Move Beyond Traditional Equity Strategies and Embrace the Precision of Data-Driven Indexing. Dive into a streamlined approach, optimizing a portfolio that resonates with the pulse of the NASDAQ-100, all while maximizing efficiency and returns.

# Optimization Report – Integer programming

Submitted on 23rd October

Avi Walyat (aw39578)
Aditya Kumar (ak49768)
Aakash Dhruva (avd667)
Joseph Bridges (jhb2854)

# Optimization Report

## 1. Introduction:

Equity-based financial strategies typically fall under two broad categories: 'active' and 'passive'. A popular passive approach is "indexing", where the objective is to design a portfolio that emulates the performance of a wider market segment or a predefined market index, giving birth to what is termed as an index fund. To create an index fund, we need to decide the number of such stocks, identify those stocks and come up with weighted allocations that can resonate with the NASDAQ-100 index.

In an ideal scenario, one might think of simply buying all n stocks constituting the index in the same proportion as the index itself. However, such a method is not only unwieldy due to the multitude of minor positions but also costly, given the potential frequent rebalancing expenses and the price response to trading activities.

### a. Getting ready (Data pre-processing)

i. **Data Ingestion**: Loaded **stocks2019.csv** and **stocks2020.csv** containing historical stock prices.

ii. **Timestamp Parsing**: For time-series analysis, the date column is identified and set as the index using **identify_date_column().**

iii. **Returns Computation**: Used **pct_change** to derive daily returns for stocks and the index, aiding in portfolio analysis and optimization

### b. How to do (Methodology):

i. **Stock Selection:** Use an integer programming model, guided by a 'similarity matrix' ($\rho$), to select an exact ‘m’ stock out of the available 'n' options, focusing on stocks with the highest correlations in returns.

ii. **Without Stock Selection:** Evaluate the approach where we only calculate weights, running our model as a Mixed Integer Program.

iii. **Portfolio Composition:** Apply linear programming to determine the exact quantity of each chosen stock to be incorporated into the portfolio.

iv. **Performance Evaluation:** Compare the newly formed index fund's performance against the NASDAQ-100 index using out-of-sample data, with varying values of 'm' to assess flexibility and accuracy.

## c. What's the end (Goal):

i. The primary goal of this project is to design an Index fund comprising of 'm' specific stocks that closely mirrors the performance of the NASDAQ-100 index.

ii. For a specified value of 'm' (the total stocks in our index), we need to determine:

1. Pinpoint the stocks that most accurately mirror the collective value of the index's entirety.
2. Calculate the weightage for all 'm' stocks.

| date | NDX | ATVI | ADBE | AMD | ALXN | ALGN | GOOGL | GOOG | AMZN | AMGN | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2019-01-02 | 6360.870117 | 46.350380 | 224.570007 | 18.830000 | 98.050003 | 202.119995 | 1054.680054 | 1045.849976 | 1539.130005 | 182.458298 | ... |
| 2019-01-03 | 6147.129883 | 44.704514 | 215.699997 | 17.049999 | 100.209999 | 184.779999 | 1025.469971 | 1016.059998 | 1500.280029 | 179.681961 | ... |
| 2019-01-04 | 6422.669922 | 46.488358 | 226.190002 | 19.000000 | 106.000000 | 186.710007 | 1078.069946 | 1070.709961 | 1575.390015 | 185.824142 | ... |
| 2019-01-07 | 6488.250000 | 47.799141 | 229.259995 | 20.570000 | 107.940002 | 189.919998 | 1075.920044 | 1068.390015 | 1629.510010 | 188.324738 | ... |
| 2019-01-08 | 6551.850098 | 49.247898 | 232.679993 | 20.750000 | 108.610001 | 192.949997 | 1085.369995 | 1076.280029 | 1656.579956 | 190.739777 | ... |

Figure 1(c-i): Glimpse of returns table for year 2019

| | NDX | ATVI | ADBE | AMD | ALXN | ALGN | GOOGL | GOOG | AMZN | AMGN | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2020-01-02 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... |
| 2020-01-03 | -0.008827 | 0.000341 | -0.007834 | -0.010183 | -0.013260 | -0.011421 | -0.005231 | -0.004907 | -0.012139 | -0.006789 | ... |
| 2020-01-06 | 0.006211 | 0.018238 | 0.005726 | -0.004321 | 0.001598 | 0.019398 | 0.026654 | 0.024657 | 0.014886 | 0.007674 | ... |
| 2020-01-07 | -0.000234 | 0.010043 | -0.000959 | -0.002893 | 0.002533 | -0.009864 | -0.001932 | -0.000624 | 0.002092 | -0.009405 | ... |
| 2020-01-08 | 0.007452 | -0.007623 | 0.013438 | -0.008705 | 0.016191 | 0.010386 | 0.007118 | 0.007880 | -0.007809 | 0.000756 | ... |

Figure 1(c-ii): Glimpse of returns table for year 2020

## d. Identifying stocks

To capture the stocks that most closely mirror the overall index's value, we initially determine the daily return for each stock. Using these yields, we produce a matrix showcasing the correlation amongst all stocks in the index, as depicted in Figure 1(c -i)

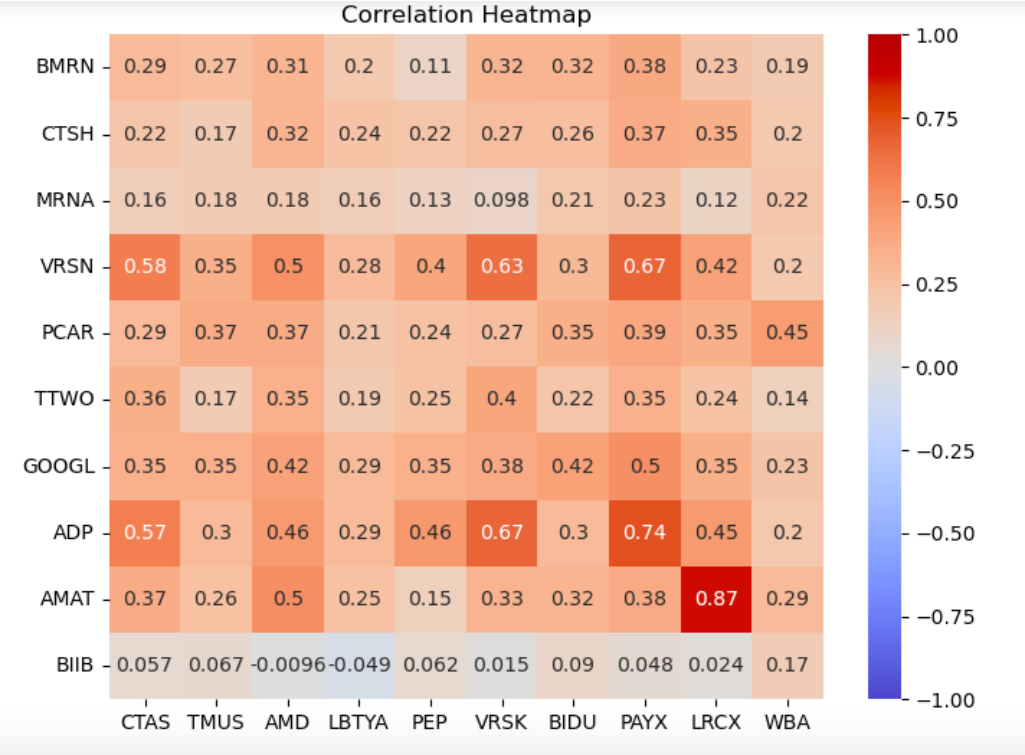


*Figure 1(d): A partial sampling of the correlation matrix for the year 2019*

For this optimization problem, we have the following decision variables, objective function and constraints:

**Decision variable** –

$y_j$: This is a binary decision variable which is -

= 1 if stock j from the index is selected for the fund.

= 0 otherwise

$x_{ij}$: This is another binary decision variable for each stock in the index i = 1, 2,…,n

= 1 if stock j in the index is the best representative for stock i.

= 0 otherwise

**Objective** -

Maximize the correlation that exists between our n-stock and their representatives in the Fund

$$\max_{x,y} \sum_{i=1}^{n} \sum_{i=1}^{n} \rho_{ij} x_{ij}$$

**Constraints** -

- The first constraint $\sum y_j = m$ indicates the number of stocks that must be present in $_{j=1}$ the m-stock fund.
- The second constraint $\sum x_{ij} = 1 \; for \; i = 1, 2, \ldots, n$ , where $j$=1 to n states that each stock i can only be "most similar" to one stock j.
- The third constraint $x_{ij} \leq y_j \; for \; i, j = 1, 2, \ldots, n$ indicates that the similarity between stock i and j can only occur if stock j is selected to be part of the m-stock fund.
- The last constraint denotes that $x_{ij}$ and $y_j$ can only be either 0 or 1. This constraint is not incorporated into the constraint matrix, instead, the variable type is set as 'Binary' in Gurobi.

**Stocks Optimization Code:**

```
# Number of funds
m = 5
# n = min(100, len(returns_matrix_19.columns))
n = len(returns_matrix_19.columns)

# Number of rows
p = len(returns_matrix_19)

# Model
stocks = gp.Model()
stocksy = stocks.addMVar(100, vtype='B')
stocksx = stocks.addMVar((100, 100), vtype='B')
# Constraint 1
stocks.addConstr(gp.quicksum(stocksy[i] for i in range(n)) == m)

# Constraint 2
stocks.addConstrs(gp.quicksum(stocksx[i, j]
                  for j in range(n)) == 1 for i in range(n))

# Constraint 3
stocks.addConstrs(stocksx[i, j] <= stocksy[j]
                  for j in range(n) for i in range(n))

# Objective function
stocks.setObjective(gp.quicksum(stocksx[i, j]*correlation_matrix_19.iloc[i, j] for j in range(n) for i in range(n)),
                    gp.GRB.MAXIMIZE)
```

*Figure 2(d): Python code for the above optimization problem*

## e. Calculating weights

Once we've selected stocks that maximize the likeness between the fund and the index, we must also define their respective proportions in the fund's makeup. Thus, **the goal of the weight allocation model is to assign a specific proportion to each chosen stock, ensuring**

**the difference between the returns of the fund and the index over a duration is as minimal as possible**. Essentially, if "q" symbolizes the return of the index on day "t", "$w_i$" indicates the proportion of a chosen stock "i", and "$r_{it}$" denotes the return of stock "i" on day "t", our objective is to minimize the subsequent equation.

**Objective**:

$$\min_{w} \sum_{t=1}^{T} \left| q_t - \sum_{i=1}^{m} w_i r_{it} \right|$$

Utilizing the absolute differences between the returns of the index and the fund ensures that negative and positive variations don't offset one another. However, this approach introduces complications since there isn't a straightforward method to incorporate absolute values within the objective function. To navigate this challenge, one can split "q" into two portions, introducing a new decision variable $y_i$ for each selected stock i.

$$\left| q_t - \sum_{i=1}^{m} w_i r_{it} \right|$$

1. First portion would look like this –

$$y_i \geq q_t - \sum_{i=1}^{m} w_i r_{it}$$

2. Second portion would look like this –

$$y_i \geq -\left(q_t - \sum_{i=1}^{m} w_i r_{it}\right)$$

**Decision variable** –

1. $w_i$ : weights of stock “i” selected for portfolio
2. $y_i$ : max value of the possible difference between index returns at t and weighted return of selected indexes

**Constraints** –

1. Sum of weights $w_i$ for ‘m’ stocks in the fund index equals 1

2. The difference between the index returns and the weighted return of the stocks should be less than $y_i$

**Weights Optimization Code:**

```
# model to calculate the portfolio weights
wt = gp.Model()

# variables
wtx = wt.addMVar(m)
wtz = wt.addMVar(p)

# Objective function
wt.setObjective(gp.quicksum(wtz[i] for i in range(p)), gp.GRB.MINIMIZE)

# Total portfolio weights = 1
wt.addConstr(gp.quicksum(wtx[i] for i in range(m)) == 1)

# Constrain 1
wt.addConstrs(wtz[i] >= q_2019[i] -
              gp.quicksum(wtx[j]*returns_matrix_19.iloc[i,
                          selected_stock_index[j]] for j in range(m))
              for i in range(p))
# Constrain 2
wt.addConstrs(wtz[i] >= gp.quicksum(wtx[j]*returns_matrix_19.iloc[i, selected_stock_index[j]] for j in range(m))
              - q_2019[i] for i in range(p))
```

*Figure 1(e): Python code for the above optimization problem*

## 2. Identifying 5 best stocks:

We ran the model for m=5. We found the following stocks of the index also shown in the image –

a) LBTYK: Liberty Global PLC (multinational telecommunications company)
b) MXIM: Maxim Integrated (analog and mixed-signal integrated circuits company)
c) MSFT: Microsoft (multinational technology corporation)
d) VRTX: Vertex Pharmaceuticals Incorporated (a biopharmaceutical company)
e) XEL: Xcel Energy Inc (utility holding company)

```
['LBTYK', 'MXIM', 'MSFT', 'VRTX', 'XEL']
```

*Figure 2.1: 5 best stocks given by our stock optimization model*

From the optimization outlined in section e), we can also calculate the optimal weights for these 5 stocks in a tracking portfolio. Microsoft has the highest weight followed by Maxim Integrated and so on:

```
array([0.04886175, 0.21038806, 0.58035198, 0.07119022, 0.089208  ])
```

*Figure 2.2: Weights obtained for all the 5 stocks picked*

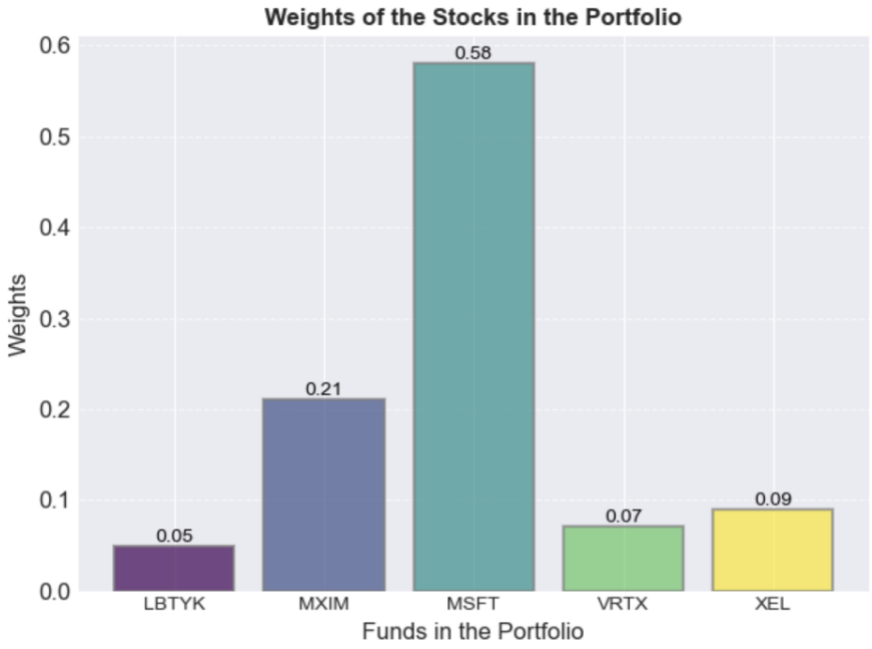


*Figure 2.3: Graphical comparison of the weights for the best 5 stocks picked*

From Figure 2.4, we can evaluate the returns of our selected five stocks against the NASDAQ index for 2020.

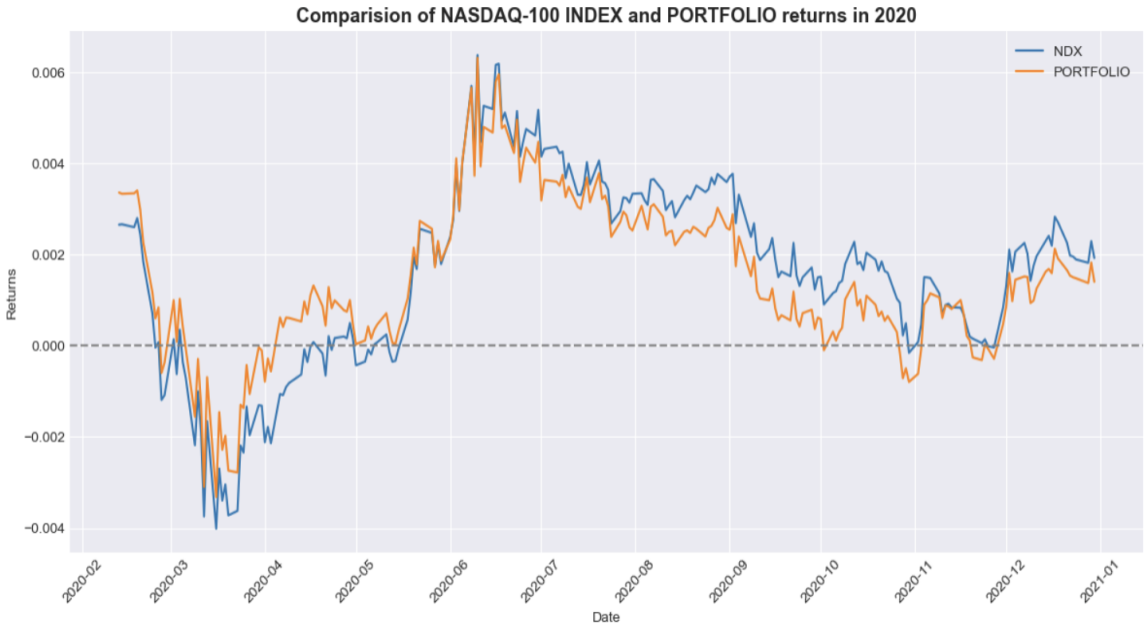


*Figure 2.4: Daily Returns of 5 Stock Portfolio vs NASDAQ Index for 2020*

1. **Closely Tracked Movements**: For a significant portion of the year, the Portfolio's returns mirror the NASDAQ-100 Index. This indicates that the portfolio has a strong correlation with the market and effectively captures broader market movements.

2. **Initial Variance and Recovery**: In the early months of 2020, there's a notable variance between the Portfolio and NDX, with the Portfolio experiencing a sharper drop in returns. However, it swiftly recovers and starts to align closely with the NDX by mid-year.

3. **Outperformance and Underperformance Instances**: Throughout the year, there are select periods where the Portfolio outperforms the NDX, especially noticeable around July and late October. Conversely, there are also moments, such as late February and late November, where the Portfolio slightly underperforms compared to the NDX.

4. **Stability in Returns**: Both the Portfolio and the NDX show considerable stability from August onwards, with fewer sharp fluctuations. This could be indicative of a more stable market environment during this period or effective risk management strategies deployed for the Portfolio.

5. **End-of-Year Divergence**: Towards the end of 2020, there's a slight divergence with the Portfolio experiencing a minor dip in returns compared to the NDX. This will be a point of interest for investors and could be attributed to specific stock selections or market events impacting the portfolio holdings more than the broader index.

## 3. Identifying the optimal no. of stocks from a range of 'm' values

To determine the ideal stock count, multiple models were executed with varying numbers of stocks for the new fund. Iterating over a range of values for m (the chosen stock count) allowed us to gather outcomes from numerous models. Figure 3.1 depicts this table comparison of error for year 2019 with 2020

| | total_correlation | selected_stocks_index | selected_stocks | selected_stocks_weights | ObjVal_19 | ObjVal_20 |
|---|---|---|---|---|---|---|
| **5** | 54.841179 | [56, 59, 63, 94, 98] | [LBTYK, MXIM, MSFT, VRTX, XEL] | [0.04886174835252394, 0.21038806005665583, 0.5... | 0.789178 | 1.112437 |
| **10** | 59.333081 | [0, 4, 40, 53, 56, 59, 63, 79, 94, 98] | [ATVI, ALGN, EXPE, KHC, LBTYK, MXIM, MSFT, ROS... | [0.044200081625158275, 0.024995874294039707, 0... | 0.701218 | 1.102404 |
| **20** | 66.648971 | [0, 4, 5, 10, 15, 17, 24, 36, 40, 51, 53, 56, ... | [ATVI, ALGN, GOOGL, ANSS, ADP, BIIB, CHTR, DLT... | [0.017568251619136558, 0.015084885097906304, 0... | 0.466268 | 0.855446 |
| **30** | 72.697515 | [0, 1, 5, 12, 15, 17, 19, 28, 30, 34, 35, 36, ... | [ATVI, ADBE, GOOGL, AMAT, ADP, BIIB, BKNG, CTX... | [0.020985623199727105, 0.04838004819167127, 0.... | 0.409792 | 0.767367 |
| **40** | 78.259750 | [0, 1, 4, 5, 8, 12, 15, 17, 19, 23, 24, 25, 29... | [ATVI, ADBE, ALGN, GOOGL, AMGN, AMAT, ADP, BII... | [0.025818176010387134, 0.03621862263898249, 0.... | 0.363281 | 0.767891 |
| **50** | 83.316676 | [0, 1, 3, 4, 5, 8, 12, 15, 17, 22, 23, 24, 25,... | [ATVI, ADBE, ALXN, ALGN, GOOGL, AMGN, AMAT, AD... | [0.01945631641398465, 0.034341414889951856, 0.... | 0.332540 | 0.772100 |
| **60** | 87.877830 | [1, 2, 3, 4, 5, 8, 12, 15, 16, 17, 18, 19, 22,... | [ADBE, AMD, ALXN, ALGN, GOOGL, AMGN, AMAT, ADP... | [0.05055168498815333, 0.016880429624208538, 0.... | 0.352056 | 1.164932 |
| **70** | 92.062665 | [2, 3, 4, 5, 8, 12, 15, 16, 17, 18, 19, 21, 22... | [AMD, ALXN, ALGN, GOOGL, AMGN, AMAT, ADP, BIDU... | [0.005603083016214723, 0.0, 0.0087478992817680... | 0.233143 | 0.861893 |
| **80** | 95.729016 | [0, 2, 3, 4, 5, 8, 11, 12, 14, 15, 16, 17, 18,... | [ATVI, AMD, ALXN, ALGN, GOOGL, AMGN, AAPL, AMA... | [0.00861135341944156, 0.0064529408149833515, 0... | 0.147683 | 0.537323 |
| **90** | 98.517181 | [0, 1, 2, 3, 4, 6, 7, 8, 10, 11, 12, 13, 14, 1... | [ATVI, ADBE, AMD, ALXN, ALGN, GOOG, AMZN, AMGN... | [0.0040721125909736165, 0.020500079760527982, ... | 0.053827 | 0.370506 |
| **100** | 100.000000 | [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,... | [ATVI, ADBE, AMD, ALXN, ALGN, GOOGL, GOOG, AMZ... | [0.004222720908251523, 0.017137939539245004, 0... | 0.044911 | 0.368671 |

*Figure 3.1: Comparing error of portfolio for various values of m in 2019 and 2020*

Now graphically speaking, let's take a look at figure 3.2 and understand a few key insights

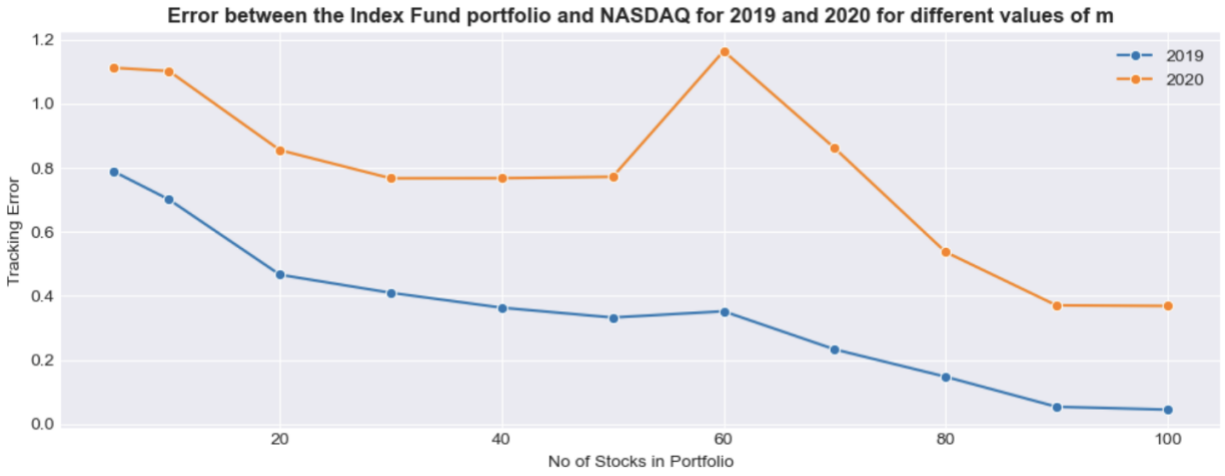


*Figure 3.2: Comparing errors across 2019 and 2020*

- **Shift in Tracking Error Dynamics (2020 vs. 2019):** In 2020, there's a marked peak in tracking error with portfolios around 40 stocks, indicating increased volatility or discrepancies in this range. In contrast, 2019 maintains a consistent, downward trajectory in tracking error as portfolio size increases, reflecting a more predictable behavior.

- **Optimal Stock Count for 2020**: Post the 40-stock peak in 2020, there's a pronounced reduction in error as we approach 60 stocks. However, for minimizing tracking error in 2020, portfolios hovering between 80 to 100 stocks appear to be the most stable.
- **Year-on-Year Comparison at Key Stock Numbers**:
  - At 40 stocks, 2020 exhibits a considerably higher error than its 2019 counterpart, signifying the former's heightened sensitivity or potential external influences impacting stock performance.
  - At the 60-stock point, while both years see a decline, 2020 remains elevated in its tracking error compared to 2019, suggesting underlying factors that may have uniquely influenced the 2020 market.
  - By the time we reach the 80-100 stock range, the two years converge closer in terms of tracking error, indicating a level of parity in portfolio behavior at these higher stock counts.
- **Strategic Considerations**: For stakeholders or investors focused on stability, a portfolio within the 80-100 stock range for the year 2020 would be advisable. However, for those looking at historical performance and predictability, expanding the portfolio consistently, as seen in 2019, might be better.

## 4. MIP Model

In our refined methodology, akin to the primary approach, we are trying to navigate beyond stock selection by evaluating portfolios with increments ranging from 5 to 100 stocks, all the while utilizing the same absolute difference criterion. Essentially, a diminished metric value indicates a portfolio's heightened alignment with the index.

The graphical representation (Figure 4.1) elucidates an anticipated trend: as the portfolio's stock count escalates, its performance sees enhancement. Notably, this revamped strategy outpaces our initial method, evident from the reduced test error. Furthermore, the test error's trajectory closely mirrors that of the training error, underscoring our portfolio's adeptness in echoing the returns.

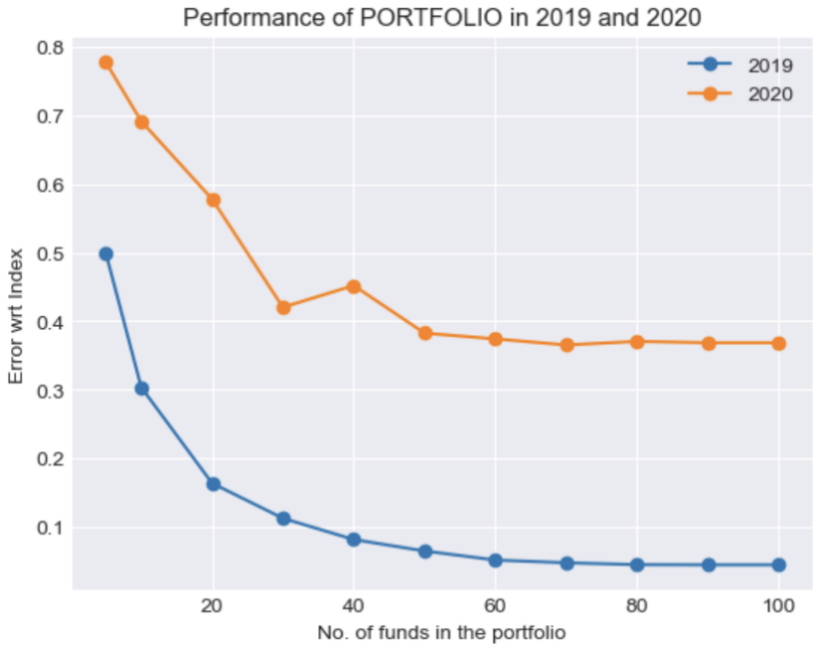


*Figure 4.1: Comparison of errors for 2019 and 2020*

Based on figure 4.2, for both 2019 and 2020, the results are consistent. After incorporating 40 stocks, the gains start to taper off. Yet, the performance plateaus, suggesting that a portfolio with 40 stocks mirrors the performance of one with 100 stocks quite closely.

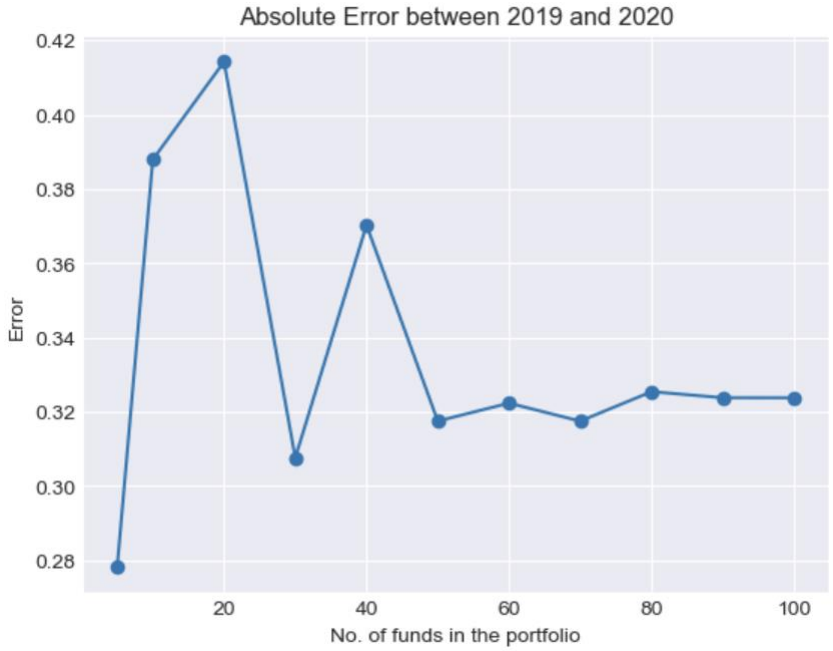


*Figure 4.2: Absolute difference in error between 2019 and 2020*

Now before we deep dive into making recommendation on how many stocks to choose from, followed by calculating their weights, we must compare the difference between the two models. We have plotted both models and their respective errors in figure 4.3

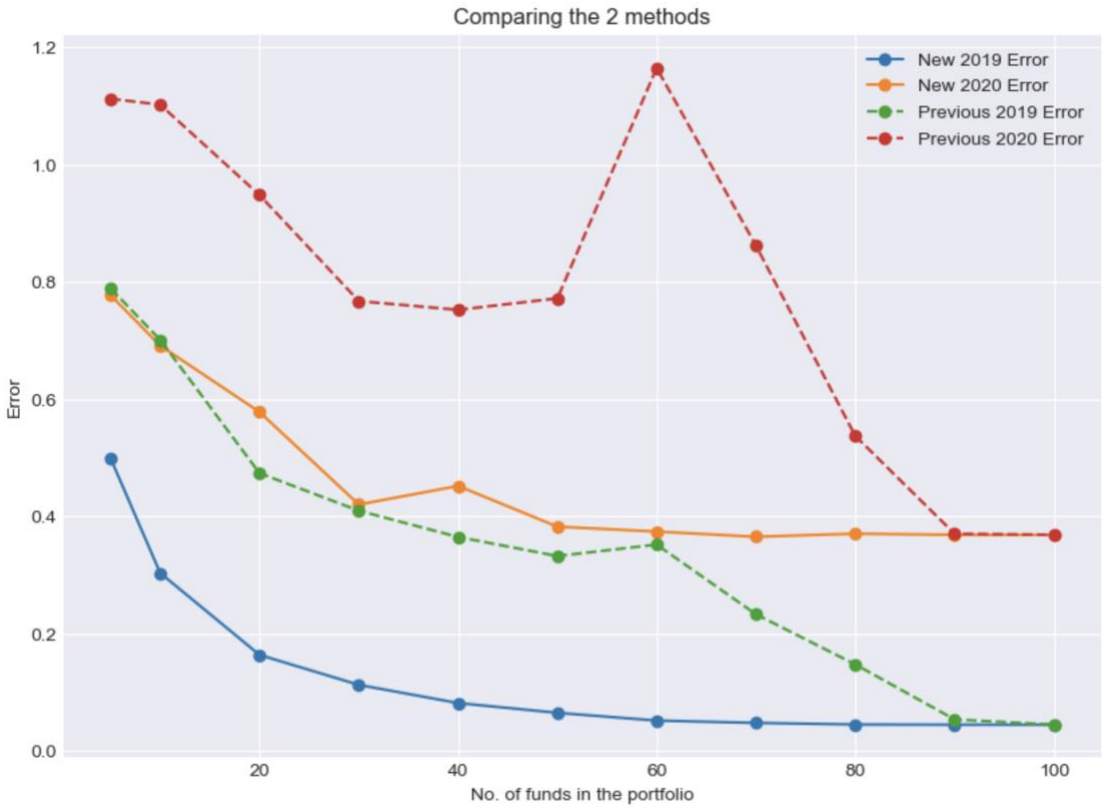


*Figure 4.3: Comparison of errors for 2019 and 2020*

Based on this, we can notice that Method 2 (i.e., MIP Model) consistently outperforms Method 1 across most values of 'm', achieving optimal results around m=50, or within the 40-50 range, where we see an elbow in the graph. Thereafter, its performance levels out. Conversely, Method 1 displays more variability and shows a gradual enhancement in performance as 'm' escalates.

**Mixed Integer Approach Code:**

```
for k, m in enumerate(a):
    # Model
    Mixed = gp.Model()
    Mixedy = Mixed.addMVar(n, vtype='B')
    Mixedw = Mixed.addMVar(n)
    Mixedz = Mixed.addMVar(p)

    Mixed.setObjective(gp.quicksum(Mixedz[i]
                       for i in range(p)), gp.GRB.MINIMIZE)

    # Adding Constraints
    Mixed.addConstr(gp.quicksum(Mixedw[i] for i in range(n)) == 1)
    Mixed.addConstr(gp.quicksum(Mixedy[i] for i in range(n)) == m)
    Mixed.addConstrs(Mixedw[i] <= M*Mixedy[i] for i in range(n))

    # Constraints
    Mixed.addConstrs(Mixedz[i] >= q_2019[i] -
                     gp.quicksum(
                         Mixedw[j]*returns_matrix_19.iloc[i, j] for j in range(n))
                     for i in range(p))
    Mixed.addConstrs(Mixedz[i] >= gp.quicksum(Mixedw[j]*returns_matrix_19.iloc[i, j] for j in range(n))
                     - q_2019[i] for i in range(p))
```

*Figure 4.4: Python code for the MIP Model*

## 5. Conclusion

Growth of an investment in an index is given by the cumulative product of daily returns for that year. Tracking the cumulative return provides a way of comparing an investment in the curated, small stock portfolio vs investment in the Nasdaq 100. Below are plots of the cumulative return of both investments (the 5-stock portfolio and the QQQ-100) in 2019 and 2020.


*Figure 5.1: Cumulative Returns of 5 Stock Portfolio vs NAS in 2019*

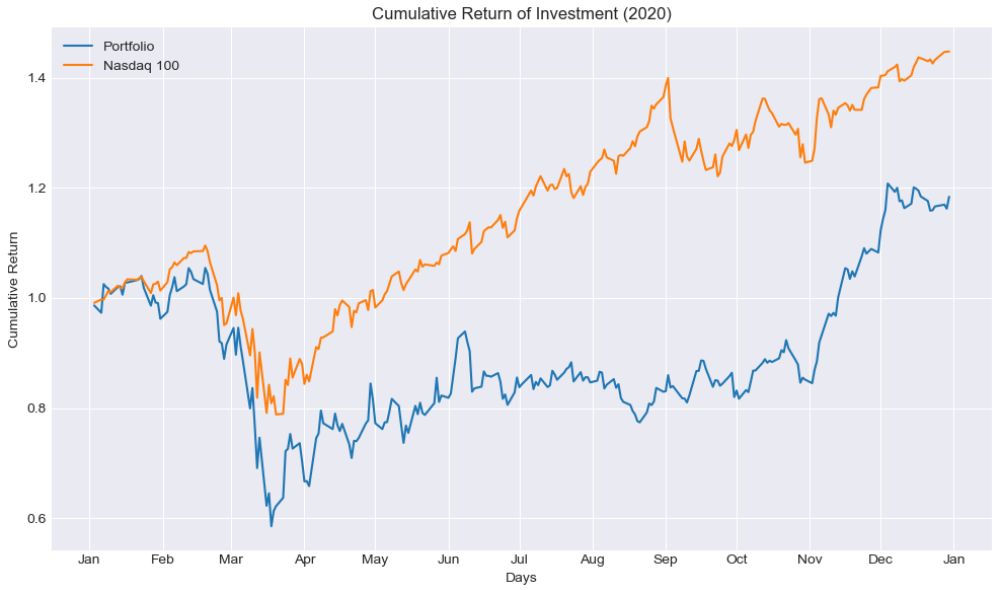


*Figure 5.2: Cumulative Returns of 5 Stock Portfolio vs NAS in 2020*

The relative heights of the graphs suggest that the optimal 5-stock portfolio outperformed the market in 2019 and underperformed in 2020. The movements of the graphs also suggest that the highly compressed 5-stock portfolio is still effective for tracking the movements of the larger index. As is exhibited by the 2019 plot, tracking errors can compound and lead to a gradual drift in the returns of the portfolio and the index.

## 6. Final Recommendation

Based on our comprehensive analysis, we've discerned the effectiveness of index compression in achieving the dual objectives of reducing the size of the index while preserving the essential benefits of diversification. For this problem, reductions in tracking error plateaued at around 40-50 stocks, but this range may differ for other problems. However, it's paramount to recognize the existence of certain drawbacks in this strategy.

To effectively address these concerns, we suggest the following:

**Selective Stock Inclusion**: Instead of integrating every available option, a discerning approach targeting enhanced portfolio diversification is recommended. This is best achieved through the adoption of a 50-stock with weights calculated through mixed integer programming model. It provides a balance by minimizing tracking error and allowing for a streamlined, yet comprehensive portfolio in comparison to the original QQQ index.

**Advantages of Mixed-Integer Optimization**: Going beyond the confines of traditional integer programming, the mixed-integer optimization technique introduces an element of flexibility and adaptability into portfolio construction. Unlike its counterpart, which restricts stock selection, mixed-integer optimization enhances portfolio allocation by constraining the number of non-zero weights to be integers. This ensures a detailed representation of the complex interplay between stock selection and overarching portfolio goals.

**Balancing Act**: Through mixed-integer optimization, an ideal balance is struck between transaction costs and tracking error. This method's inherent flexibility is instrumental in effectively handling diverse constraints and objectives, leading to a portfolio that's more aligned with the modern and ever-evolving financial market landscape.

**Opportunity for Refinement**: By allocating more time to the optimization process, there's a potential to yield even more refined results, thereby enhancing overall portfolio performance. But there is tradeoff with the accuracy and time constraint to reach the final results

**Going forward** our methodology for optimizing a tracking portfolio can be repeated for other indices and adjusted towards different goals. Anyone performing this optimization simply needs to decide how they will balance the transaction costs of having a larger index against the cost of higher tracking error for a small index.