio weight for some period. BK and T are held in the first 6 period but sold during next 6 period. It can be seen that “Minimum Variance” portfolio is more about minimizing risk instead of maximizing return, so the portfolio has only about 20% of return in two years.



Angela Hsu
1003328874

Dynamic changes in “Maximum Sharpe Ratio” portfolio is shown as below. The portfolio change is big for every two period. For example, T is held for 2-4 period; NVDA is held for 6-8 period; AMZN is held 7-8 period; F is held 8-12 period. The portfolio weight for each stock change a lot for each period. To maximize return for given risk, the portfolio will predict which stock may have biggest return and hold it with large weight and predict which stock may have lowest return and hold it with little weight.



According to the daily portfolio values, I would choose “Maximum Sharpe Ratio” portfolio due to its return. It is evident that minimizing variance is far more than enough because I would basically have the return that can barely cover the inflation. Since Sharpe Ratio has taken expected return and risk into account, it is the most important indicator for investment.

Discussion

Testing one more strategy “Equally Weighted Always”, I found that the new strategy produces better results. Maybe rebalancing 20 stocks every period does require lots of transaction cost. As stock price increases, the transaction cost also increases. Therefore, saving transaction cost every two-month is necessary.

I suggest that “Maximum Sharpe Ratio” portfolio could implement re-balancing with less frequency. Re-balancing every 6 month may produce better result.

Angela Hsu
1003328874

Output:

```
Period 1: start date 01/02/2019, end date 02/28/2019
  Strategy "Buy and Hold", value begin = $ 1000070.06, value end = $ 1121179.83
  Strategy "Equally Weighted Portfolio", value begin = $ 991124.38, value end = $ 1097031.81
  Strategy "Mininum Variance Portfolio", value begin = $ 991702.28, value end = $ 1057440.44
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 990119.39, value end = $ 1016524.41
  Strategy "Equally Weighted Always", value begin = $ 991578.79, value end = $ 1087997.58

Period 2: start date 03/01/2019, end date 04/30/2019
  Strategy "Buy and Hold", value begin = $ 1126131.27, value end = $ 1075001.89
  Strategy "Equally Weighted Portfolio", value begin = $ 1103260.47, value end = $ 1188731.33
  Strategy "Mininum Variance Portfolio", value begin = $ 1055378.90, value end = $ 1107930.67
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1007118.10, value end = $ 1076636.20
  Strategy "Equally Weighted Always", value begin = $ 1093760.45, value end = $ 1168273.75

Period 3: start date 05/01/2019, end date 06/28/2019
  Strategy "Buy and Hold", value begin = $ 1070867.54, value end = $ 969057.81
  Strategy "Equally Weighted Portfolio", value begin = $ 1181234.03, value end = $ 1169139.09
  Strategy "Mininum Variance Portfolio", value begin = $ 1091907.85, value end = $ 1099494.27
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1060338.50, value end = $ 1073404.96
  Strategy "Equally Weighted Always", value begin = $ 1160703.91, value end = $ 1152157.37

Period 4: start date 07/01/2019, end date 08/30/2019
  Strategy "Buy and Hold", value begin = $ 976973.31, value end = $ 933721.61
  Strategy "Equally Weighted Portfolio", value begin = $ 1179634.22, value end = $ 1149869.96
  Strategy "Mininum Variance Portfolio", value begin = $ 1097336.69, value end = $ 1129311.06
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1071286.10, value end = $ 1140344.20
  Strategy "Equally Weighted Always", value begin = $ 1162106.54, value end = $ 1135272.98

Period 5: start date 09/03/2019, end date 10/31/2019
  Strategy "Buy and Hold", value begin = $ 922211.42, value end = $ 1028337.74
  Strategy "Equally Weighted Portfolio", value begin = $ 1138167.02, value end = $ 1252745.95
  Strategy "Mininum Variance Portfolio", value begin = $ 1115582.54, value end = $ 1182495.62
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1139157.61, value end = $ 1246555.84
  Strategy "Equally Weighted Always", value begin = $ 1125068.62, value end = $ 1224046.22

Period 6: start date 11/01/2019, end date 12/31/2019
  Strategy "Buy and Hold", value begin = $ 1037933.42, value end = $ 1099403.03
  Strategy "Equally Weighted Portfolio", value begin = $ 1270461.87, value end = $ 1373479.86
  Strategy "Mininum Variance Portfolio", value begin = $ 1184511.14, value end = $ 1255872.45
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1248630.54, value end = $ 1370124.66
  Strategy "Equally Weighted Always", value begin = $ 1240445.22, value end = $ 1338333.10

Period 7: start date 01/02/2020, end date 02/28/2020
  Strategy "Buy and Hold", value begin = $ 1112112.69, value end = $ 900207.54
  Strategy "Equally Weighted Portfolio", value begin = $ 1396296.22, value end = $ 1258330.19
  Strategy "Mininum Variance Portfolio", value begin = $ 1256164.22, value end = $ 1159346.53
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1379534.38, value end = $ 1284831.08
  Strategy "Equally Weighted Always", value begin = $ 1361448.84, value end = $ 1238520.51

Period 8: start date 03/02/2020, end date 04/30/2020
  Strategy "Buy and Hold", value begin = $ 924774.25, value end = $ 856285.51
  Strategy "Equally Weighted Portfolio", value begin = $ 1312225.31, value end = $ 1215208.23
  Strategy "Mininum Variance Portfolio", value begin = $ 1209653.03, value end = $ 1077520.12
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1340554.90, value end = $ 1417410.10
  Strategy "Equally Weighted Always", value begin = $ 1289371.41, value end = $ 1225727.25

Period 9: start date 05/01/2020, end date 06/30/2020
  Strategy "Buy and Hold", value begin = $ 822532.65, value end = $ 875128.45
  Strategy "Equally Weighted Portfolio", value begin = $ 1171040.04, value end = $ 1316082.51
  Strategy "Mininum Variance Portfolio", value begin = $ 1046783.13, value end = $ 1081097.46
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1357711.42, value end = $ 1638069.98
  Strategy "Equally Weighted Always", value begin = $ 1185492.54, value end = $ 1317481.02

Period 10: start date 07/01/2020, end date 08/31/2020
  Strategy "Buy and Hold", value begin = $ 852159.31, value end = $ 852474.32
  Strategy "Equally Weighted Portfolio", value begin = $ 1307022.80, value end = $ 1493983.95
  Strategy "Mininum Variance Portfolio", value begin = $ 1084116.26, value end = $ 1242140.70
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 1699166.66, value end = $ 2238757.92
  Strategy "Equally Weighted Always", value begin = $ 1313933.45, value end = $ 1551040.99

Period 11: start date 09/01/2020, end date 10/30/2020
  Strategy "Buy and Hold", value begin = $ 857122.42, value end = $ 795062.75
  Strategy "Equally Weighted Portfolio", value begin = $ 1504676.72, value end = $ 1407362.52
  Strategy "Mininum Variance Portfolio", value begin = $ 1244341.51, value end = $ 1193122.73
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 2286127.06, value end = $ 2163972.69
  Strategy "Equally Weighted Always", value begin = $ 1566986.96, value end = $ 1432040.01

Period 12: start date 11/02/2020, end date 12/31/2020
  Strategy "Buy and Hold", value begin = $ 811070.20, value end = $ 972162.37
  Strategy "Equally Weighted Portfolio", value begin = $ 1419803.51, value end = $ 1682239.09
  Strategy "Mininum Variance Portfolio", value begin = $ 1203779.19, value end = $ 1273097.92
  Strategy "Maximum Sharpe Ratio Portfolio", value begin = $ 2162013.24, value end = $ 2492775.63
  Strategy "Equally Weighted Always", value begin = $ 1439636.72, value end = $ 1669612.63
```