Portfolio Optimization

# Introduction

The datasets provided contained stock price of US equities from 1993-2019, however, I only looked at the visuals of portfolios in periods of 10 years. This was done using Mathematical Optimization Algorithm, Monte Carlo Simulation as well as the Modern Portfolio Theory. The visuals were created using Riskfolio-Lib. This report will look at the various methods used in exploratory data analysis and data wrangling and visualization, in order to assess the portfolios that maximize returns for the risk assumed.

# Exploratory Data Analysis

The analysis was done in a systematic order where I dropped the empty rows and the rows outside the time being analyzed, and copied it to stock\_short as these columns would not be in S&P 500. Then we Store the columns names, and calculate the return as (FinalPrice-StartPrice)/StartPrice, and the volatility as StandardDeviation/StartPrice. The Sharpe Ratio should have some no risk return to compare this against as it gives an indication of the performance of the stock over the period of interest.

After doing this, I plotted the volatility against the return for visual purposes along with getting a reference line to give and indication of he best performing stocks. It is worth noting that the best performing stocks are in contact with line through the origin with the largest slope.

# Data Wrangling

Following the completion of exploratory data analysis, I decided to compute the returns and statistics necessary to run both the Monte Carlo Simulation and the Mathematical Optimization Algorithm. However, due to the size of the dataset, we created a smaller dataset named data1 with the 5 best performing stocks. I conducted data wrangling on both data frames concurrently as the results helped me determine which algorithm is suitable for the job.

This was done by getting the daily returns for each stock using the pct\_change() method. This was my preferred method as they have the material of being asset-additive which in turn is needed to compute portfolio returns. Subsequently, the daily returns were annualized using .mean() method whilst assuming the number of trading days are 252.

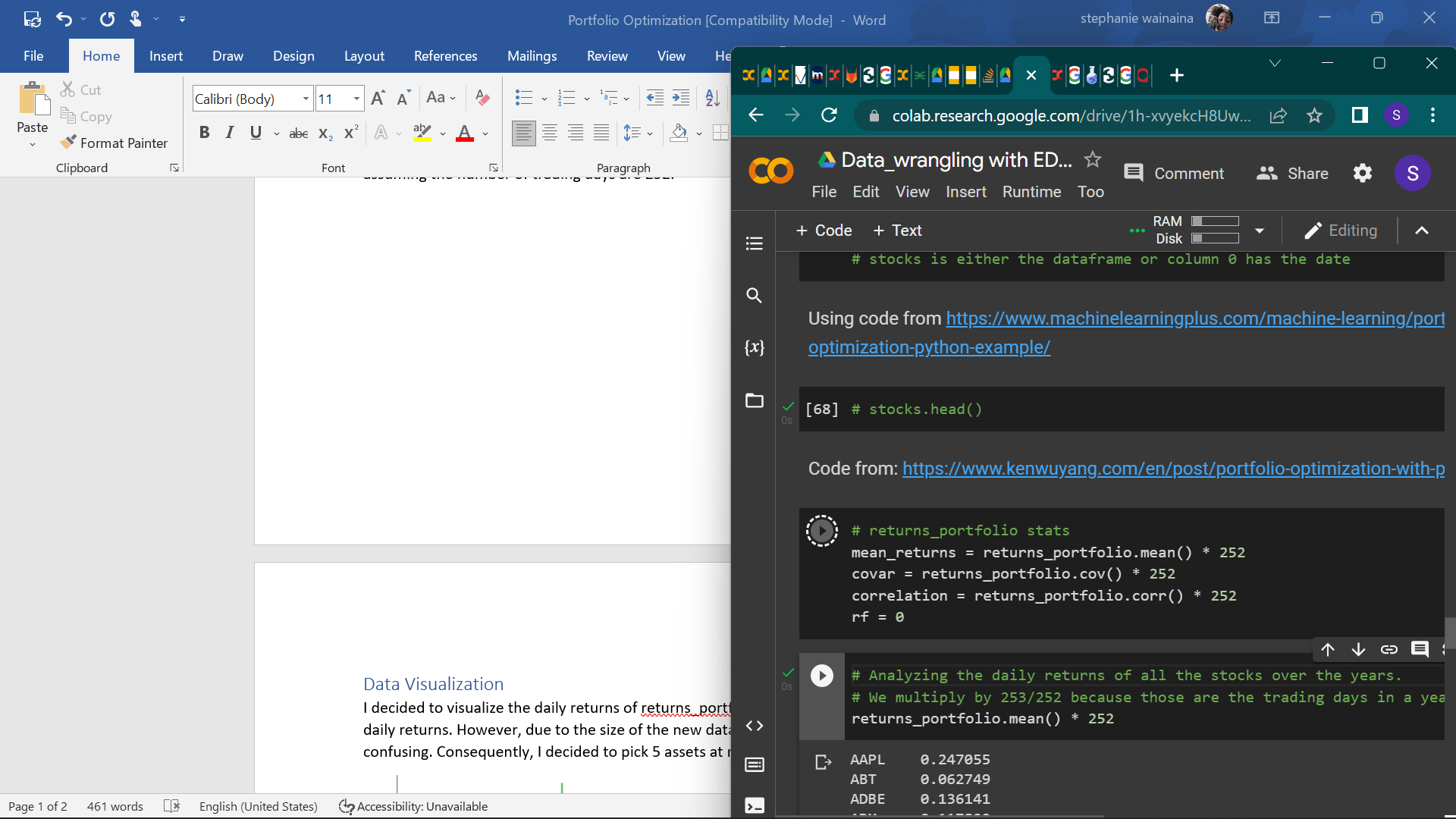


Figure 1: stats for retruns\_ portfolio

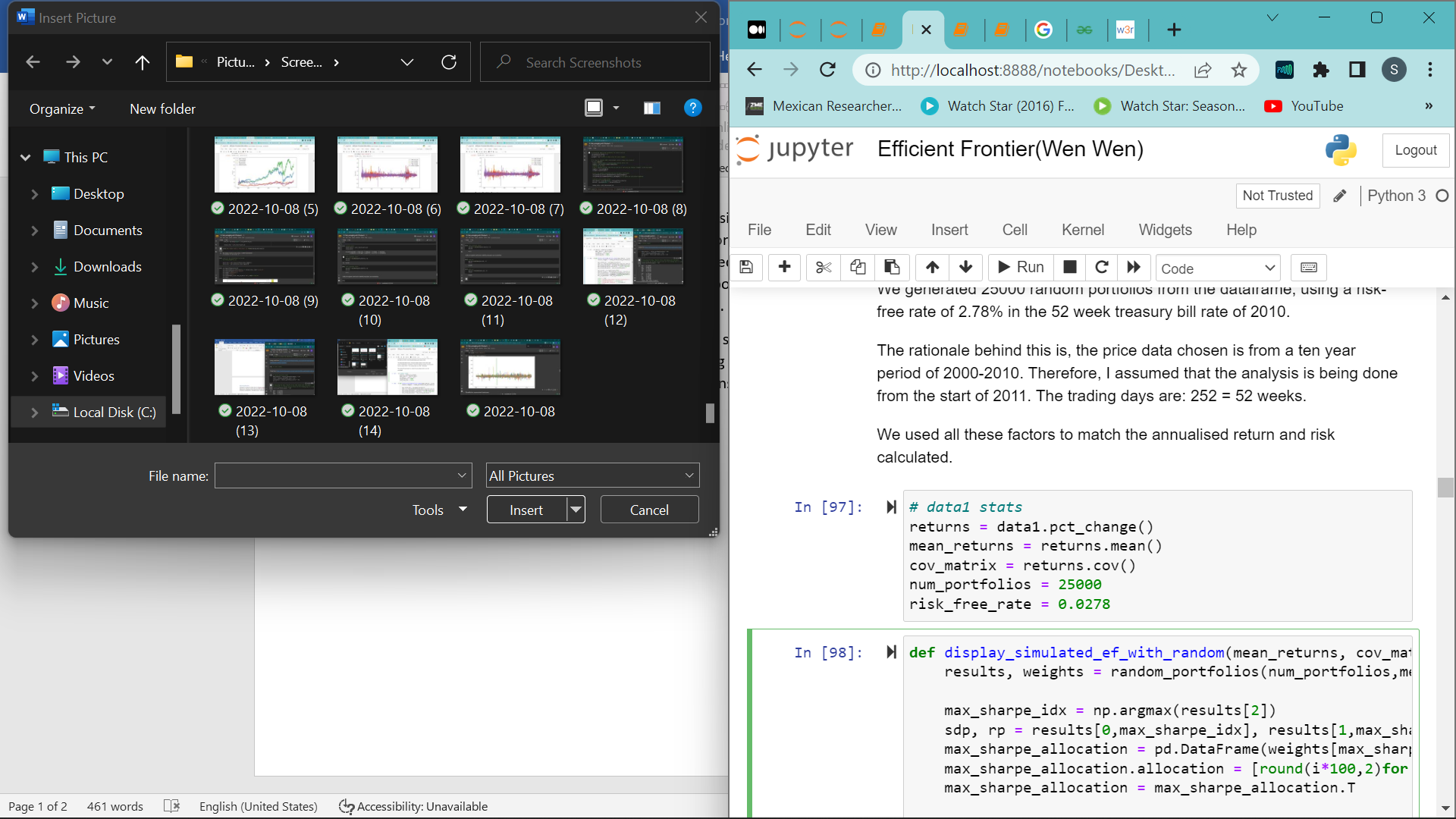


Figure 2: stats for data1

The risk factor in returns\_portfolio is set to 0 because it will use Monte Carlo simulation whereas data1 will apply the Mathematical Optimization Algorithm. Thus, using a risk-free rate of 2.78% in the 52-week treasury bill rate of 2011.

## Random Portfolio Generation.

Generating random portfolios was quite challenging as I had to find a method that spreads the capital provided between the assets in a manner that ensures maximum return of the portfolio. It is worth noting, there is no specific solution for this problem however, there are a cluster of solutions known as the efficient frontier. I tackled this task using 3 models:

* The Modern Portfolio Theory
* The Monte Carlo Simulation
* The Mathematical Optimization Algorithm

### The Modern Portfolio Theory

The Markowitz model for the solution of the portfolio optimization problem has a twin objective of maximizing return and minimizing risk, built on the Mean-Variance framework of asset returns and holding the basic constraints.

I applied the Modern Portfolio Theory in order to get the efficient frontier. In favor of plotting the graph of efficient frontier using Markowitz Portfolio Optimization, I ran a loop. In each iteration, the loop considers different weights for assets and calculates the return and volatility of that particular portfolio combination. This loop runs n times.

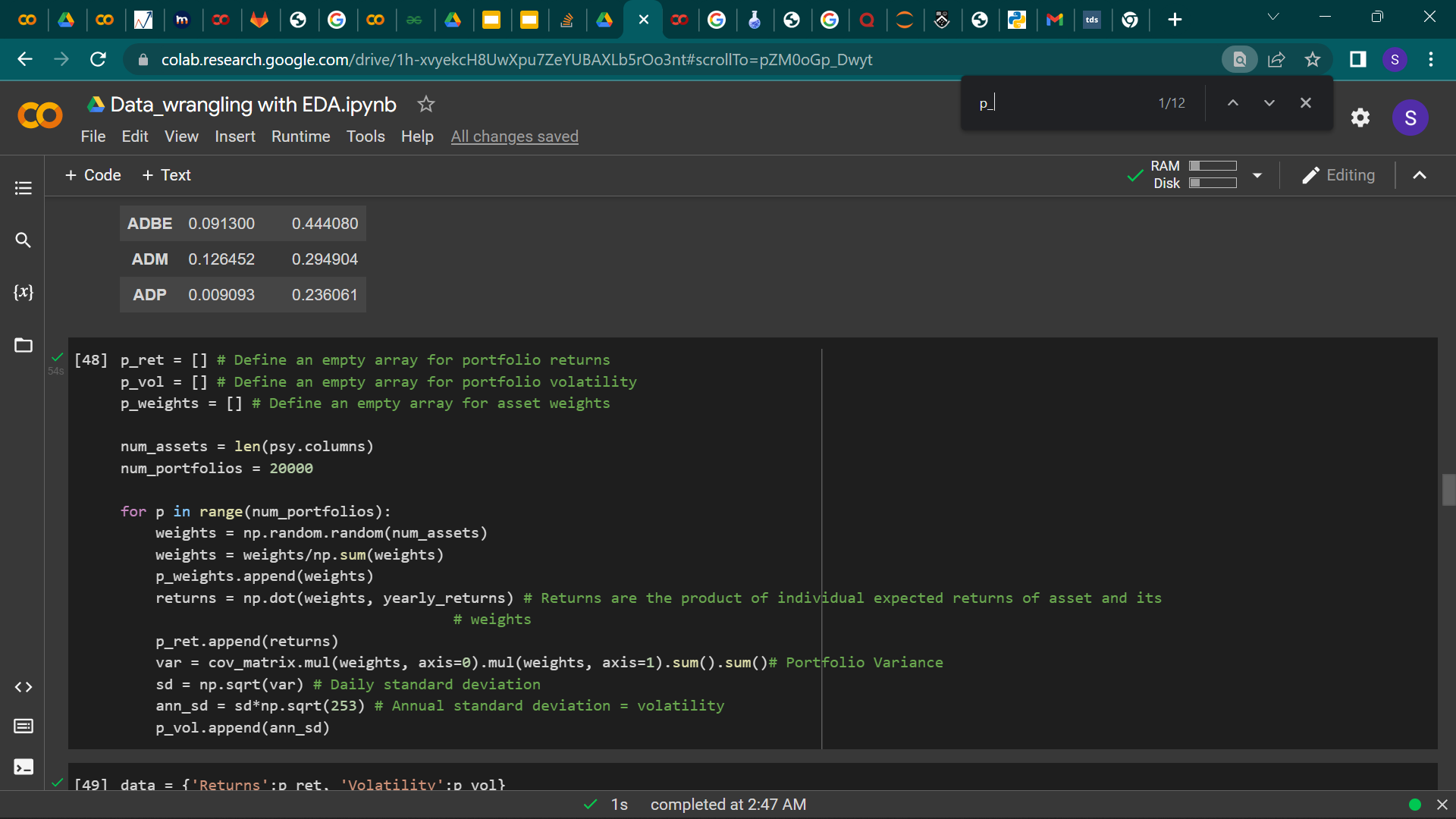


Figure 3: Markowitz optimization code

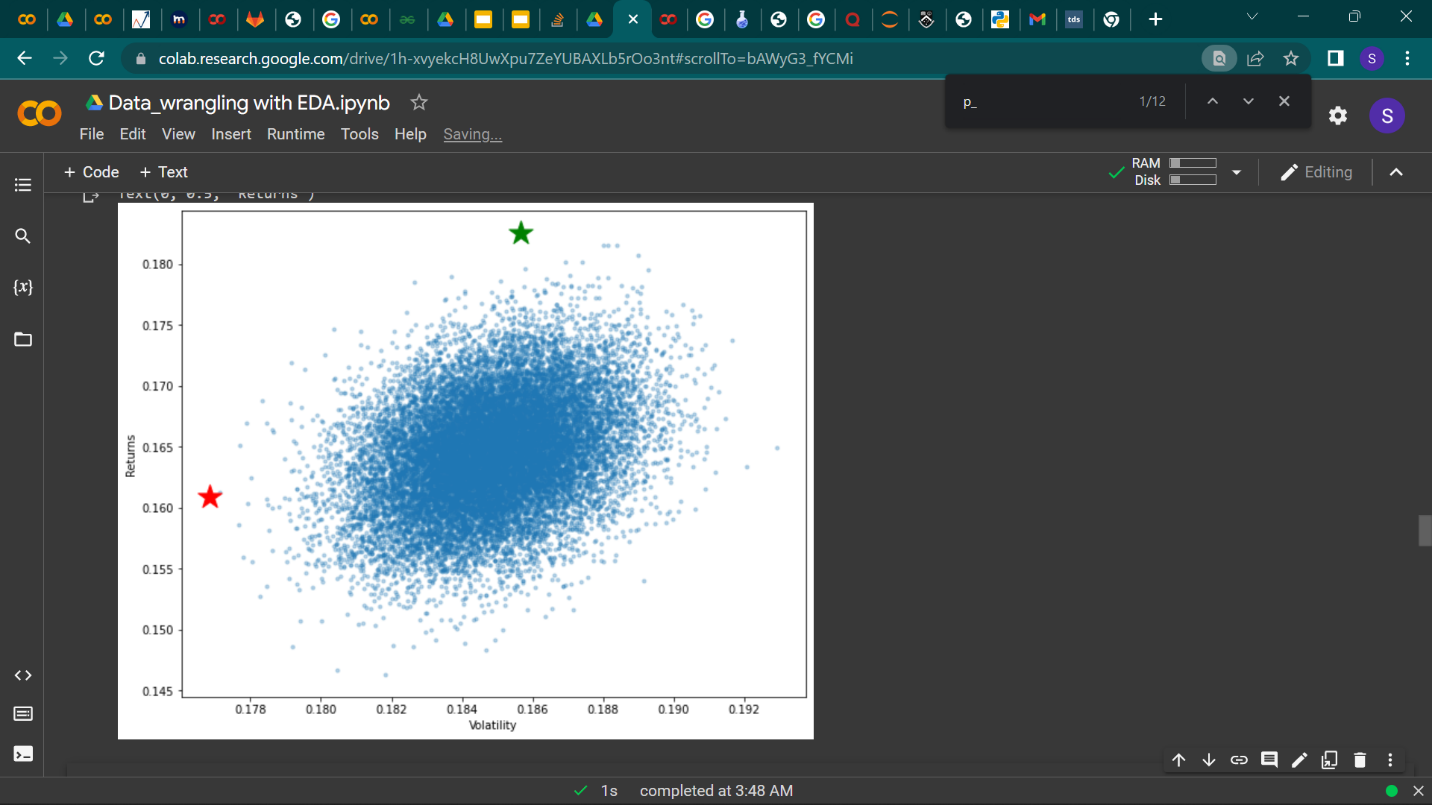


Figure 4: Effective frontier with minimum volatility and maximum sharpe ratio

The red star denotes the minimum volatility while the green star denotes the optimal risk portfolio.

### The Monte Carlo Simulation

While applying this simulation, the first thing I did was mimic a random set of portfolios to visualize the risk-return profiles of our given set of assets. Consequently, I defined 2 functions that take inputs of asset weights and output the expected portfolio return and, standard deviation.

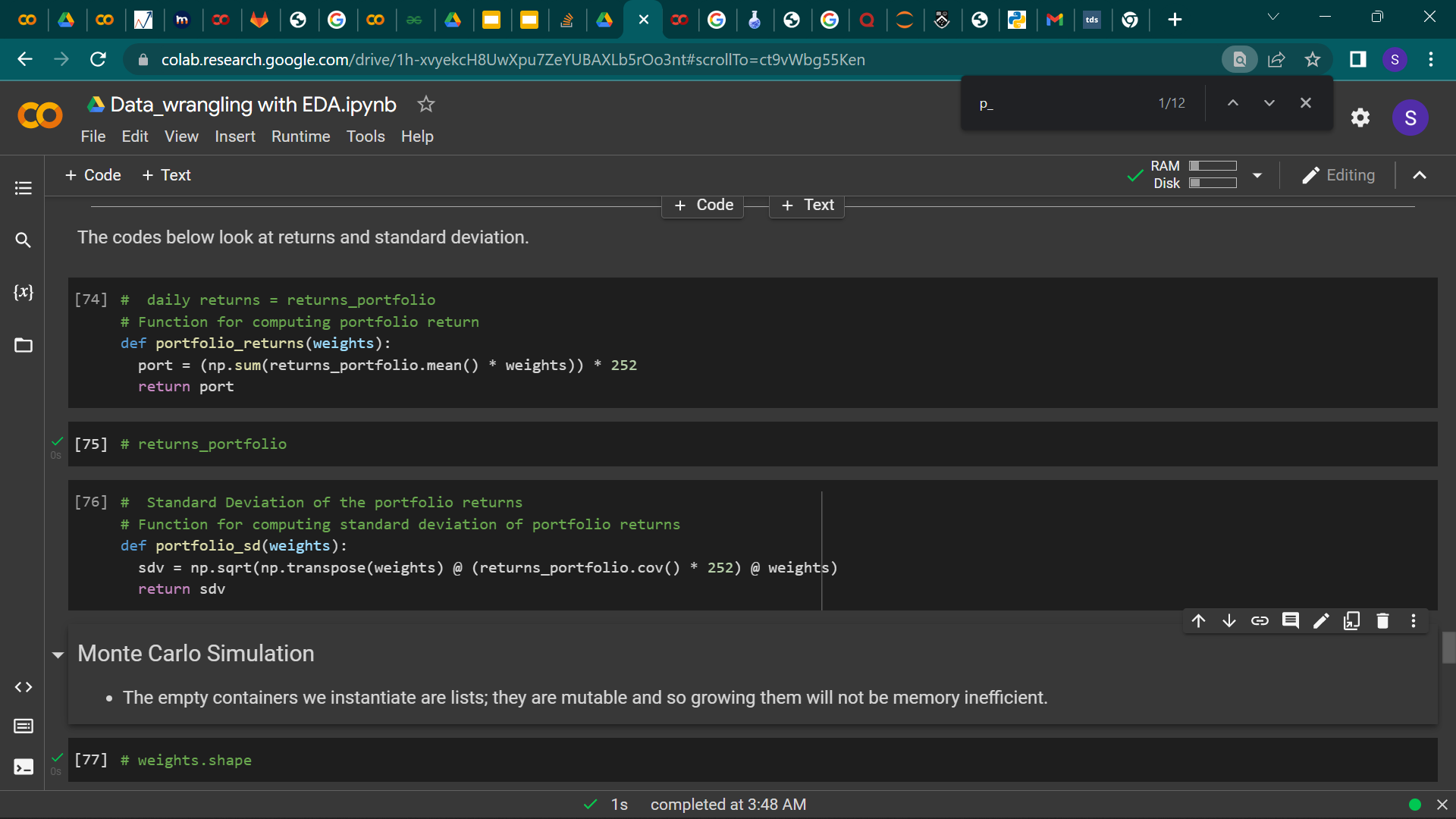


Figure 5: expected return function and standard deviation

A for loop was then created to simulate the random vectors of asset weights by processing the expected portfolio and standard deviation for each combination of weights. This action occurs randomly. It Is worth noting that the Monte Carlo simulation is computationally intensive due to the size of the dataset hence, making us settle for the mathematical optimization.

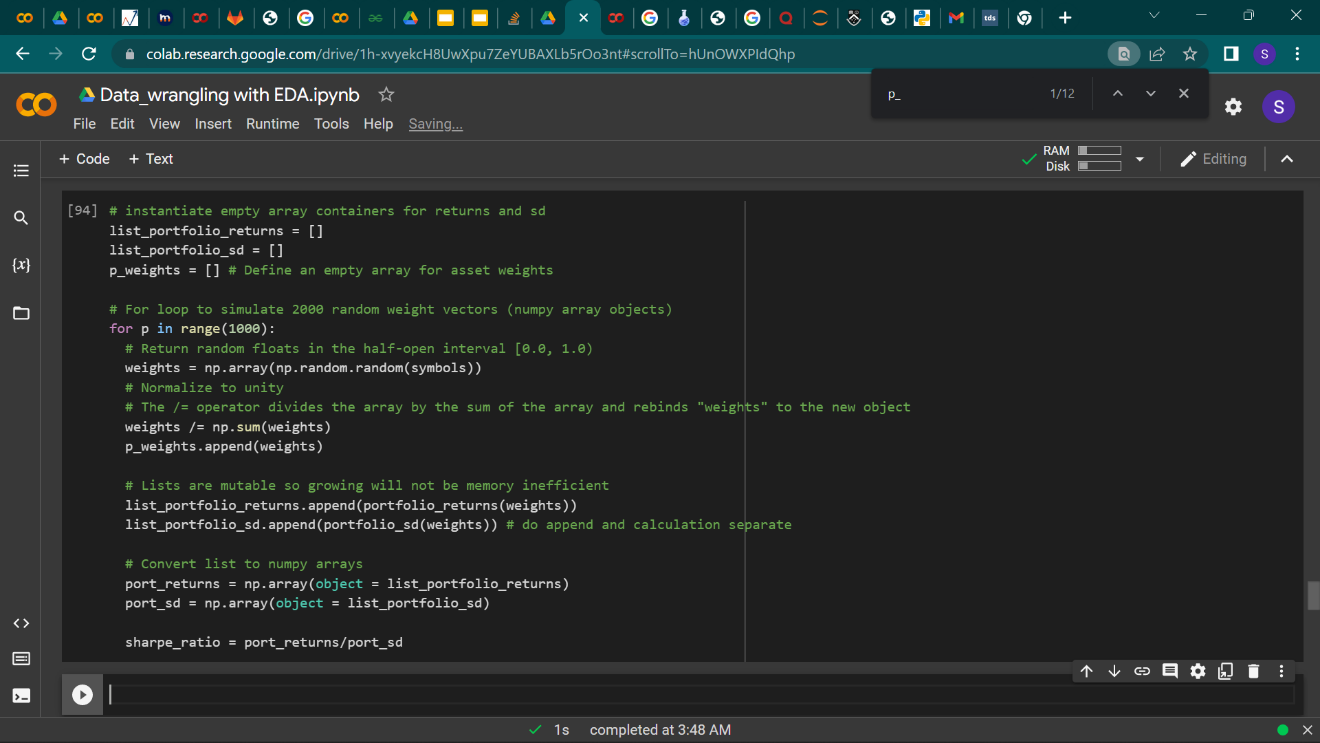


Figure 6: Monte Carlo Simulation code

### The Mathematical Optimization Algorithm

This algorithm uses the optimization functions such as SciPy library to find the optimal weights mathematically. SciPy’s in-built optimization algorithm creates an optimizer that attempts to minimize the negative Sharpe ratio. In turn, this helps calculate the weight allocation for the portfolio. This is seen in the function below.

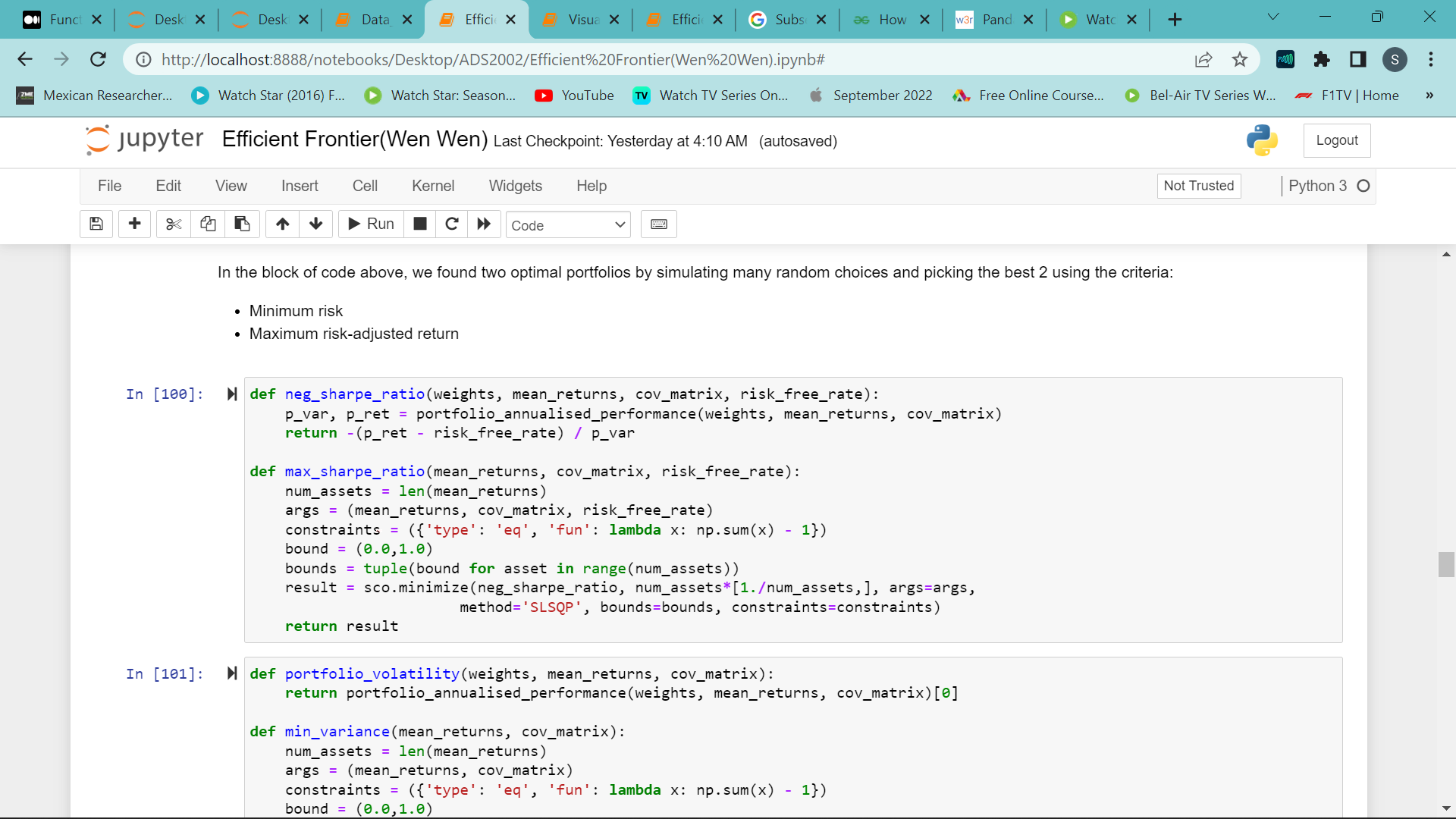


Figure 7: Maximum Sharpe Ratio Portfolio

Since there is no maximize in the SciPy library, the function passes something that is minimized hence the neg\_sharp\_ratio. SciPy’s minimize function can be used to generate portfolios that will also plot the efficient frontier as seen below.

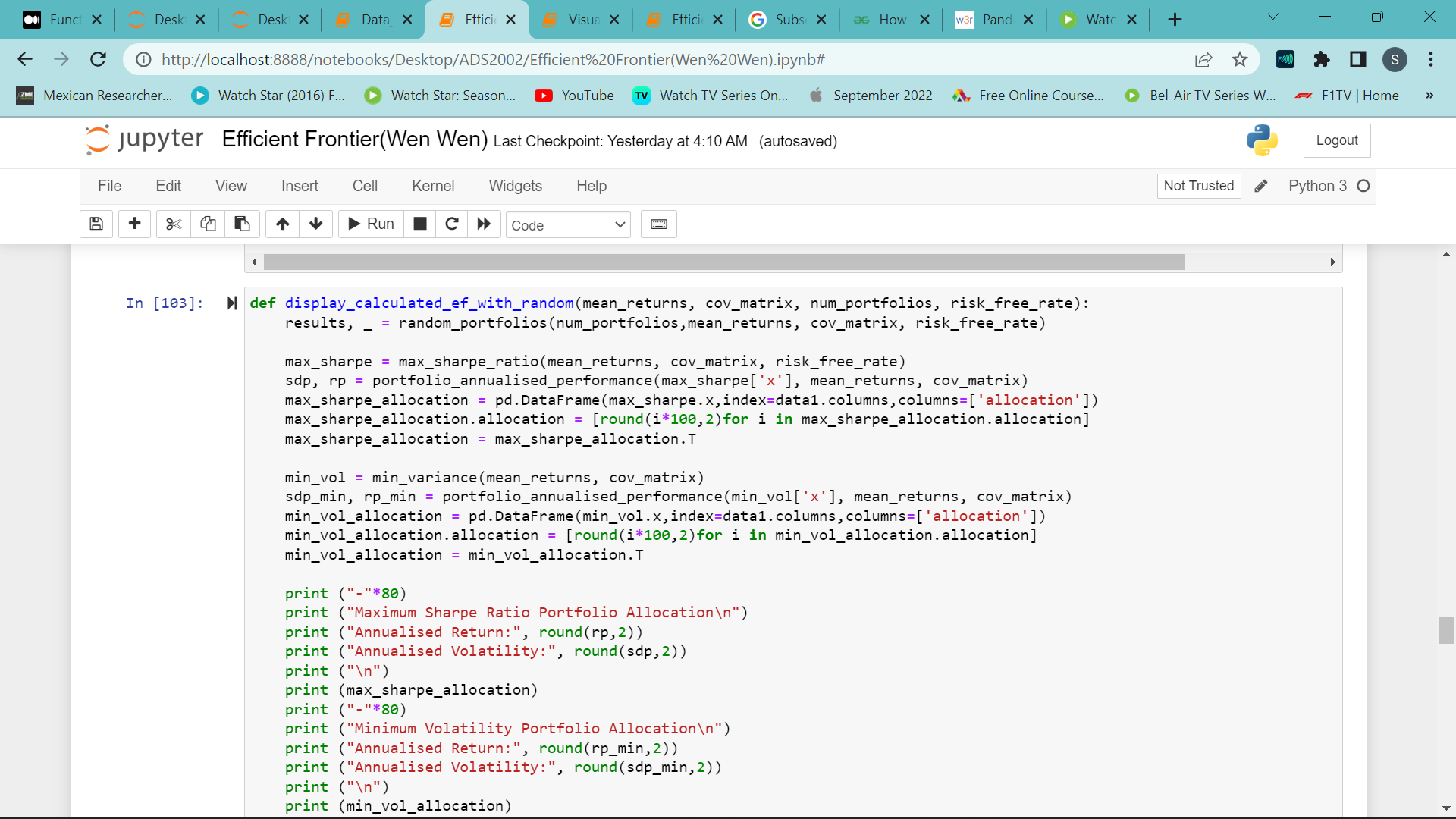


Figure 8: SciPy's minimize function in use

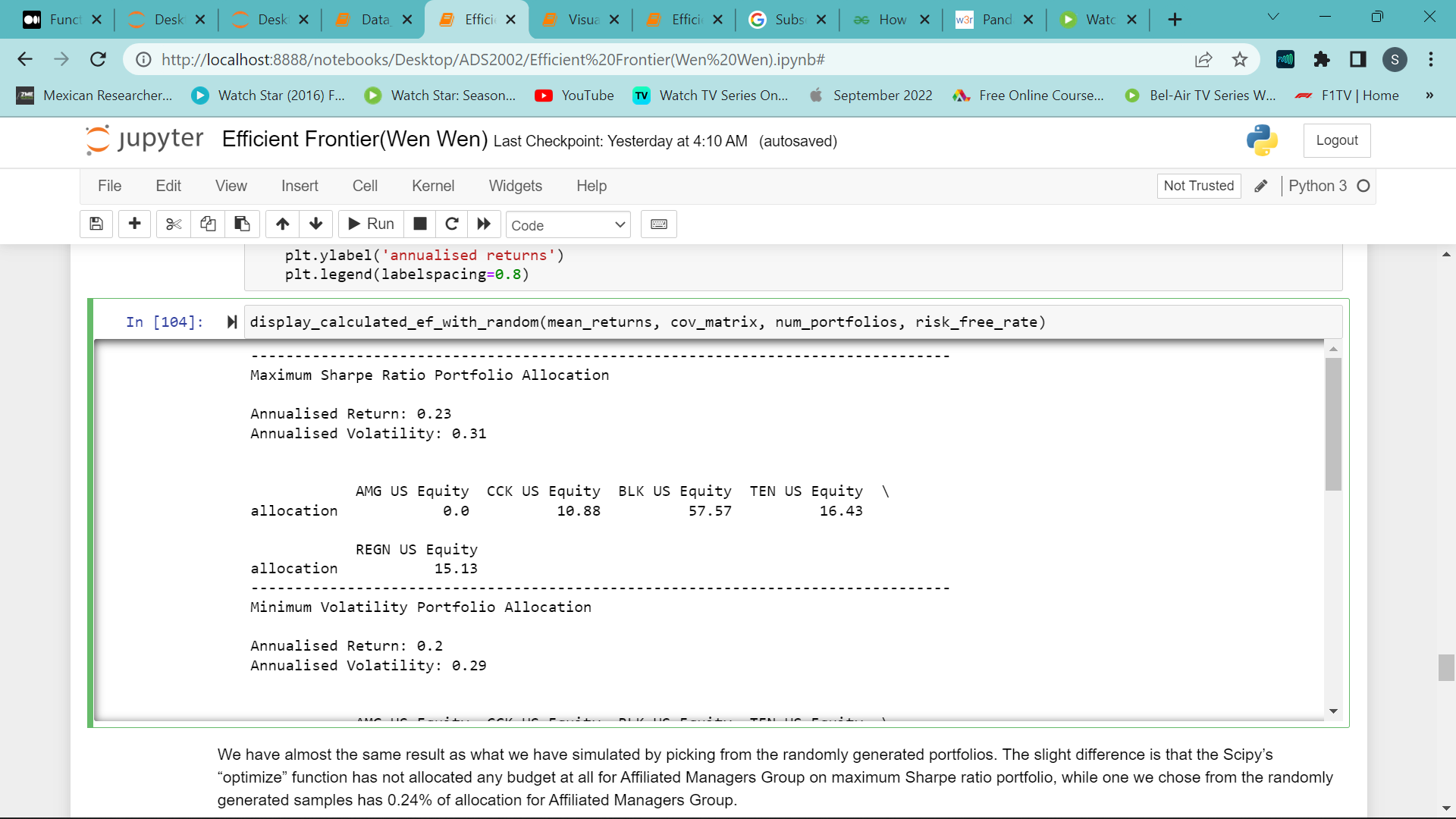
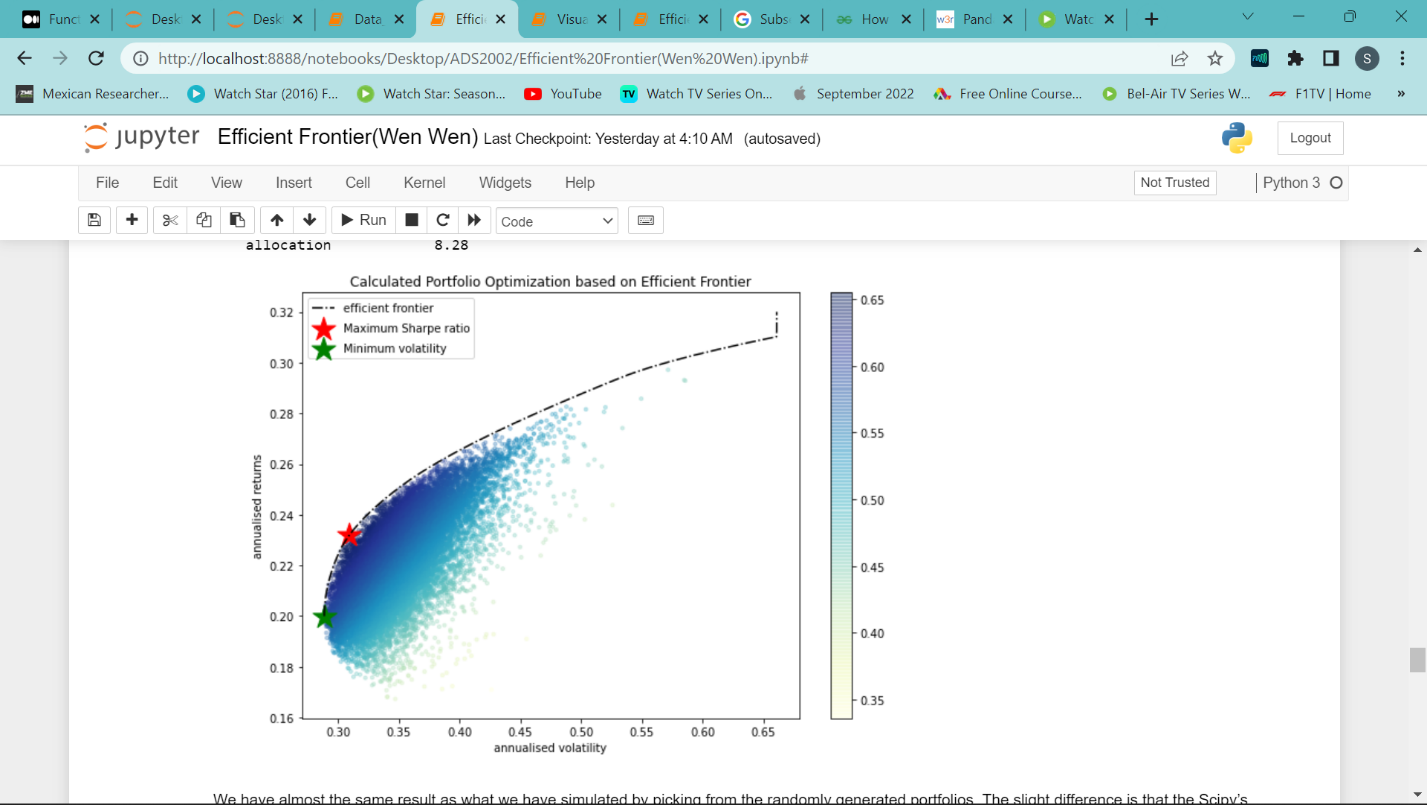


Figure 9: Efficient Frontier using SciPy's Figure 10: Portfolio Allocation

We have achieved relatively similar results with the randomly generated portfolios; however, this is simulated. The slight difference is that the Scipy’s “optimize” function has not allocated any budget for Affiliated Managers Group (AMG) on maximum Sharpe ratio portfolio, while, one we chose from the randomly generated samples has 0.24% of allocation for Affiliated Managers Group.

## Data Visualization of the efficient frontiers.

Extensive data visualization was only done on the smaller dataset as the original dataset was too big so, it would have resulted in overloaded plots.

Riskfolio-Lib calculates optimum portfolios and quantitative strategic asset allocation. This in turn results from optimization of one of the following functions: Maximum Return Portfolio, Minimum Risk Portfolio, Maximum Utility Portfolio and Maximum Risk Adjusted Return Ratio Portfolio. I used this library to compose and organize the portfolios and assets.

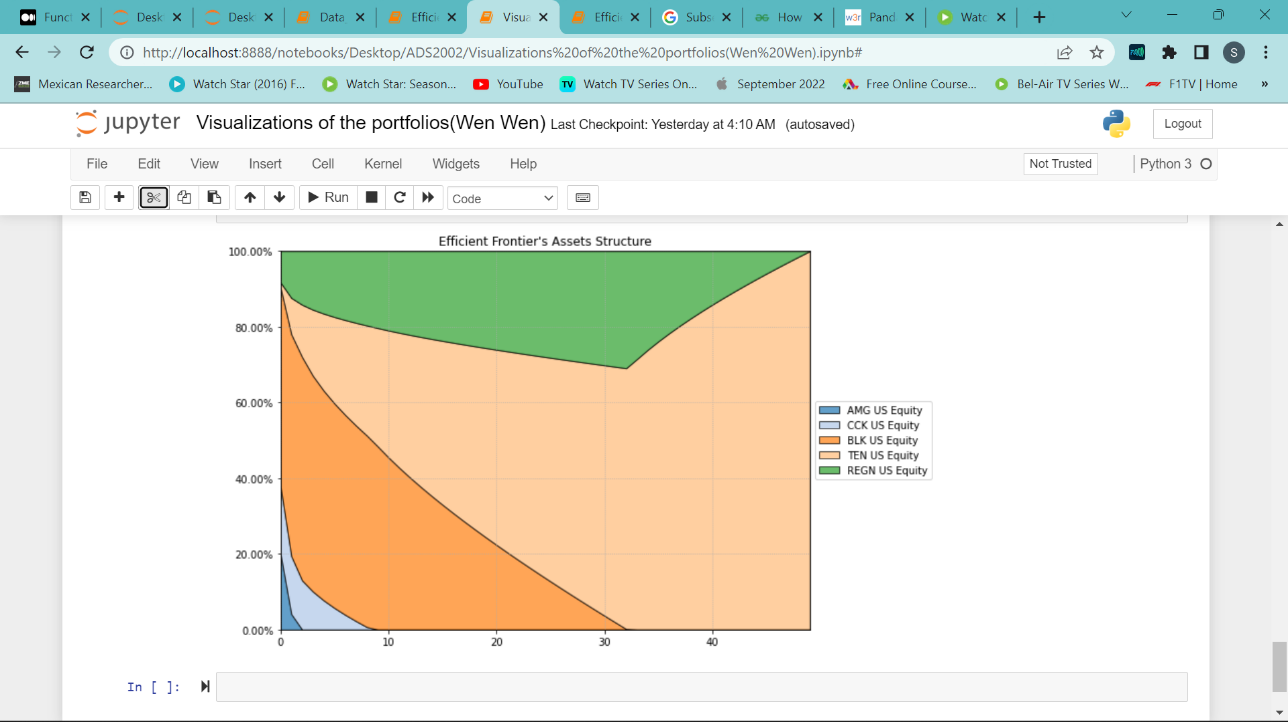
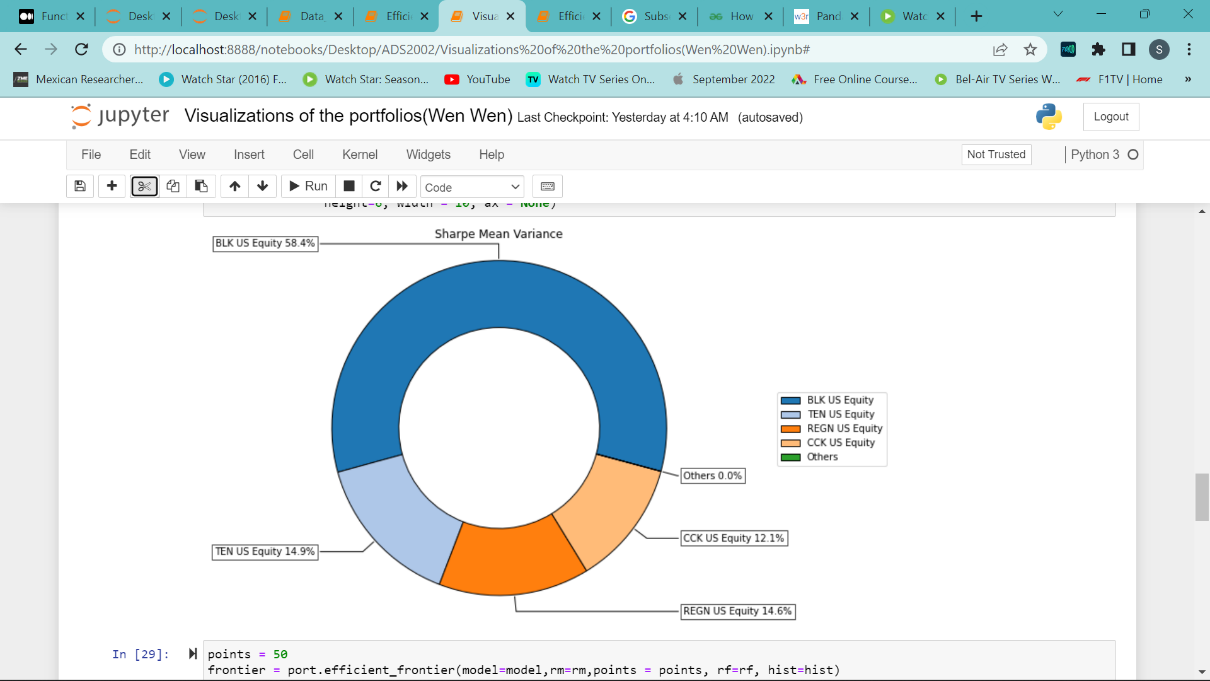


Figure 11:Portfolio Composition Figure 12: Assets structure

BLK US Equity takes up the most in the portfolio composition similarly to figure 10. However, it takes up the least in Asset structure, it is not known why.