LIBERA UNIVERSITÀ INTERNAZIONALE DEGLI STUDI SOCIALI

Master's Degree in Corporate Finance

Department of Business and Management - Chair of Risk Management

Machine Learning Portfolio Optimization:

Hierarchical Risk Parity and Modern Portfolio Theory

SUPERVISOR

Professor Francesco Cerri

CO - SUPERVISOR

Professor Marco Vulpiani

CANDIDATE

Luca Bechis

709261

ACADEMIC YEAR 2019 – 2020

To Eugenio

Table of Contents

INTRODUCTION

The Portfolio allocation strategy and construction is perhaps one of the most challenging and frequent issues in the asset management industry. Every day, millions of investors around the world seek to maximize their investment in order to achieve their most important financial goals: investing for retirement, buying a house, paying for college or simply earning returns in excess of a particular market benchmark. They try to build up portfolios today that can deliver the financial outcomes they need in the future.

A brilliant 24 years old economist, Harry Markowitz, with the publication of his famous paper "*Portfolio Selection"* in 1952, marked a milestone in the portfolio theory by introducing the mean-variance portfolio allocation approach. Regarded the father of Modern Portfolio Theory (MPT), Markowitz was the first scholar recognizing that various levels of risks are associated with different optimal portfolios depending on the investor's risk-return preferences. His mean-variance approach was revolutionary since it provided scholars and asset managers with an intuitive quantitative framework to adopt. Before Markowitz work, investment managers usually considered the optimal portfolio as the one achieving the highest expected return. Since it is enormously difficult to optimally allocate a portfolio with the highest expected return, investors should reconsider distributing their resources across alternative investments to build a more diversified portfolio. Furthermore, Markowitz argued that investors, when allocating their wealth, should be interested in only two distinct yet interrelated elements: the expected return of an investment and the risk of the same. According to the mean-variance environment, everyone faces a trade-off when constructing his/her optimal portfolio. Indeed, there is a one to one direct relationship between the variance and the return of an investment. Risk-averse investors would be willing to give up a bit of return in change of safer portfolios, while risk-seeking people want to maximize the expected return no matter the variance. Markowitz argued that the only efficient portfolios are those that for any given amount of volatility, have the highest possible expected return. The set of all these portfolios would build up the so called "efficient frontier" where investors construct their investment choices according to their specific risk-return preferences.

Markowitz, considered the father of quadratic programming, implemented his groundbreaking mean-variance approach into an optimization algorithm, the Critical Line Algorithm (CLA).

Indeed, the Markowitz efficient frontier solution requires both an equality constraint (that the portfolio's weights sum up to one) and an inequality constraint (a lower and upper bound for the weights, which are 0 and 1 respectively), in order to be solved. As there is no analytic solution to this problem, the breakthrough of the young American economist was to develop an open-source algorithm that could solve inequality-constrained portfolio optimization problems and compute the optimal set of efficient portfolios lying on the curve.

Since Markowitz work, asset managers as well as academics around all the world have been focusing on carrying out theories and methodologies to construct robust portfolios that could minimize the risk while still securing an "alpha". The financial industry, however, is a very volatile one: a constant critical investigation of the strategies and approaches analyzing the risk-return relationship is therefore always required. The global financial crisis of 2008 has displayed all the limitations and drawbacks of the traditional portfolio allocation methodologies. The Markowitz efficient frontier theory has demonstrated to lead to inconsistent outcomes especially due to the challenges in estimating the expected returns and the covariances for the different asset classes. Furthermore, even the CLA solution somewhat produced unstable results. Indeed, small deviations in the forecasted returns lead the algorithm to develop very different portfolios. The recent credit financial crisis unveiled even more the portfolio diversification and performance weaknesses, raising the need in the asset management industry to build new theoretical frameworks with strong empirical results. Among the new portfolio allocation approaches, the ones that grasped the most attention from practitioners and researchers are the so-called risk-based strategies. Since the expected returns are considered unpredictable, these new methodologies try instead to estimate the risk factors and focus mainly on the covariance matrix. The new portfolio weights depend only on the specific risk factors affecting each security in the portfolio. Some of the most relevant risk- based models include: Equal Risk Contribution Portfolio (ERC), Risk Parity Portfolio (RP), Global Minimum Variance (GMV), Maximum Diversification Portfolio (MDP), Maximum Sharpe Ratio Portfolio (MSP), Inverse Volatility Strategy (IV) and Market-Capitalization-Weighted Portfolio (MCWP).

However, "*dropping the forecasts on returns does not prevent the instability issues. The reason is, quadratic programming methods require the inversion of a positive-definite covariance* *matrix. This inversion is subject to large errors when the covariance matrix has a high condition number"*[1].

The Hierarchical Risk Parity portfolio allocation approach (HRP) developed and proposed by the Spanish economist Marcos Lòpez de Prado in 2016, tries to fill the gap in the literature, not only by solving the Markowitz algorithm instability issues but also by producing portfolios that could outperform the traditional risk-based allocation strategies. By avoiding the inversion of the covariance matrix and identifying a hierarchical structure in the portfolio weights, HRP applies graph theory and machine learning techniques to construct a diversified portfolio based on the information contained in the covariance matrix. Furthermore, de Prado's famous article "*Building Diversified Portfolios that Outperform Out-of-Sample"* has proven that adopting the HRP algorithm leads to more robust portfolios out-of-sample, characterized by a lower volatility compared to other allocation strategies.

The Markowitz mean-variance approach marks the foundation of every portfolio theory and it is widely known in the world of finance. However, it has shown to be subject to many inconsistencies and issues which do not allow an optimal allocation of the investor's resources. I was intrigued therefore, by the possibilities of the de Prado's Hierarchical Risk Parity model as it provides an innovative and dynamic portfolio optimization framework which remarkably uses a machine learning algorithm, offering a high-level understanding through digital images. I also observed the empirical literature on the HRP model is quite scarce, something which may prevent other practitioners from further adopting the algorithm.

Therefore, this thesis will deeply investigate the traditional allocation approaches and theories as well as their advantages and main limitations. It will then present the HRP portfolio allocation model, making a comparison with the risk-based methodologies. In the end, it will evaluate and study two different portfolios: an index-based portfolio built with the 30 securities of the Dow Jones Index and all ETFs portfolio, consisting of the 15 most liquid ETFs tracking the major index in the US. The main object of the thesis is to analyze which allocation strategy applied on the two afore-mentioned portfolios, outperforms both in-sample and out-of-sample in terms of

[1] de Prado, M. L. (2016). Building diversified portfolios that outperform out of sample. *The Journal of Portfolio Management*, *42*(4), 59-69.

weights diversification, risk minimization and performance metrics. Therefore, the main research questions of the thesis can be formulated in the following way:

- How do different portfolio models allocate wealth?
- How concentrated are the HRP portfolio weights relative to other traditional allocation strategies?
- Does an HRP based portfolio outperform the traditional allocation strategies in and out-of-sample?
- Does the HRP algorithm better apply on an index-based portfolio (the Dow Jones Index portfolio) compared to an all ETF portfolio?

It is my desire that this thesis will shed some further light on the Hierarchical Risk Parity approach as I believe it presents the reader with a useful tool in the asset allocation process. My genuine passion for the financial world, and for the portfolio and risk management studies, motivated me to carry out this research in which I am going to assess the approach using a computer programming language. All the empirical analysis will indeed be codified and implemented on the Anaconda Jupyter notebook of the Python computer language, version 3.7.

This thesis is organized as follows. In the 1st chapter, a comprehensive study of the Modern Portfolio theory and risk-based portfolio strategies is offered. The chapter, starting from Markowitz, goes over the main portfolio allocation methodologies, outlining their strengths and weaknesses. A practical application on Python comparing two different portfolio strategies is provided at the end of the chapter. Chapter 2, instead, goes directly to the crucial topic of the thesis: The Hierarchical Risk Parity algorithm. It firstly analyzes the Markowitz Critical Line Algorithm and then it points out how and why the HRP model outstands the CLA approach. At the end of the chapter is reported an application on Python to better visualize and understand how the HRP works. In the end, chapter 3 provides the empirical analysis of the different portfolio allocation strategies using the two chosen portfolios and tries to give an answer to the research questions of the thesis.

1. THE DEVELOPMENT OF MODERN PORTFOLIO THEORY

1.1 Markowitz Portfolio Theory

Portfolio optimization problem is a cornerstone area in finance. One of the fundamental assumptions made in finance is that, due to the scarcity of resources, all economic choices face some form of trade-off. When coming through an investment decision, the main kind of "compromise" a rational investor has to encounter is the choice between how much return he would like to earn and the amount of risk he is willing to accept, given that return. Consequently, a critical phase of the investment process should concentrate on questioning about how and where to allocate your financial resources, rather than merely selecting the securities to own.

Around the fourth century a certain Rabbi Issac bar came up with an interesting yet oversimplified rule for asset allocation: "*One should always divide his wealth into three parts: a third in land, a third in merchandise and a third ready to hand"*[2]. The American economist Harry Markowitz, used Isaac's rule himself when asked about how he manages his own funds, stating that: *"My intention was to minimize my future regret. So, I split my contribution fifty-fifty between bonds and equities"*[3]. The previous two statements, though presenting a very rough and simple scenario, both underline the importance of splitting the wealth across different categories of assets. Economists usually quote the Miguel Cervantes Don Quixote phrase *"do not put all your eggs in the same basket"* to explain the diversification concept in finance. Diversification, in portfolio management, is regarded as a "free lunch"; indeed it allows investors to increase their portfolio expected return while keeping the level of risk unchanged or *vice-versa*, to decrease the portfolio risk without affecting the total expected return. In the 1950s, Markowitz, the father of the worldwide accepted Modern Portfolio Theory (MPT), developed a thesis that could explain investors rationale when trying to allocate their resources in the most efficient way. Although diversification cannot completely cancel portfolio variance out, there is a rule according to which the *"investor does (or should) diversify his funds among all those securities*

[2] See Babylonian Talmud: Tractate Baba Mezi'a, folio 42a.
[3] Zweig J. (1998). "Five investing lessons from America's top pension fund", *Money*, 115-118.

which give maximum expected return"[4]. This principle assumes there is a portfolio having both maximum expected return and minimum variance. Markowitz theory, today known as the Modern Portfolio Theory, represents the foundation and the basis of all Investment Management literature and portfolio optimization methods. It managed to formulate the "optimal" approach for allocating resources across risky securities in a static world where people are only interested in the mean and variance of the portfolio's return. The MPT provides a formal yet tractable procedure to find optimal portfolios which will build the "efficient frontier" defined as the bundle of optimal portfolios that show the highest expected return for a given level of risk or the lowest risk for a given level of expected return.

1.1.1 <u>The Mean-Variance portfolio model</u>

In March 1952 Harry Markowitz published the famous paper "Portfolio Selection" in which he first presented the theory of a portfolio optimization based on a mean-variance trade-off. *"The investor does (or should) consider expected return a desirable thing and variance of return an undesirable thing"*[5]. According to the American economist, following certain assumptions and conditions, investors decision can be reduced only to the expected return and the variance of the portfolio.

Markowitz rejects the hypothesis that the investor maximizes discounted return. Indeed, the previous theories did not take into account the benefits of diversification[6]. The practice of merely maximizing discounted expected returns leads to considering two securities with the same value as good as any combination of them. In doing so, the investor would be indifferent to the portfolio's weights since any "mix" of different assets would yield the maximum expected return. But according to Markowitz, *"Diversification is both observed and sensible; a rule of behavior which does not imply the superiority of diversification must be rejected both as a hypothesis and as a maxim"*, therefore investors should diversify their financial resources over a variety of securities[7]. The only efficient portfolios are those that for any

[4]Harry Markowitz (1952). "Portfolio Selection", *The Journal of Finance*, 7(1), <u>77-91</u>.
[5] Markowitz (1952), op. cit., pp.<u>77-78</u>.
[6] See Williams, J. B. (1938). "The theory of investment value" (No. HG4521 W48).
[7] Markowitz (1952), op. cit., pp.<u>79.</u>

defined amount of variance, have the highest possible expected return. The set of all these portfolios would construct the so called "efficient frontier" where investors have to build their investment choices according to their specific risk-return preferences. The decision of where to invest does not depend anymore on the distinct feature of the security. Indeed, each investor has to consider how each security co-moves with all the others in the portfolio. The total risk of the portfolio will be affected not only by the single assets' returns variance but also by the set of covariances of all the securities. Therefore, it is essential to take into account all the possible interactions among the different investments. By doing this, the investor creates a portfolio with the same expected return but with a lower risk than a portfolio that does not consider these pairwise assets co-movements.

The mean-variance optimization problem requires looking for the portfolio weights w_i that maximize the portfolio expected return $\mathbb{E}(R_p)$ given the variance $\sigma^2{}_p$ of a portfolio p.

The expected return of a portfolio is the weighted average (weighted by the probability of the outcomes) of A: the total population of returns of each portfolio constituent.

In numerical terms we do have:

$$\mathbb{E}(\mathrm{R}_i) = \mathrm{p}_{i1}r_{i1} + p_{i2}r_{i2} + \cdots + p_{iA}r_{iA}$$

Or

$$\mathbb{E}(\mathrm{R_i}) = \sum_{k=1}^{A} p_{ik}r_{ik} \ (1)$$

Where p_{ik} represents the probability of the event k of the security i's return and r is the return of the individual asset i. In the special case where the probabilities of the events are exactly the same the previous formula becomes the simple average of all possible different events.

$$\mathbb{E}(\mathrm{R_i}) = \sum_{k=1}^{A} \frac{r_{ik}}{A} \ (2)$$

The variance $\sigma^2{}_i$ instead, is the squared deviation of a random variable from its mean value. It tells how much the returns of the single assets deviate from the mean portfolio return.

$$\sigma_i^2 = \sum_{k=1}^{A} [p_{ik}(r_{ik} - \mathbb{E}(R_i)^2] \ (3)$$

Or in the case of the same probabilities

$$\sigma_i^2 = \sum_{k=1}^{A} \frac{(r_{ik} - \mathbb{E}(R_i)^2}{A} \ (4)$$

The variance is the commonly used measure in finance to express risk or volatility. The standard deviation is a static measure of the dispersion of the population observations from its mean value. It is computed as the square root of the variance.

$$\sigma_i = \sqrt{\sigma_i^2} \ (5)$$

All the equations outlined so far are used for a single security in the portfolio, which is the starting point for an asset allocation study or portfolio analysis. If an investor wants to allocate a portion X_i of his financial resources into different assets constituting a portfolio p, the portfolio total expected return in given by (6)[8].

$$\mathbb{E}(R_p) = \sum_{i=1}^{N} X_i \mathbb{E}(R_i) \ (6)$$

Since each investor has to consider the co-movements of each security in the portfolio, the total risk cannot be expressed merely by the variance of the portfolio. It depends also on the covariance of all the asset returns. The covariance of the returns of two assets l and q, if joint outcomes have the same likelihood, is given by

[8] Elton, E.J., Gruber, M.J. (2011). “Modern Portfolio Theory and Investment Analysis”, *Wiley*, 9.

$$\sigma_{lq} = \sum_{k=1}^{A} \frac{(r_{1k} - \mathbb{E}(\mathrm{R}_1))(\mathrm{r}_{2\mathrm{k}} - \mathbb{E}(\mathrm{R}_2))}{A}$$

Hence, the variance of the total portfolio σ_p^2 can be calculated as

$$\sigma_p^2 = X_l^2\sigma_l^2 + X_q^2\sigma_q^2 + 2X_lX_q\sigma_{lq} \; (7)$$

From the covariance formula it is possible to obtain the correlation coefficient which measures how two securities move in relation to each other.

$$\rho_{lq} = \frac{\sigma_{lq}}{\sigma_l\sigma_q}$$

The coefficient ranges from -1 to +1. A value of +1 means the two securities are perfectly correlated, therefore if one exhibits a one period return of 0.8 so does the other. By exchanging the covariance term in (7) with the correlation coefficient:

$$\sigma_p^2 = X_l^2\sigma_l^2 + X_q^2\sigma_q^2 + 2X_lX_q\sigma_l\sigma_q\rho_{lq}$$

The general expression for the portfolio variance in the case of N assets is given by (8)[9].

$$\sigma_p^2 = \sum_{k=1}^{N}(X_k^2\sigma_k^2) + 2\sum_{k=1}^{N}\sum_{j>1}^{N}\left(X_kX_j\sigma_{kj}\right) (8)$$

By taking the square root of the variance of the portfolio we have the standard deviation of the portfolio p, σ_p.

$$\sigma_p = \sqrt{\sigma_p^2}$$

In equation (8) the diversification effect is defined by the first term of the left side part, which indeed depends only on the individual variances of the securities. The more assets are included

[9] Markowitz (1952), op. cit., pp.81.

in the portfolio, the more that side of the equation gets closer to 0. Hence, those investors who allocate their total wealth into different assets, will see the total variance of the portfolio decreasing. However not the whole portfolio volatility can be diversified away. As more assets are included in the investments band, the variance is reduced only up to a certain limit. The right-side term of the equation (8) indeed, depends only by the covariance of the assets and is not affected by the number of securities in the portfolio[10].

1.1.2 The efficient frontier

Markowitz was the first economist that in the 1950s developed a theory providing a meticulous mathematical structure for portfolio optimization. The mean (return) – variance (volatility) problem can be solved using two-dimensional geometry. The search of the weights w_k that maximizes the $\mathbb{E}$ (R_{p}) given the σ_p of the portfolio p, creates the so called “investments opportunity set” or the investor’s risk-return area. This bidimensional space is characterized by all the possible combinations of expected return $\mathbb{E}$ (R_k) and variance σ_k of the portfolios.

Since I am going to work with a portfolio composed by many securities, for the sake of completeness I am going to re-write the previous formulas using vectors and the matrix form.

Markowitz considers a universe of N securities and a vector of weights in the portfolio $X = (X_1, \ldots, X_n)$ which are not random variable but are fixed by the investors[11]. Two main assumptions are made to develop the theoretical framework. The first one states that the portfolio is fully invested. Therefore, being X_k a percentage term, all the weights sum up to one:

$$\sum_{k=1}^{N} X_k = 1^T X = 1 \ (9)$$

[10] For a deeper study, see Elton, E. J., & Gruber, M. J. (1997). Modern portfolio theory, 1950 to date. *Journal of Banking & Finance*, *21*(11-12), 1743-1759.
[11] Markowitz (1952), op. cit., pp.83

Moreover, Markowitz does not consider the case for short-sales, therefore X cannot take negative values[12].

$$\forall i X\_i \geq 0$$

The vector of asset returns is R = R (R_1, …., R_n) and the return of the portfolio p is the weighted sum of the returns of the single securities.

$$R_p = \sum_{k=1}^{N} X_k R_i = X^T R \ (10)$$

By defining the vector of the expected values of the returns $\bar{R} = \mathbb{E}\,(R_i)$ it is possible to determine the variance-covariance matrix Ω[13].

$$\Omega = \mathbb{E}[(R - \bar{R})(R - \bar{R})^T] \ (11)$$

Therefore, the expected returns and variance of the portfolio are respectively:

$$\bar{R}\,(X) = \mathbb{E}[\mathrm{R(X)}] = \mathrm{X}^{\mathrm{T}}\bar{R} \ \ (12)$$

$$\sigma^2(X) = \mathbb{E}\left[\left(\mathrm{R(X)} - \bar{R}\,(X)\right)\left(\mathrm{R(X)} - \bar{R}\,(X)\right)^T\right] = X^T \Omega \mathrm{X} \ (13)$$

The Markowitz mean-variance optimization problem requires either to minimize the total portfolio variance subject to a lower limit on the expected return or to maximize the investor expected utility that can be fully described by the mean and variance of the portfolio.

$$\begin{aligned} &min\ X^T \Omega \mathrm{X}, \\ &\qquad w.r.t. \qquad\qquad (14) \\ &\mathrm{X}^{\mathrm{T}}\bar{R} > \mathbb{E}^{\circ} \end{aligned}$$

[12] A short sale is basically when an investor sells an asset that is not included in his inventory, thus borrowing an asset from someone who does (an investment bank) and sells it on the financial marketplace.

[13] A variance-covariance matrix is a square matrix composed by the variances and covariances linked with several securities. The diagonal elements of the matrix contain the variances of the variables and the off-diagonal elements contain the covariances between all possible pairs of variables.

$\mathbb{E}^{\circ}$ is the specific expected return target the investor wishes to achieve given the portfolio variance.

At each time the investor tries to maximize his utility's preferences function by selecting the different weights X_k to be allocated in the N risky assets.

$$\max X_k^T \, \overline{R} \, - \frac{\delta}{2} X_k^T \, \Omega_k \, X_k \ (15)$$

Where δ is a coefficient representing the investor specific risk aversion. The vector weights in the portfolio composed by N risky asset, $X = (X_1, \ldots, X_n)$, at time t is:

$$X_t = \frac{\Omega_{\mathrm{t}}^{-1} \, \overline{R}_t}{1_N \, \Omega_{\mathrm{t}}^{-1} \, \overline{R}_t} \ (16)$$

Where 1_N is a N dimensional vector of ones. The problem can be solved with a plug-in procedure starting by equation (15) and consequently substituting the respective values of mean and covariance matrices[14].

The Markowitz Efficient Frontier curve is characterized by the set of all the optimal portfolios in which the trade-off between risk and return is maximized. The portfolio's investment opportunity space can be plotted as a graph with the x-axis showing the standard deviation of all different portfolios and the y-axis the expected return. All the portfolios that lay down the efficient frontier curve (the investment opportunity area) are sub-optimal since in that case, there will always be a better combination of mean and variance. Therefore, the efficient frontier curve is just the representation of all the optimal portfolios in the investment universe. The portfolio placed on the leftmost point of the curve is the one exhibiting the lowest possible variance.

[14] DeMiguel, V., Garlappi, L., & Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy?. *The review of Financial studies*, *22*(5), 1915-1953.

1.2 The Capital Asset Pricing Model (CAPM)

In 1964, the brilliant economist William Sharpe, published the paper “Capital asset prices: A theory of market equilibrium under conditions of risk”, that would have become one of the most inspiring financial models for estimating firms cost of capital and evaluating portfolios ‘performance. The model proposed and developed by Sharpe is widely considered as the one marking the birth of asset pricing theory.

The Capital Asset Pricing theory (CAPM) became another evidence for researchers and investors of the interconnections between asset risk and asset return. According to Sharpe: “*In equilibrium there will be a simple linear relationship between the expected return and standard deviation of return for efficient combinations of risky assets*”[15].

Starting from Markowitz, Sharpe further develops the mean-variance portfolio theory by assuming that a rational investor would allocate his wealth in such a way that his optimal set of portfolios would lie anywhere on the security market line (SML). Together with the contribution of Lintner and Mossin, he defines the condition of an equilibrium in a market characterized by investors having the same risk preferences (all investors are rational, and all rational investors are risk-averse) and with the same interest rate for borrowing and lending[16]. The CAPM, unlike the Markowitz approach, introduces the important concept of systematic and unsystematic risk. Indeed, Sharpe distinguishes among that part of risk which is correlated with the market unexpected events (systematic risk) and that variance of returns which doesn’t vary with the market, but rather depends on the company/industry specific characteristics. This idiosyncratic or unsystematic risk doesn’t affect the whole market and has an impact at the microeconomic level. It can be reduced or even eliminated through the effect of the portfolio diversification. On the other hand, the former kind of risk cannot be canceled out in portfolio construction and affects all the securities (e.g. a change in the interest rate or inflation). Investors of the CAPM world, are rewarded with a higher expected return, only for

[15] Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *The journal of finance*, *19*(3), 425-442.

[16] Lintner, J. (1965). Security prices, risk, and maximal gains from diversification. *The journal of finance*, *20*(4), 587-615; Mossin, J. (1966). Equilibrium in a capital asset market. *Econometrica: Journal of the econometric society*, 768-783.

taking on the systematic risk, represented by Beta (β). This risk distinction represents a clear difference with the mean-variance model where all assets take on the same kind of variance.

Having all homogenous market expectations and with borrowing and lending at a risk-free rate, all investors will hold in equilibrium the same risky portfolio. Littner and Sharpe in their paper demonstrate that if the condition of market equilibrium is respected, everyone will hold the market portfolio[17]. The relationship between the expected return and the systematic risk or Beta is graphically represented by the Security Market Line whose equation (17) defines the expected return for all assets and portfolio of assets:

$$\mathbb{E}(\mathrm{R_i}) = R_f + \beta_i\left[\mathbb{E}(\mathrm{R}_i) - R_f\right](17)$$

In this equation $\mathbb{E}(\mathrm{R_i})$ is the expected return on asset *i* while β_i measures the security 'sensitivity to the market movements and it is algebraically equal to

$$\beta_i = \frac{\mathrm{cov}(\mathrm{R}_i, \mathrm{R}_M)}{\sigma^2(\mathrm{R}_M)} \ (18)$$

Where R_M is the market return. Equation (18) tells us that the market Beta of asset *i* is equal to the covariance between the security's return and the market one, divided by the variance of the market return.

A better β_i interpretation is given by the two American researchers Eugene F. Fama and Kenneth R. French : "*Since the market beta of asset i is also the slope in the regression of its return on the market return, a common (and correct) interpretation of beta is that it measures the sensitivity of the asset's return to variation in the market return. But there is another interpretation of beta more in line with the spirit of the portfolio model that underlies the CAPM. The risk of the market portfolio, as measured by the variance of its return (the denominator of* β_i*), is a weighted average of the covariance risks of the assets in the market M (the numerators of* β_i *for different assets). Thus,* β_i *is the covariance risk of asset i in M measured relative to the average covariance risk of assets, which is just the variance of the*

[17] Elton et. Al. (2011), op cit., pp. 272.

market return. In economic terms, β_i is proportional to the risk each dollar invested in asset i contributes to the market portfolio"[18].

1.2.1 A graphical representation of the CAPM

As mentioned above, the Security Market Line (SML) is the graphical representation of the relationship between Beta and the asset return. As the name suggests, this relationship is a linear one whose intercept is the risk-free rate where the investors have 100% of exposure to the portfolio and the slope is the Beta coefficient. The graph can be better analyzed through the following simple example, considering 5 different stocks[19].

I assume a risk-free rate of 3% and an equity risk premium of 5%. The SML analysis could tell us whether the securities in which we want to invest are overvalued or undervalued with respect to the market. For the 5 stocks under consideration, we observe the following Beta coefficients and required rate of return observed in the market by using the dividend discount model *(Figure 1):*

Stock	**Beta**	**Observed Required Return**
A	1,1	8%
B	1,3	12%
C	1,5	9%
D	1,9	15%
E	0,7	7%

Figure 1, Market Returns & Beta Coefficient. Data Source: Personal elaboration on Excel.

By applying the Capital Asset Pricing Model equation (17) we can extract the required rate of return on each stock as we can observe in the last column of *Figure 2* reported below.

[18] Fama, E. F., & French, K. R. (2004). The capital asset pricing model: Theory and evidence. *Journal of economic perspectives*, *18*(3), 25-46.
[19] The data of the example considered are not real market data but they are the result of a personal elaboration on Excel.

Stock	**Beta**	**Observed Required Return**	**CAPM Return**	**Formula (17)**
A	1,1	8%	9,6%	=3% + 1,1 x 6%
B	1,3	12%	10,8%	=3% + 1,3 x 6%
C	1,5	9%	12,0%	=3% + 1,5 x 6%
D	1,9	15%	14,4%	=3% + 1,9 x 6%
E	0,7	7%	7,2%	=3% + 0,7 x 6%

Figure 2, CAPM Returns & Beta Coefficient. Data Source: Personal elaboration on Excel.

If the CAPM rates of return are plotted together with the Beta coefficients we obtain the SML (*Figure 3)* whose intercept is given by the risk-free rate and slope by the different Betas. The point on the line having a β of 1 represents the market portfolio which therefore has a one to one direct relationship whit movements in the market.

In the given example the stocks A, B and D are undervalued with respect to the market, because for a given amount of risk (β), they yield a higher return. On the other hand, the securities C and E are overvalued because for a given amount of risk, they yield a lower return.

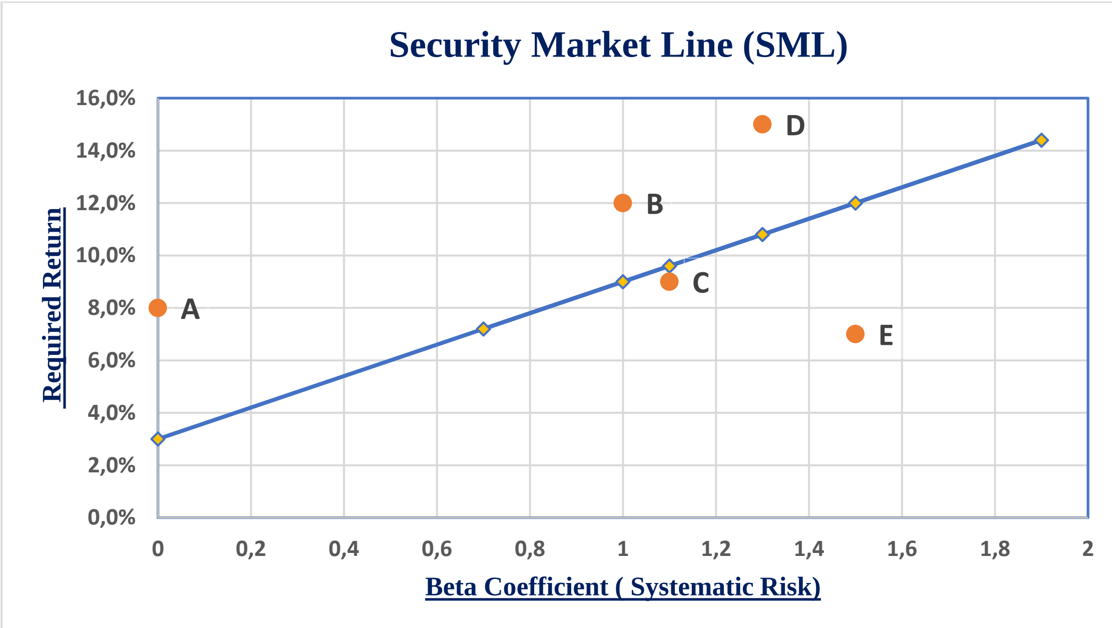


Figure 3, The Security Market Line (SML). Data Source: A personal elaboration on Excel.

1.2.2 Critiques of the Capital Asset Pricing theory

Although widely used in the academic financial world due to its simplicity and well-defined framework for estimating the cost of capital, the CAPM model presented many drawbacks in the empirical applications. Indeed, empirical literature demonstrates that the relationship between Beta and the required rate of return in the Sharpe-Littner model is too steep if compared to what really happens in the market when tested with real data. As a consequence, CAPM estimates of the cost of equity for high beta stocks are too high and estimates for low beta stocks are too low[20].

CAPM assumptions are considered too unrealistic and restrictive. Several other asset pricing models that take unsystematic risk into account, assume that investors for some exogenous reasons hold undiversified portfolios. An instance of the Sharpe theorem evolution is the

[20] Friend, I., & Blume, M. (1970). Measurement of portfolio performance under uncertainty. *The American economic review*, *60*(4), 561-575.

"Expanded CAPM" which includes in the equation two additional risk premiums, for small size and one attributable to the specific company or to the industry[21].

One of the main pitfalls of the CAPM equation is that it does not take into account all the possible sources of risk when determining the cost of capital. Fama & French three-factors method, add an important contribution in the practical analysis, since according to the two American economists : " *Two easily measured variables, size and book-to-market equity, combine to capture the cross-sectional variation in average stock returns associated with market β, size, leverage, book-to-market equity, and earnings-price ratios. Moreover, when the tests allow for variation in β that is unrelated to size, the relation between market β and average return is flat, even when β is the only explanatory variable*"[22].

The Fama-French 3 factors model is empirically driven, nonetheless not theoretically supported. One study contrasting the CAPM and FF for the U.S. markets found that whereas differences in the CAPM beta explained on average 3% of the cross-sectional differences in the stock returns over the following year, the FF betas explained on average 5% of the differences[23].

Another important step through the development of the Portfolio optimization theory was made by the Black and Litterman model for portfolio allocation. The two financial scholars believed that many of the previous mathematical theories and models had not been able to achieve the expected empirical results. In order to reduce estimation errors the two economists designed a model which, using a Bayesian approach, tries to combine the subjective views of an investor regarding the expected returns of one or more assets with the market equilibrium vector of expected returns (the prior distribution) to form a new, mixed estimate of expected returns. The resulting new vector of returns (the posterior distribution), leads to intuitive portfolios with sensible portfolio weights[24].

---

[21] For a deeper study see: Merton, R. C. (1980). *On estimating the expected return on the market: An exploratory investigation* (No. w0444). National Bureau of Economic Research; Pratt, S. P., & Grabowski, R. J. (2008). *Cost of capital*. John Wiley & Sons.
[22] Fama, E. F., & French, K. R. (1992). The cross-section of expected stock returns. *the Journal of Finance*, *47*(2), 427-465.
[23] Bartholdy, J., & Peare, P. (2005). Estimation of expected return: CAPM vs. Fama and French. *International Review of Financial Analysis*, *14*(4), 407-427.
[24] Idzorek, T. (2007). A step-by-step guide to the Black-Litterman model: Incorporating user-specified confidence levels. In *Forecasting expected returns in the financial markets* (pp. 17-38). Academic Press.

1.3 The Traditional Risk Based Approaches

The problems of the Markowitz mean-variance methodology, such us estimation errors and inconsistency, led to the development of several other attempts by academics to find possible portfolio solutions that could result in an optimal asset allocation. The mean variance theory has many practical drawbacks due to the difficulties in estimating the expected returns and the covariances for the different asset classes. Portfolios diversification and performance problems became even more evident during the recent credit financial crisis of 2008 which raised the need in the asset management industry, to build new theoretical frameworks with strong empirical results. The new models are risk-based ones, meaning that they try to estimate the risk factors rather than the expected returns which are considered unpredictable. The new portfolio's weights do not consider expected returns and depend only on the specific risk factors affecting each security in the portfolio.

Despite the common view that diversification failed during the recent credit crisis, Risk Parity strategies passed an acid test in 2008 by performing well relative to traditional portfolios[25]. A commonly used practice was to allocate a 60% portion to equities, given their higher returns, and a lower one (40%) to bonds following an equity portfolio risk contribution of almost 90% due to the much higher volatility exhibited by stocks. However, this strategy resulted in very poor performance mainly because of the lack of diversification. Therefore, risk parity portfolios tried to fill the gap by investing in different asset classes in order to spread evenly the whole market risk to each category. Instances of new asset allocations include bonds, equities, real estate, commodities, hedge funds, etc. The rationale behind Risk Parity approaches is to split the percentage risk contribution of each asset class. But why should investors be interested into each asset class risk contribution? It has been empirically shown that risk contribution is a very accurate indicator of loss contribution[26].

Analyzing the typical 60/40 portfolio strategy on a portfolio based on the Russel 1000 and Lehman Aggregate Bond Indices, let us better understand the limits of poor diversification which leads to a higher total expected loss.

---

[25] Hurst, B., Johnson, B., & Ooi, Y. H. (2010). Understanding risk parity. *AQR Capital Management*.

[26] Qian, E. E. (2005). On the financial interpretation of risk contribution: Risk budgets do add up. *Available at SSRN 684221*.

Loss	Equity	Bonds	# of components
2%	95,6%	4,4%	44
3%	100,1%	-0,1%	25
4%	101,9%	-1,9%	15

Figure 4, Average loss contribution for the 60/40 portfolio based on the Russell 1000 and Lehman Aggregate Bond Indiced: 1983-2004. Data Source: A personal elaboration of a PanAgora study & Bloomberg Terminal data.

Figure 4 reported above, shows the risk contribution of equity resulting into different approximations of the portfolio expected losses. For losses higher than 4% indeed, stocks have more than 100% weight on the overall risk. On the other hand, the diversification power of the bond class is almost irrelevant, implying that large losses on the stock market would cause most of the portfolio bad performance. The key to Risk Parity is to diversify across asset classes that behave differently across economic environments: in general, equities do well in high growth and low inflation environments, bonds do well in deflationary or recessionary environments, and commodities tend to perform best during inflationary environments[27].Therefore, building a balanced portfolio could lead to much more robust returns. Risk parity portfolios usually invest more in low volatility securities than traditional asset allocation strategies.

Some of the most relevant risk- based models include:

- Equal Risk Contribution Portfolio (ERC)
- Risk Parity Portfolio (RP)
- Global Minimum Variance (GMV)
- Maximum Diversification Portfolio (MDP)
- Maximum Sharpe Ratio Portfolio (MSP)
- Inverse Volatility Strategy (IV)
- Market-Capitalization-Weighted Portfolio (MCWP)

[27] Hurst et. Al., (2010), Op. cit., pp. 2.

The interesting paper *"Risk Parity, Maximum Diversification and Minimum variance: An Analytic Perspective"* published in 2013 by the researchers Clarke, De Silva and Thorley, conducts an empirical analysis of the most prominent risk-based portfolio allocation methods studying their performances and robustness over a sample of 1,000 common stocks in the CSRP database at the end of each month from 1968 to 2012[28]. The results are reported in *Figure 5* and *Figure 6.*

Performance Indicators	**MCWP**	**EW**	**RP**	**MD**	**GMV**
Average Excess Return	5,3%	7,4%	7,4%	5,7%	5,7%
Standard Deviation	15,5%	17,8%	16,6%	19,1%	12,4%
Sharpe Ratio	0,34	0,42	0,45	0,30	0,46
Compound Return	4,2%	5,9%	6,2%	3,9%	5,1%
Market Beta	1,00	1,09	1,01	0,94	0,51
Average Positions	1000	1000	1000	81,9	61,8

Figure 5, Performance of Risk-Based Portfolios from 1968 to 2012. Data source: A personal elaboration from the paper "Risk parity, maximum diversification, and minimum variance: An analytic perspective".

[28] Clarke, R., De Silva, H., & Thorley, S. (2013). Risk parity, maximum diversification, and minimum variance: An analytic perspective. *The Journal of Portfolio Management*, *39*(3), 39-53.

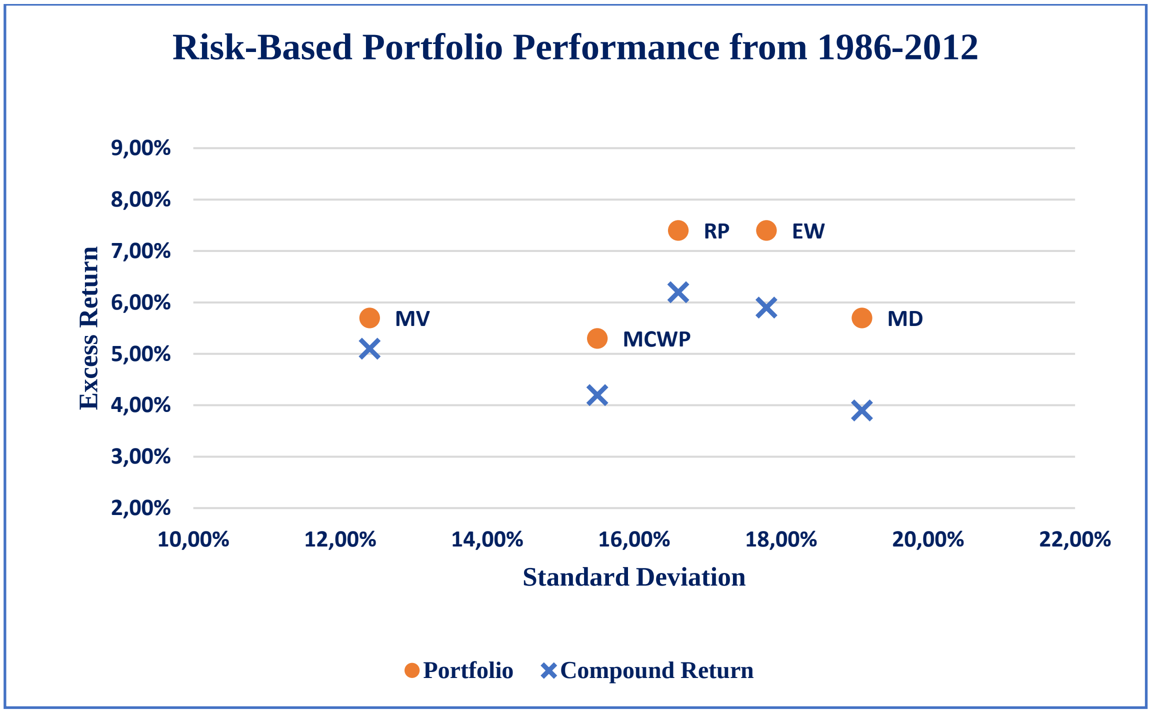


Figure 6, Risk-Based Portfolio Performance from 1986-2012. Data Source: A personal elaboration from the paper "Risk parity, maximum diversification, and minimum variance: An analytic perspective".

1.3.1 Risk based portfolio properties

An important step for better understanding risk parity portfolios is to define the marginal risk parity contribution and how each asset individually contributes to the total portfolio risk. The marginal risk contribution is defined as the change in total risk of the portfolio by an infinitesimal increase of X_k[29]. It determines the effect of a single asset to the entire portfolio risk.

Therefore, the MRC (Marginal Risk Contribution) of a security can be written using the formula (*19*) reported below:

$$MRC_k = \partial \frac{\sqrt{X^T \Omega X}}{\partial X_k} = \frac{(\Omega X)_k}{\sqrt{X^T \Omega X}} \quad (19)$$

[29] Demey, P., Maillard, S., & Roncalli, T. (2010). Risk-based indexation. *Available at SSRN 1582998.*

Where $(\Omega X)_k$ represents the *kth* row of the vector from the product of Ω with *X*[30]. The individual risk contribution of the security *k* instead, is calculated as the product of the weight in asset *k* with its marginal risk contribution as shown by formula (20).

$$RC_k = X_k \frac{(\Omega X)_k}{\sqrt{X^T \Omega X}} \ (20)$$

Since the volatility is a homogenous function of degree 1, it satisfies Euler's theorem[31]. Therefore, it can be written as the sum of its arguments multiplied by their first partial derivatives[32].

The total risk assets contribution of the whole portfolio is:

$$TRC = \sum_{k=1}^{N} RC_k = X^T \frac{\Omega X}{\sqrt{X^T \Omega X}} = \sqrt{X^T \Omega X} \ (21)$$

1.3.2 <u>The Equally Risk Contribution portfolio strategy (ERC)</u>

The ERC portfolio allocation strategy is a portfolio where each security contributes exactly the same amount to the overall portfolio volatility. The ERC portfolio is derived from the simplest techniques of risk budgeting[33]: If we consider that risk and correlation can be predicted easily but that it is impossible to estimate the expected return, then assigning an equal contribution of risk to all of the portfolio elements looks the most reasonable choice. Each weight X_k will be then defined according to the single risk contribution of the asset *k*. If all the securities in the portfolio show the same variance, then the portfolio would be constructed in the same manner as an equally-weighted one. As we will see, even the

[30] Demey et. Al., (2010), Op. cit., pp. 10
[31] The Euler's theorem is a generalization of Fermat's little theorem dealing with powers of integers modulo positive integers. Let *I* be a positive integer and let *b* be an integer that is relatively prime to *I*. Then, $b^{\emptyset(I)} \equiv (mod\ I)$, where $\emptyset(I)$ is the Euler's totient function which counts the number of positive integers ≤ *I* which are relatively prime to *I*. For a better understanding of the Fermat's little theorem see: Smyth, C. J. (1986). A coloring proof of a generalization of Fermat's Little Theorem. *The American Mathematical Monthly*, *93*(6), 469-471.
[32] Roncalli, T., & Weisang, G. (2016). Risk parity portfolios with risk factors. *Quantitative Finance*, *16*(3), 377-388.
[33] Scherer, B. (2007). Portfolio construction and risk budgeting. *Edhec Business School*, *5*.

Minimum Variance (MV) portfolio does equalize risk contributions, but only on a marginal basis and the total risk contributions for each asset in the portfolio is far from equal[34].

According to what outlined so far, an ECR portfolio must satisfy the following property (22).

$$RC_k = RC_j \quad (22)$$

It is important to stress out the differences between a normal Naïve (equal weight) portfolio allocation strategy and the ERC one. Investing evenly among the different assets, therefore allocating the same percentage to each security in the portfolio, is far from being equal to let each asset equally contributing to the whole risk of the portfolio. The example reported below in *Figure 7* illustrates the situation of 3 assets, with the first two being perfectly correlated and the third uncorrelated with either. The main distinction is that the ERC strategy uses the covariance matrix which incorporates information both about the volatility and the correlation framework among the securities. Intuitively, investors understand that an equal dollar allocation to stocks and bonds does not correspond to an equal risk allocation, since historically stock markets have been much more volatile than government bonds[35].

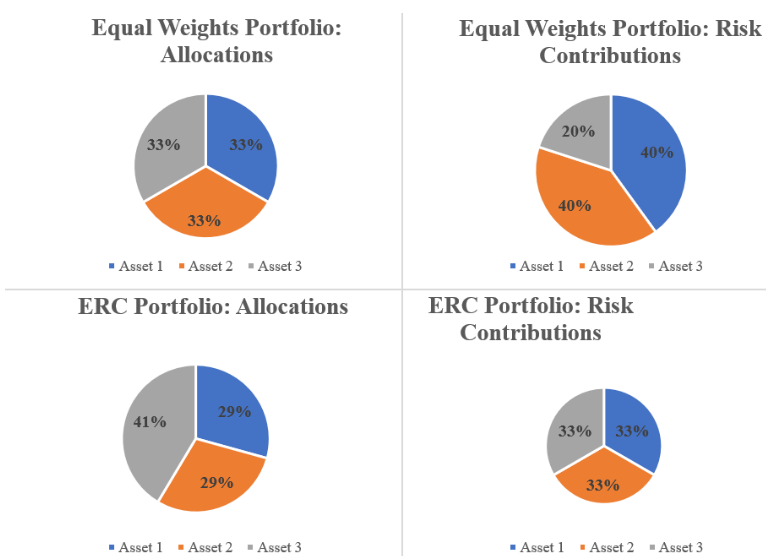


Figure 7, ERC & Naive Portfolio Allocation and Risk Contribution. Data Source: A personal elaboration on excel of J.P.Morgan Asset Management data.

[34] Demey et. Al., (2010), Op. cit., pp. 15.
[35] Forseth, E. & Tricker, Ed., (2019). Equal Risk Contribution Portfolios. *Graham Capital Management, Research Note*.

The rationale behind the ERC strategy is to find the optimal allocation, diversifying the risk rather than capital across all the securities.

By imposing as a constraint that weights sum up to one and by prohibiting short-sales, the problem for a portfolio characterized by *N*-assets can be structured as follows:

$$X_{ERC} = \{X \in [0,1]^N : \sum X_k = 1, \qquad RC(X_k) = RC(X_i) \, , \forall k, i \; (23)$$

By substituting equation (20) to the above:

$$X_{ERC} = \{X \in [0,1]^N : \sum X_k = 1, \qquad X_k(\Omega \mathrm{X})_k = X_i(\Omega \mathrm{X})_i \, , \forall k, i \; (24)$$

If all the correlation coefficients are equal for each *i* and *k* in the portfolio, it is possible to obtain a simple analytical solution to the ERC[36].

$$X_k = \frac{\sigma_k^{-1}}{\sum_{i=1}^{N} \sigma_i^{-1}} \; (25)$$

Therefore, the weight of each security *i* is equal to the ratio of the inverse volatility with the average of all the volatilities. As a consequence, an investor building an ERC portfolio would allocate less to those assets which are considered riskier than others. Since the effect of the correlation coefficients using equation (25) is completely neutralized, this strategy is frequently defined as “Naïve Risk Parity” or “Inverse Volatility”.

By removing the restriction regarding the correlation coefficients, equation (25) can be solved using the so called sequential quadratic programming (SQP) algorithm where the system to be solved is determined as follows[37]:

$$X_{ERC} = \min_{X \in \mathrm{R^N}} f(X)$$

Such that

---

[36] Maillard, S., Roncalli, T. and Teïletche, J. (2010). The Properties of Equally Weighted Risk Contribution Portfolios. The Journal of Portfolio Management, vol. 36(4), pp. 60-70.
[37] Maillard et. Al., (2010), Op. cit., pp. 60-70.

$$\begin{Bmatrix} X^T 1 = 1 \\ 0 \leq X \leq 1 \end{Bmatrix} \; (26)$$

Where

$$f(X) = \sum_{k=1}^{N} \sum_{i=1}^{N} (X_k (\Omega X)_k - X_i (\Omega X)_i)^2 \; (27)$$

Equation (27) calculates the square of the difference between all the combination of the assets risk contributions. In the case *f(X)* is equal to 0, the above system has a unique solution which solves the ERC portfolio allocation problem:

$$X_k (\Omega \mathrm{X})_k - X_i (\Omega \mathrm{X})_i = 0, \forall k, i \; (28)$$

The ERC portfolio seems to have a solution which is in between the Naïve (1/*N*) strategy and the Minimum Variance (MV) allocation problem.

1.3.3 The naïve portfolio strategy- Equally Weighted allocation (EW)

The Equally Weighted Portfolio Strategy is considered one of the easiest ways to create a well-balanced and robust portfolio. It indeed requires to evenly invest investor financial resources into different assets. It is sometimes regarded as a good benchmarking tool, since this strategy does not involve any optimization or estimation technique and completely ignores the data[38].

An important analysis about the performance of the Naïve strategy was carried out by the academics De Miguel, Garlappi and Uppal in 2009 in their paper "Optimal Versus Naïve Diversification: How Inefficient is the 1/N Portfolio Strategy?". The authors want to understand why, despite the development of different portfolio optimization techniques and frameworks, most investors still use these simple rules to allocate their financial resources across assets[39]. By comparing 14 other portfolio allocation methods with the 1/N strategy, the

[38] DeMiguel, V., Garlappi, L., & Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy?. *The review of Financial studies*, *22*(5), 1915-1953.
[39] Indeed, according to different studies Investors allocate their wealth across assets using the naïve 1/N rule. For a further explanation see: Benartzi, S., & Thaler, R. H. (2001). Naive diversification strategies in defined contribution saving

economists show that "*none of them is better than the naïve 1/N benchmark in terms of Sharpe Ratio, certainty equivalent return, or turnover*…."furthermore, "…*.the "allocation mistakes" caused by using the 1/N weights can turn out to be smaller than the error caused by using the weights from an optimizing model with inputs that have been estimated with error*"[40].

The attractiveness of the naïve allocation strategy is without any doubts its easy implementation relying only on one simple allocation rule (29) :

$$X_k = \frac{1}{N}$$

Where X_k is the weight of a single security *k* in the portfolio. Therefore, the vector of all the portfolio securities weights is:

$$X_{EW} = \left(\frac{1}{N}, \dots\dots, \frac{1}{N}\right)^T \quad (29)$$

Although sometimes the naïve strategy performs better than the other traditional portfolio optimization approaches, it still underperforms the efficient frontier of the Markowitz mean-variance theory as we can understand by looking at *Figure 8* below.

plans. *American economic review*, *91*(1), 79-98; Huberman, G., & Jiang, W. (2006). Offering versus choice in 401 (k) plans: Equity exposure and number of funds. *The Journal of Finance*, *61*(2), 763-801.

[40] DeMiguel et. Al., (2009), Op.cit., pp. 1920.

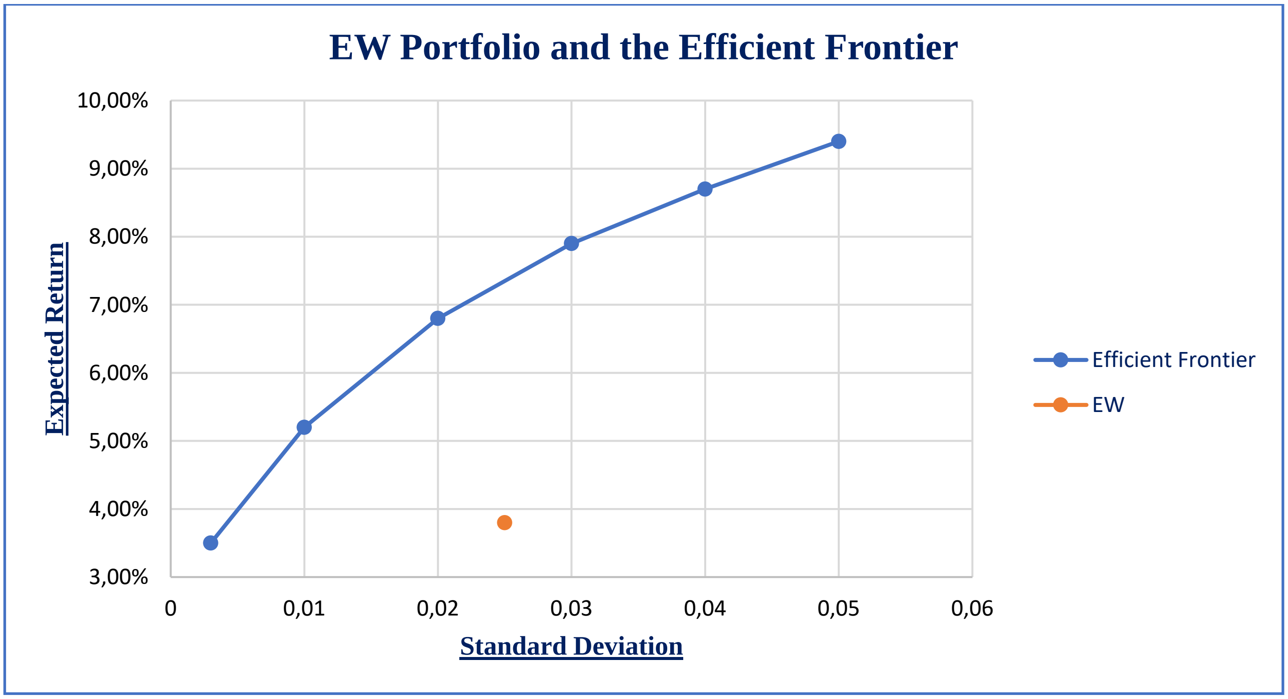


Figure 8, The EW Portfolio Position on the Efficient Frontier. Data Source: A personal elaboration of Bloomberg Data, using Excel[41].

From the above graph, the EW portfolio shows an expected return of 3,8% by maintaining a risk exposure, represented by the standard deviation, of 2,5%. Nonetheless, with the same level of volatility, the Markowitz efficient frontier approach exhibits an expected return of 7,3%.

1.3.4 The Global Minimum Variance strategy (GMV)

According to the Markowitz mean-variance portfolio theory, investors when allocating their wealth across assets, face an important trade-off between increasing the returns as much as possible and lowering the assets returns variance. However, as we have already pointed out, it is impossible to have a higher portfolio performance without accepting more risk. Nonetheless, this strategy is hardly taken by those people who are considerably risk averse.

[41] The data used to construct the graph is gathered from Bloomberg Database. It consists of index prices of 15 European Indices divided in Government Bonds, Corporate Bonds and Equities. The data consider a time-period of 5 years (2013-2018).

Indeed, these types of investors would rather invest in a portfolio that exhibits the minimum possible variance, at the cost of giving up some return.

Along the efficient frontier, the Global Minimum Variance Portfolio (GMV) lies on the left-most point *(Figure 9)*. Its main interpretation is that an investor can't hold a portfolio of risky assets which has a lower volatility than the GMV portfolio. Furthermore, the GMV portfolio shows the highest possible expected return for that level of risk. Being a risk-based portfolio strategy, the GMV approach avoids the estimation of expected return, and all stock portfolios differ only with respect to their risk[42]. Therefore, the GMV portfolio requires only the estimation of the covariance matrix of asset returns[43].

If the short-sales Markowitz restriction is respected and the portfolio's weights sum up to 1:

$$X = \varepsilon[0;1]$$

$$X^T 1 = 1$$

The GMV quadratic optimization problem can be solved through the following algorithm which minimized the standard deviation (30).

$$\begin{Bmatrix} X^* = \min f(X) \\ f(X) = \frac{1}{2} X^T \Omega X \end{Bmatrix} \quad (30)$$

[42] Kempf, A., & Memmel, C. (2006). Estimating the global minimum variance portfolio. *Schmalenbach Business Review*, *58*(4), 332-348.
[43] Chan, L. K., Karceski, J., & Lakonishok, J. (1999). On portfolio optimization: Forecasting covariances and choosing the risk model. *The review of Financial studies*, *12*(5), 937-974.

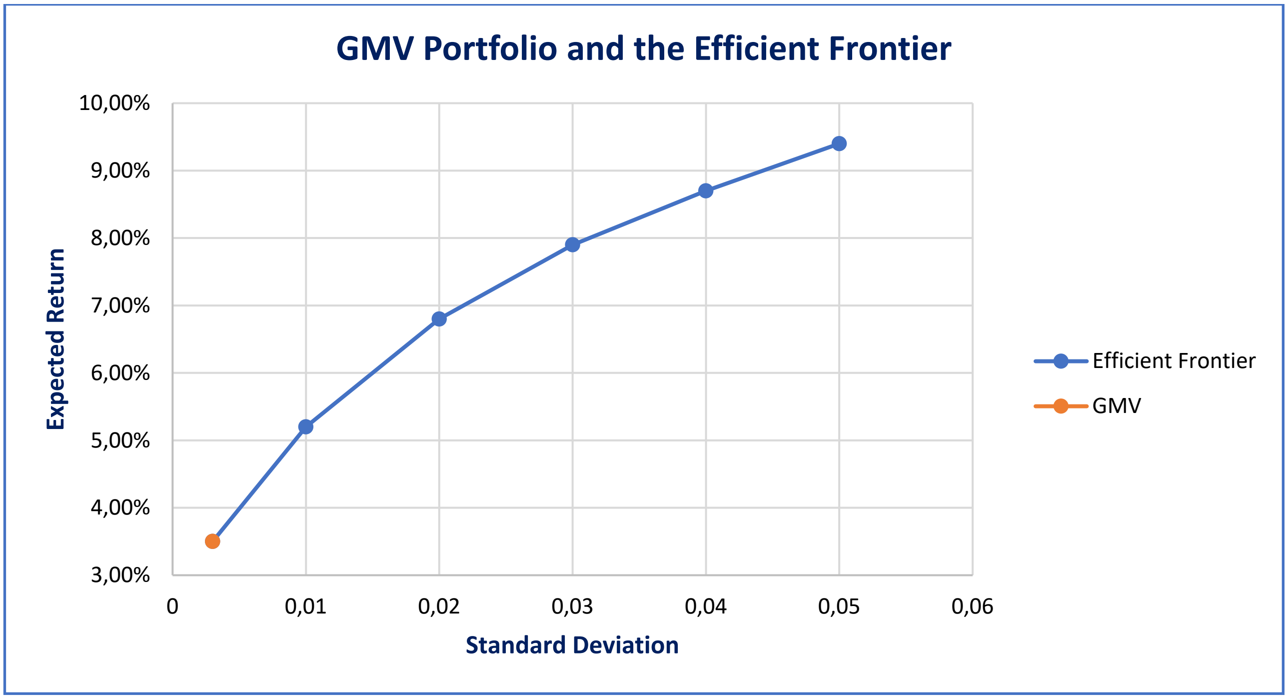


Figure 9, The GMV Portfolio on the Efficient Frontier. Data Source: A personal elaboration of Bloomberg Data, using Excel[44].

1.3.5 The Maximum Diversification portfolio (MDP)

The main rationale behind the Maximum Diversification Portfolio, as the name suggests, is to build a portfolio which incorporates the potential benefits of diversification through the maximization of the diversification ratio (DR) (31).

$$DR(X) = \frac{X^T r}{\sqrt{X^T \Omega X}} \quad (31)$$

Where r is the vector of asset volatilities:

[44] The data used to construct the graph is gathered from Bloomberg Database. It consists of index prices of 15 European Indices divided in Government Bonds, Corporate Bonds and Equities. The data consider a time-period of 5 years (2013-2018).

$$r = \begin{bmatrix} \sigma_k \\ \sigma_i \\ . \\ . \\ \sigma_N \end{bmatrix}$$

The diversification ratio can be understood as the weighted average of volatilities divided by the whole portfolio volatility which considers the correlation between the different securities. The ratio "*measures the gains from not having perfectly correlated assets*"[45].

The optimization problem for the MDP follows the same techniques used to evaluate the mean variance portfolio. In this case, it requires to minimize the denominator of equation (31) which indeed has embedded all the correlation coefficients of the underlying assets. Requiring the portfolio to invest in those assets that minimize the correlation coefficients is the same as maximizing the diversification of the portfolio.

The algorithm then becomes:

$$\begin{Bmatrix} X^*_{MDP} = \max f(X) \\ f(X) = \sum_{k=1}^{N} \frac{X_k \sigma_k}{\sigma} = \frac{X^T \sigma}{\sqrt{X^T \Omega X}} \end{Bmatrix} \quad (32)$$

Given the usual constraints:

$$1^T = 1$$

$$0 \leq X \leq 1$$

According to the empirical results, the Maximum Diversification Portfolios: "*have higher Sharpe ratios than the market cap-weighted indices and have had both lower volatilities and higher returns in the long run, which can be interpreted as capturing a bigger part of the risk premium*"[46].

[45] Choueifaty, Y., & Coignard, Y. (2008). Toward maximum diversification. *The Journal of Portfolio Management*, *35*(1), 40-51.
[46] Choueifaty et. Al., Op. cit., pp. 51.

1.3.6 The Maximum Sharpe Ratio portfolio (MSP)

The Sharpe ratio is a measure of performance introduced by William Sharpe in 1966. The ratio is often adopted by practitioners to evaluate fund and portfolio managers, representing the excess return over the risk-free rate per unit of volatility; therefore, it is an important tool in asset allocation.

The Maximum Sharpe Ratio Portfolio (MSR), also known as the tangency portfolio, finds an optimal capital allocation in the presence of a riskless asset[47]. It is a portfolio that lies on the efficient frontier where the line that goes from the point (0, risk-free rate) is tangent to the efficient frontier. The optimization problem is structured in the same way as the one for the MDP.

The method requires solving the following system of equations (33):

$$\begin{Bmatrix} X^*_{MDP} = \max(f(X)) \\ f(X) = \dfrac{\overline{R}}{\sigma} = \dfrac{X^T\overline{R}}{\sqrt{X^T\Omega X}} \end{Bmatrix} \quad (33)$$

With the constraints

$$1^T = 1$$

$$0 \leq X \leq 1$$

Where $\overline{R}$ is the vector of expected returns.

1.3.7 The Inverse Volatility portfolio (IV)

The Inverse Volatility Portfolio (IV), also called Naïve Risk Parity approach, is a portfolio allocation strategy which allocates to each component a weight equal to the inverse of their volatilities, measured by the standard deviation. This method is a very simple technique since it does not require any optimization process, nor solving any quadratic algorithm. The only

[47] Khokhlov, V. (2011). An Algorithm for Finding a Portfolio with the Highest Sharpe Ratio. *Available at SSRN 1767338*.

rule behind it is that safer assets (i.e. Government Bonds) will be given a higher weight in the portfolio compared with riskier securities (i.e. Equity). Weighting each asset inversely to its volatility will not produce very nice weights, therefore it is important to normalize them, rescaling all securities to sum up to one. Although this strategy can be erroneously mixed up with the risk parity approach, it is indeed quite different. That is because, being the portfolio volatility a non-additive measure, each security will present a different weight[48].

A main limitation of the IV portfolio, is that it does not consider the variance-covariance matrix, therefore eliminating all the diversification benefits produced by the correlation coefficients. Therefore, assets exhibiting a higher standard deviation are penalized no matter their pairwise correlations.

The optimal portfolio weights vector is then computed in the same way as the ERC method in the case of homogenous correlations assumption (34).

$$X_{IV}^{*} = \frac{\sigma_k^{-1}}{\sum_N 1/\sigma_k} \quad (34)$$

1.3.8 <u>Market Capitalization portfolio – Cap-weighted (MCWP)</u>

A Market Capitalization Portfolio (MCWP), also referred as Cap-weighted, is a portfolio which computes the weights as the average of the market capitalizations of the portfolio constituents over the sum of the average of the same capitalizations (35)[49].

$$X_{MCWP}^{*} = \left(\frac{M_1}{\sum_{k=1}^{N} M_k}, \ldots\ldots, \frac{M_N}{\sum_{k=1}^{N} M_k} \right) (35)$$

Where M_k is the market capitalization of the *kth* security.

The benefits of allocating a portfolio according to the market-caps of its components are multiples. Among all, these kinds of portfolios allot the greatest weights to the largest firms,

[48] For a deeper analysis see: https://investresolve.com/blog/portfolio-optimization-simple-optimal-methods/.

[49] Jain, P., & Jain, S. (2019). Can Machine Learning-Based Portfolios Outperform Traditional Risk-Based Portfolios? The Need to Account for Covariance Misspecification. *Risks*, *7*(3), 74.

which are also the most liquid ones, given the high correlation between market capitalization and liquidity.

Nonetheless, the economist Hsu, in his research paper "*Cap-Weighted Portfolios Are Sub-optimal Portfolios"* published in 2004, shows that Cap-weighted portfolios are empirically sub-optimal because: "*If stock prices are inefficient in the sense that they do not fully reflect firm fundamentals, then underpriced stocks will have smaller capitalizations than their fair equity value and similarly over-priced stocks will have larger capitalizations than their fair equity value. A cap-weighted portfolio would on average shift additional weights into the over-priced stocks and shift weights away from the underpriced stocks. As long as these pricing errors are not persistent, market prices will collapse toward fair value over time and a cap-weighted portfolio would tend to experience greater price decline than other non-price-weighted portfolios due to its heavier exposure to stocks with positive pricing error"*[50].

1.4 Portfolios Performance Evaluation Techniques

In this section I am going to briefly analyze some of the most used portfolio's performance evaluation tools which are a good indicator of the robustness of the portfolios. These measures are divided according to their nature. For what concerns a portfolio's return the most widely accepted techniques are: the Sharpe, Sortino and Treynor ratios. On the other hand, for a deeper risk analysis the Maximum Drawdown, Expected Shortfall and Value at Risk are taken under consideration.

1.4.1 Sharpe ratio

The Sharpe ratio or "Reward to Variability ratio" is one of the most known methods for evaluating the portfolio performance. It was developed in 1966 by William Sharpe to carry

[50] Hsu, J. C. (2004). Cap-weighted portfolios are sub-optimal portfolios. *Journal of investment Management*, *4*(3).

out an analysis regarding the performance of mutual fund managers[51]. This ratio helps investors understand how well their investments are compensated for the risk they have taken.

The excess return is defined as:

$$R_{ER} = R_p - r_f \ (36)$$

Where R_p is the return of the portfolio and r_f is the risk-free rate. The geometric average is instead used when investors re-invest in all periods (37)[52].

$$R_{G.ER} = \left(\prod_{t=1}^{T}(1 + R_{ER,t})^{\frac{1}{T}}\right) - 1 \ (37)$$

By using the measure σ_p defined in section 1.1, the Sharpe Ratio becomes:

$$Sharpe\ Ratio\ (SR) = \frac{R_{G.ER}}{\sigma_p} \ (38)$$

Therefore, the Reward to Variability Ratio measures the extra return an investor earns per unit of increase in volatility. A higher portfolio ratio means a better risk-adjusted performance; hence it is a fundamental tool to critically analyze your investment decisions. However, the ratio is based on some important assumptions which may limit its reliability as a good risk-performance indicator. For instance, standard deviation (volatility) is used as a proxy for risk even if returns in the financial markets have proven to be skewed away from the average because of many sudden and large price moves -either up or down[53].

1.4.2 <u>Sortino ratio</u>

Another good performance metric when investigating the portfolio robustness is the Sortino ratio, which was developed by the researchers Sortino and Price in 1994[54]. The indicator,

---

[51] Goetzmann, W., Ingersoll, J., Spiegel, M. I., & Welch, I. (2002). Sharpening sharpe ratios. *National bureau of economic research.*

[52] Sharpe, W. F. (1994). The sharpe ratio. *Journal of portfolio management*, *21*(1), 49-58.

[53] For a deeper analysis see: https://www.investopedia.com/terms/s/sharperatio.asp.

[54] Sortino, F., & Price, L.N. (1994). Performance Measurement in a Downside Risk Framework. *Journal of Investing,* 3(3), 50-58.

unlike the Sharpe ratio, considers only the downside deviation from the mean as a measure of risk, therefore correcting for the limitations presented by the standard deviation which instead, treats in the same way both negative and positive dispersions from the mean.

Within the computation of the Sortino ratio, the applied volatility expressing the portfolio risk in the denominator of the Sharpe ratio, is substituted by the downside deviation $\mathbb{D}$ (39)[55].

$$\mathrm{D} = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \min(0, R_{k,t} - \overline{R_{k,t}})} \ (39)$$

Therefore, the Sortino ratio is computed as:

$$Sortino\ Ratio = \frac{R_{G.ER}}{\mathrm{D}} \ (40)$$

1.4.3 <u>Treynor ratio</u>

Developed by Jack Treynor in 1973, this ratio makes use of the CAPM theory for the estimation of the risk factor. Indeed, the standard deviation of the Sharpe ratio is replaced by the Beta coefficient of the Security Market Line (SML), which measures the change in the portfolio returns for unexpected market return movements. Although it presents similarities with the Sharpe ratio, its accuracy is influenced by the choice of the applicable benchmark to measure Beta. Furthermore, being a backward-looking performance measure, it tells you nothing about the future investment's performance[56].

By using the Beta coefficient expressed in formula (18) as an indicator of risk, the Treynor Ratio becomes:

$$Treynor\ Ratio = \frac{R_{G.ER}}{\beta} \ (41)$$

[55] Füss, R., Kaiser, D. G., & Schindler, F. (2007). Dynamic linkages between hedge funds and traditional financial assets: Evidence from emerging markets.
[56] For a deeper analysis see: https://www.investopedia.com/terms/t/treynorratio.asp.

1.4.4 Value at Risk and the expected shortfall

The Value at Risk (VaR) is a widely used risk measure for losses in a portfolio of assets. It answers to the question "How much can I expect to lose tomorrow or over the next time period, and with which probability?". Therefore, Value at Risk is defined as the maximum potential loss that a portfolio/position can suffer given a specified time horizon and a specified confidence level[57].

There are different methods used to compute this risk measure, which mainly divide into: *Parametric Approach, Montecarlo Simulation and Historical Simulation.* The most common approach requires the application of a closed form formula (42) and assumes the distribution of risk is a normal one[58].

$$VaR_i = M_i \cdot \sigma_i \cdot \sqrt{t} \cdot \alpha \cdot \Delta \ (42)$$

where:

- M_i : the market value of the position
- σ (sigma): the annual volatility of the underlying risk factor
- $\sqrt{t}$: the time horizon expressed in year fraction
- α : the value of the normal cumulative function, given a probability level equal to the VaR confidence level
- Δ (delta): the sensitivity of Market Value to changes in value of the underlying risk factor

The VaR of a portfolio constituted by two securities *k* and *i*, VaR_P, is instead computed as:

$$VaR_p = \sqrt{VaR_i^2 + VaR_k^2 + 2\rho_{i,k} VaR_i VaR_k} \ (43)$$

[57] Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of risk, 2,* 21-42.
[58] Sironi, A., & Resti, A. (2007). Risk management and shareholders' value in banking: from risk measurement models to capital allocation policie. *John Wiley & Sons,*(417).

In matrix terms:

$$VaR_p = \sqrt{V * C * V^T} \quad (44)$$

Where:

- V: row vector of VaRs of each individual security
- C: correlation matrix
- V^T: transpose of matrix V

The value 100P%VaR$_p$ represents the threshold loss value, such that the probability that the loss on the portfolio over the given time horizon exceeds this value, is P.

However, one of the main shortcomings of the Value at Risk formula is that it only gives the minimum loss for a given probability, but losses can be much larger than the tail; VaR doesn't tell you how much money to put in the portfolio to offset the loss. The Expected Shortfall (ES) or Conditional Value at Risk on the other hand, is a risk measure which explains what happens beyond the point of failure. It answers the question: "If things get bad, what is the total expected loss?", therefore explaining what the average loss in the tail below a would be given confidence level.

The Expected Shortfall is defined as the expected value of all losses excess of VaR and it is defined according to equation (45)[59].

$$ES = \mathbb{E}[-(\Delta\mathrm{M} - \mathbb{E}(\Delta\mathrm{M})] \,\big|\, (\Delta\mathrm{M} - \mathbb{E}(\Delta\mathrm{M})) > VaR \quad (45)$$

<u>Value at Risk (VaR) of a stock portfolio using Python: a practical application.</u>

In order to provide a deeper analysis about the VaR application, I have conducted a simple practical exercise by using the Python programming language[60]. All the codes can be found at the end of the thesis, in the section "Python Code".

[59] Sironi, A., & Resti, A., Op. cit., pp. 268-272.
[60] The whole exercise is conducted using the Anaconda Jupiter Notebook, Python version 3.7.

In the following example, I will calculate the Value at Risk of a portfolio composed by the two American stocks in the technology industry: Facebook [‘FB’] and Apple [‘AAPL’][61]. The chosen time horizon goes from the 01/01/2018 until the 31/12/2019.

The first step for computing the Var of our portfolio is to extract the periodic returns of the stocks in the portfolio for the given time horizon and computing the variance-covariance matrix (*Figure 10 and Figure 11).* All the stocks closing prices have been downloaded using the Yahoo-Finance data through the following Python function.

```
#---------------------------------------------------------------------------------------------------------
----------------------
tickers = ['AAPL','FB']
start_date='2018-01-01'
end_date='2019-12-31'
data = pdr.get_data_yahoo (tickers, start=start_date, end=end_date)['Close']
returns = data.pct_change()
```

Snippet 1- ‘AAPL & FB’ Stocks portfolio Returns.

Symbols	**AAPL**	**FB**
Date		
2019-12-24	0,000951	-0,005141
2019-12-26	0,019840	0,013017
2019-12-27	-0,000379	0,001492
2019-12-30	0,005935	-0,017732
2019-12-31	0,007307	0,004109

Figure 10, Stocks Periodic Returns tail function, Data Source: Yahoo-Finance on Python.

[61] The methodology for the computation of VaR follows the one reported in this sample exercise: https://www.interviewqs.com/blog/value_at_risk.

Symbols	AAPL	FB
AAPL	0,000301	0,000166
FB	0,000166	0,000442

Figure 11, Variance-Covariance Matrix. Data Source: Yahoo-Finance on Python.

The next step requires the computation of the portfolio mean and standard deviation. I have arbitrarily chosen to equally distribute the investment weights (therefore 50% on each stock). I have also assumed that the initial investment in the portfolio is 1M USD. Then, we have to calculate the inverse of the normal cumulative distribution (represented by the "Inverse" function in the Python code), a specified confidence interval, standard deviation and mean. The total VaR will be given by subtracting the initial investment from the results obtained with the *Inverse* function *(Snippet 2).*

From the below function "*var*_1d1" we can understand what is the maximum loss that our portfolio can bear over a one-day period. If we solve it, we get a VaR of 37300,79 USD which means that with 99% confidence our portfolio of 1M USD will not experience losses greater than that value a day-time horizon.

```
#-----------------------------------------------------------------------------------------------------------------------------

# Calculate mean returns for each stock
avg_rets = returns.mean()

# Calculate mean returns for portfolio overall,
# using dot product to normalize individual means against investment weights
port_mean = avg_rets.dot(weights)

# Calculate portfolio standard deviation
port_stdev = np.sqrt(weights.T.dot(cov_matrix).dot(weights))

# Calculate mean of investment
mean_investment = (1+port_mean) * initial_investment

# Calculate standard deviation of investment
stdev_investment = initial_investment * port_stdev
#-----------------------------------------------------------------------------------------------------------
# Select our confidence interval (I'll choose 99% here)
conf_level1 = 0.01
#-----------------------------------------------------------------------------------------------------------
# Using SciPy ppf method to generate values for the inverse cumulative distribution function to a normal distribution
```

```
# Plugging in the mean, standard deviation of our portfolio as calculated above
from scipy.stats import norm
Inverse = norm.ppf(conf_level1, mean_investment, stdev_investment)
#---------------------------------------------------------------------------------------------------------
#Finally, we can calculate the VaR at our confidence interval
var_1d1 = initial_investment - Inverse
var_1d1
#---------------------------------------------------------------------------------------------------------
# Calculate n Day VaR
var_array = []
num_days = int(15)
for x in range(1, num_days+1):
    var_array.append(np.round(var_1d1 * np.sqrt(x),2))
    print(str(x) + " day VaR @ 95% confidence: " + str(np.round(var_1d1 * np.sqrt(x),2))
```

Snippet 2, Portfolio Value at Risk (VaR) computation. Data Source: Yahoo-Finance on Python.

The last function of the *Snippet 2* allows to compute the VaR of the portfolio for the first 15 days. This can be easily executed by the input *"var_array.append(np.round(var_1d1 * np.sqrt(x),2))"*, which multiplies the 1-day VaR by the square root of the time period. *Figure 13* reported below, plots the maximum portfolio loss which increases with the number of days.

N Day VaR Results:

1-day VaR @ 99% confidence: 37300,79

2-day VaR @ 99% confidence: 52751,28

3-day VaR @ 99% confidence: 64606,86

4-day VaR @ 99% confidence: 74601,58

5-day VaR @ 99% confidence: 83407,1

6-day VaR @ 99% confidence: 91367,9

7-day VaR @ 99% confidence: 98688,61

8-day VaR @ 99% confidence: 105502,57

9-day VaR @ 99% confidence: 111902,37

10-day VaR @ 99% confidence: 117955,45

11-day VaR @ 99% confidence: 123712,72

12-day VaR @ 99% confidence: 129213,73

13-day VaR @ 99% confidence: 134489,91

14-day VaR @ 99% confidence: 139566,78

15- day VaR @ 99% confidence: 144465,34

Figure 12, N Day VaR Results. Data Source: Yahoo-Finance on Python.

In order to check whether the VaR is a reliable measure to assess the risk trend of our portfolio, a further check requires to verify how well the historical returns of the two stocks in the portfolio have been distributed compared with the normal distribution[62]. From *Figure 14* and *Figure 15*, we can state that the returns of our chosen stocks have been normally distributed.

[62] One of the main assumptions of the parametical VaR application is that returns in the portfolio are normally distributed.

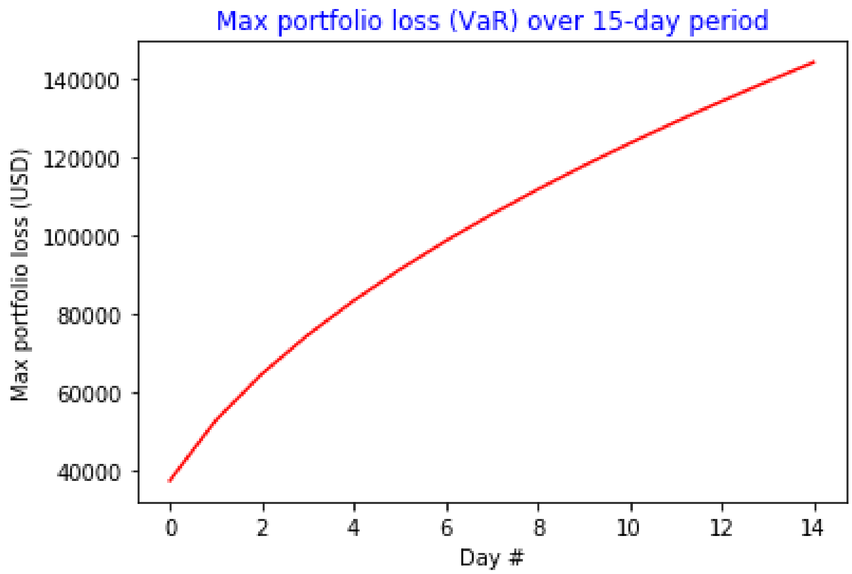


Figure 13, Value at Risk for the first 15 days. Data Source: Yahoo-Finance on Python.

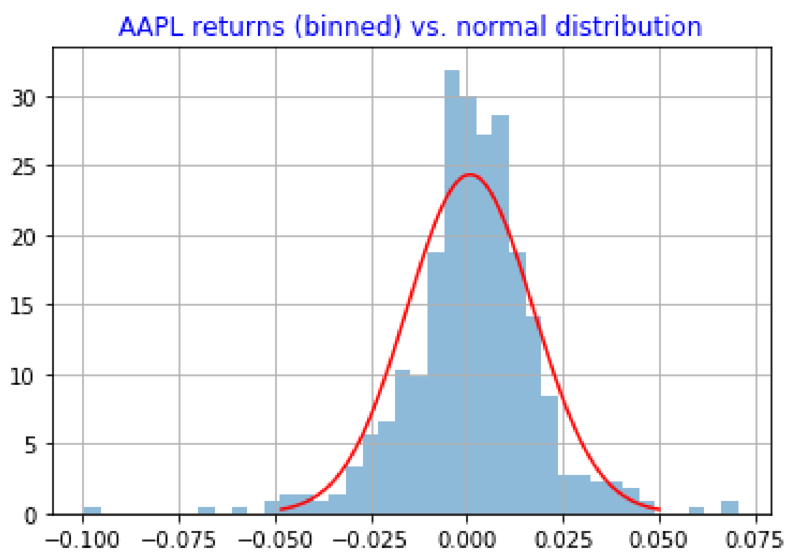


Figure 14, Apple Returns vs normal distribution. Data Source: A personal elaboration on Python.

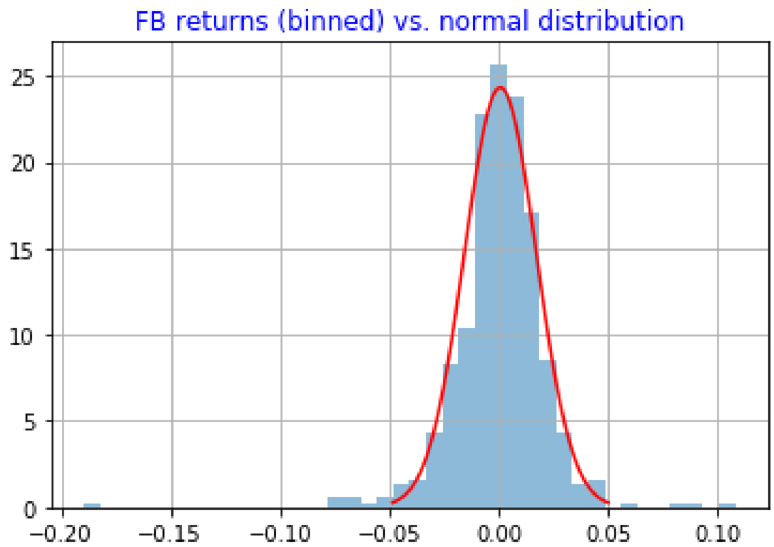


Figure 15, Facebook Returns vs normal distribution. Data Source: A personal elaboration on Python.

1.4.5 <u>The Maximum Drawdown</u>

The Maximum Drawdown (MDD) is a portfolio risk measure that considers the maximum negative trend of the stocks returns over a specified period. Indeed, it is the value observed from a peak to a trough point in a portfolio graph as shown by *Figure 16*. The Maximum Drawdown tells investors how big the loss in the portfolio is during the worst possible scenario[63]. Nonetheless, this risk measure does not provide any information regarding the frequency of the losses and how much time do investors need to recover from them. It only takes into account the largest negative value, but what about the smaller but yet considerable portfolio's losses between the peak and trough period?

$$MDD_t = \frac{P - \max(P_r)_{0<r<t}}{\max(P_r)_{0<r<t}} \quad (46)$$

[63] Magdon-Ismail, M., & Atiya, A. F. (2004). Maximum drawdown. *Risk Magazine*, *17*(10), 99-102.

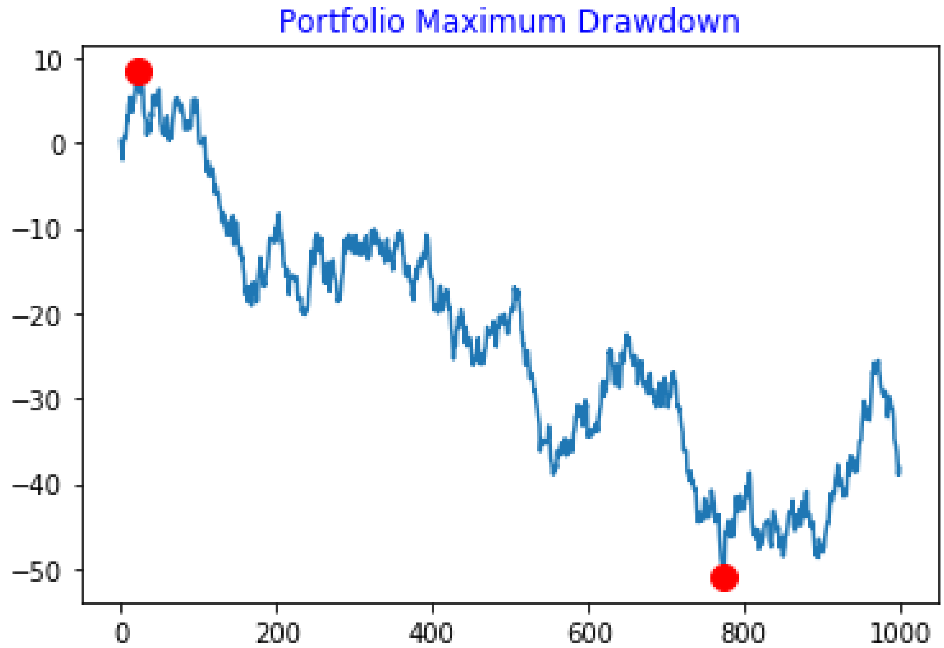


Figure 16, The Maximum Drawdown of a Portfolio. Data Source: A personal elaboration on Python.

1.5 Efficient Frontier Optimization in Python: Maximum Sharpe Ratio vs Minimum Volatility

This section of the thesis offers a practical application for the efficient frontier optimization problem comparing two different kinds of portfolios in terms of risk and performance: The Maximus Sharpe Ratio portfolio (MSP) and the Minimum Variance portfolio (MVP)[64]. All the data is downloaded from the Quandl library of the Python computer programming language[65].

The portfolio is constituted by 4 different stocks I have chosen from the technology sector: Apple ('AAPL'), Amazon ('AMZN'), Google ('GOOGL') and Facebook ('FB'). The selected time horizon for this analysis is purely arbitral and goes from: 01-01-2016 / 31/12/2017.

Figure 17 and *Figure 18* show the trend of the price of each stock during the chosen time period. The daily stock returns graph tries to embed the volatility of daily returns. From *Figure 18* it

[64] The exercise is based on the following practical application: https://towardsdatascience.com/efficient-frontier-portfolio-optimisation-in-python-e7844051e7f.

[65] The Quandl library is a free financial platform that provides all the updated closing prices for each single stock.

looks like Google is the more stable stock, exhibiting a constant trend, while Amazon appears the riskiest one due to several positive and negative spikes.

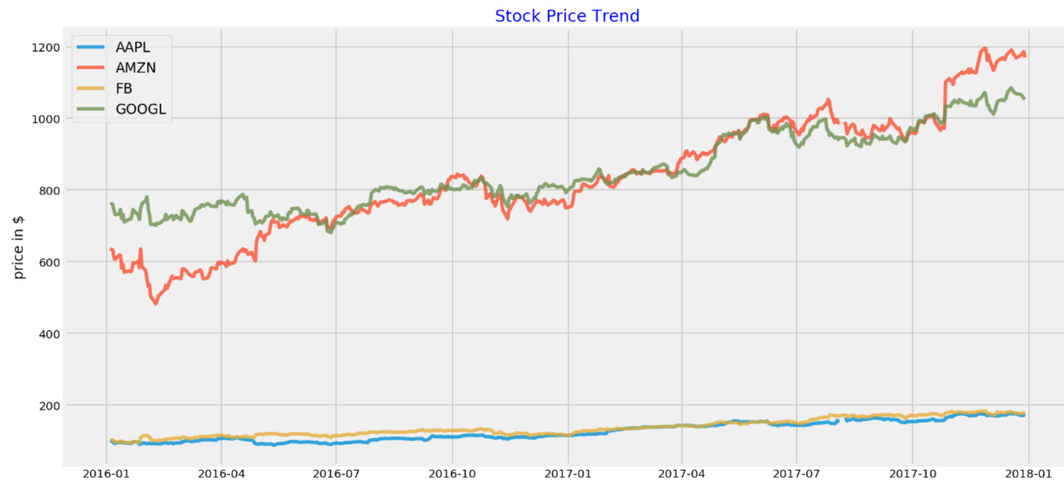


Figure 17, Annual Stock Price Trend. Data Source: Quandl on Python.

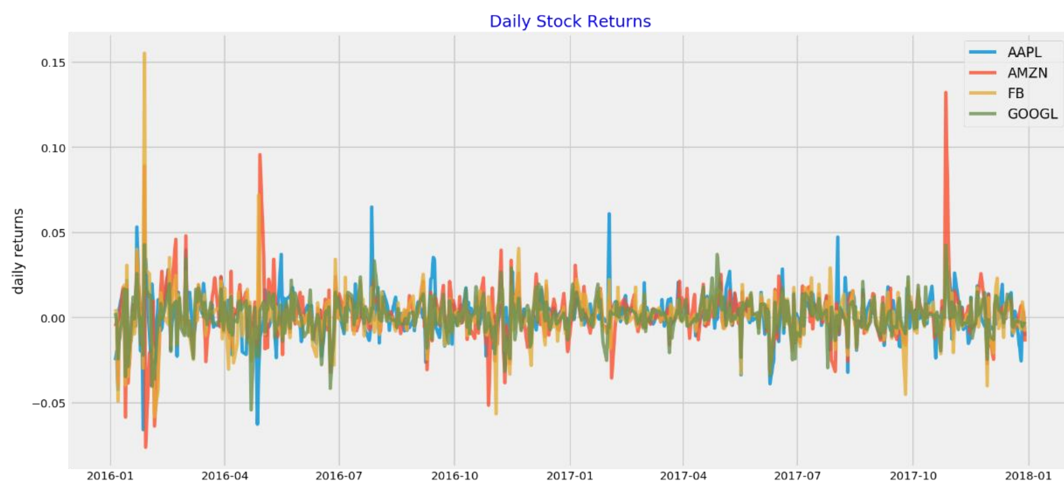


Figure 18, Daily Stock Returns Trend. Data Source: Quandl on Python.

The next important step is to allocate our initial investment over the 4 stocks in our portfolio. If we assume our investment is equal to 1M USD, we have to decide how much of that investment to allocate to each stock, thus selecting the portfolio's constituents' weights. Therefore, I am going to define a function that randomly generates the portfolio's weights.

The *portfolio_annualized_performance* function in the *Snippet 3*, computes the annualized covariance matrix, the returns and standard deviation of our randomly generated portfolios. The function *random_portfolios* instead, creates the number of portfolios we want to create, assigning random weights to each security in the portfolio. The analysis is carried out considering 25000 possible portfolios and a risk-free rate of 1,78%[66].

```
#----------------------------------------------------------------------------------------------------
#Defining the returns and standard deviation of our portfolio
def portfolio_annualised_performance(weights, mean_returns, cov_matrix):
    returns = np.sum(mean_returns*weights ) *252
    std = np.sqrt(np.dot(weights.T, np.dot(cov_matrix, weights))) * np.sqrt(252)
    return std, returns
#----------------------------------------------------------------------------------------------------
#Randomly generating portfolios that assigns weights to each stock
Def random_portfolios (num_portfolios, mean_returns, cov_matrix,
risk_free_rate,var_1d1):
    results = np.zeros((3,num_portfolios))
    weights_record = []
    for i in range(num_portfolios):
        weights = np.random.random(4)
        weights /= np.sum(weights)
        weights_record.append(weights)
        portfolio_std_dev, portfolio_return = portfolio_annualised_performance(weights,
mean_returns, cov_matrix)
        results[0,i] = portfolio_std_dev
        results[1,i] = portfolio_return
        results[2,i] = (portfolio_return - risk_free_rate) / portfolio_std_dev
    return results, weights_record
```

[66] The risk-free rate chosen is the one of 01/01/2018 published in the U.S. Department of The Treasury.

```
#-----------------------------------------------------------------------------------------------------
---------------------#Assigning data for the portfolios
returns = table.pct_change()
mean_returns = returns.mean()
cov_matrix = returns.cov()
num_portfolios = 25000
risk_free_rate = 0.0178
initial_investment = 1000000
conf_level1 = 0.01
```

Snippet 3, Functions to generate random portfolios, assign weights and ask for the desired outputs. Data Source: A personal elaboration on Python.

For the Value at Risk measure I have used the same function already applied in section 1.4.4 (*Snippet 2)*. The confidence level is 99% for the calculation of the VaR which assumes returns are normally distributed (function: *norm.ppf(con_level1*, *mean_investment1, std_investment*).

The goal of the exercise is then to create a function that randomly generates portfolios and gets the outcomes (mean return, volatility, Sharpe Ratio and Value at Risk). The portfolios I am interested into are the Global Minimum Variance portfolio (GMV) and the Maximum Sharpe Ratio portfolio (MSR). The function *display_simulated_ef_with_random(mean_returns, cov_matrix, num_portfolios, risk_free_rate,var_1d1)* in *Snippet 4* compares the GMV and the MSR portfolios composed by the 4 selected stocks, in terms of risk and performance metrics as well as weight allocation. Moreover, it plots them on the efficient frontier line to investigate the differences with the Markowitz mean-variance approach. The GMV portfolio is marked with a green star on the efficient frontier, while the MSR portfolio with a purple star *(Figure 19)*. The blue colored dots form a shape of an arch line which represents the efficient frontier. Therefore, that shape is made of the 25000 randomly generated portfolios, with each dot representing a single portfolio.

```
#-----------------------------------------------------------------------------------------------------------
#Defining the Maximum Sharpe Ratio Portoflio
    max_sharpe_idx = np.argmax(results[2])
    sdp, rp = results[0,max_sharpe_idx], results[1,max_sharpe_idx]
    mean_investment=initial_investment*(1+rp)
    sdp_investment=initial_investment*sdp
    Inverse_S = norm.ppf(conf_level1, mean_investment, sdp_investment)
    var_1d1_S = initial_investment - Inverse_S
    max_sharpe_allocation = 
pd.DataFrame(weights[max_sharpe_idx],index=table.columns,columns=['allocation'])
    max_sharpe_allocation.allocation = [round(i*100,2)for i in 
max_sharpe_allocation.allocation]
    max_sharpe_allocation = max_sharpe_allocation.T
#-----------------------------------------------------------------------------------------------------------
 #Defining the Minimum Volatility portfolio
    min_vol_idx = np.argmin(results[0])
    sdp_min, rp_min = results[0,min_vol_idx], results[1,min_vol_idx]
    mean_min_investment=initial_investment*(1+rp_min)
    sdp_min_investment=initial_investment*sdp_min
    Inverse_MV = norm.ppf(conf_level1, mean_min_investment, sdp_min_investment)
    var_1d1_MV = initial_investment - Inverse_MV
    min_vol_allocation = 
pd.DataFrame(weights[min_vol_idx],index=table.columns,columns=['allocation'])
    min_vol_allocation.allocation = [round(i*100,2)for i in min_vol_allocation.allocation]
    min_vol_allocation = min_vol_allocation.T
```

```
#-----------------------------------------------------------------------------------------------------
#Getting the desired outcomes
    print("-"*80)
    print("Maximum Sharpe Ratio Portfolio Allocation\n")
    print("Annualised Return:", round(rp,2))
    print("Annualised Volatility:", round(sdp,2))
    print ("Value at Risk:",round(var_1d1_S,2))
    print("\n")
    print(max_sharpe_allocation)
    print("-"*80)
    print("Minimum Volatility Portfolio Allocation\n")
    print("Annualised Return:", round(rp_min,2))
    print("Annualised Volatility:", round(sdp_min,2))
    print ("Value at Risk:",round(var_1d1_MV,2))
    print(min_vol_allocation)
#-----------------------------------------------------------------------------------------------------
#Plotting the results
    plt.scatter(results[0,:],results[1,:],c=results[2,:],cmap='YlGnBu', marker='o', s=10,
alpha=0.3)
    plt.scatter(sdp,rp,marker='*',color='green',s=500, label='Maximum Sharpe ratio')
    plt.scatter(sdp_min,rp_min,marker='*',color='purple',s=500, label='Minimum
volatility')
    plt.title('Simulated Portfolio Optimization based on Efficient Frontier',color='blue')
    plt.xlabel('annualised volatility')
    plt.ylabel('annualised returns')
    plt.legend(labelspacing=0.8)
```

Snippet 4, Maximum Sharpe Ratio Portfolio and Minimum Volatility Portfolio on the Efficient frontier. Data Source: A personal elaboration on Python.

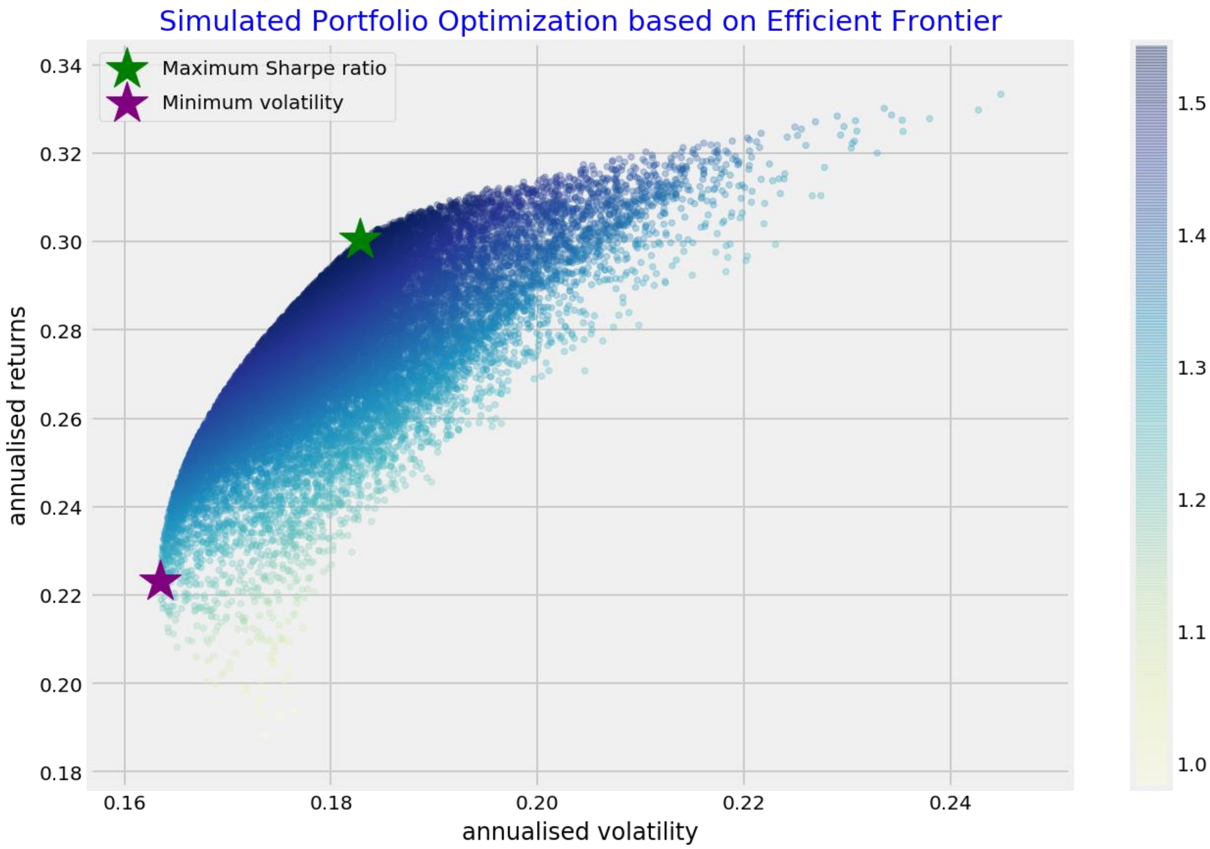


Figure 19, Simulated Portfolio Optimization based on Efficient Frontier. Data Source: A personal elaboration on Python.

If we look at the results reported below *(Figure 20),* we see that more than half (58,52%) of our initial investment is allocated to the Google stock for the Minimum Variance Portfolio. This makes sense; indeed, Google was the less risky stock among the 4. On the other hand, if we are more risk seeking, we would prefer the Maximum Sharpe Ratio portfolio, which allocates most of the total investment among the riskier securities and almost 0 (0,01%) to Google. For this reason, the latter portfolio exhibits a higher annualized return (0,3) and a higher volatility (0,18). Given that the 1-day Value at Risk on a 1M USD investment is higher for the MV portfolio, I would rather choose the Maximum Sharpe Ratio portfolio. A more detailed analysis regarding the maximum potential loss (VaR) in the portfolio over the first 15 days, is provided in *Figures 21-24.*

```
------------------------------------------------------------------------
------------
Maximum Sharpe Ratio Portfolio Allocation

Annualised Return: 0,3
Annualised Volatility: 0,18
1-day Value at Risk: 125013,25


AAPL    AMZN    FB      GOOGL
Allocation:   46,62   27,99   25,38    0,01
------------------------------------------------------------------------
------------
Minimum Volatility Portfolio Allocation

Annualised Return: 0,22
Annualised Volatility: 0,16
1-day Value at Risk: 157282,7


AAPL    AMZN    FB    GOOGL
Allocation:   33,66   1,23   6,59   58,52
```

Figure 20, MSR Portfolio and GMV Portfolio % outputs. Data Source: A personal elaboration on Python.

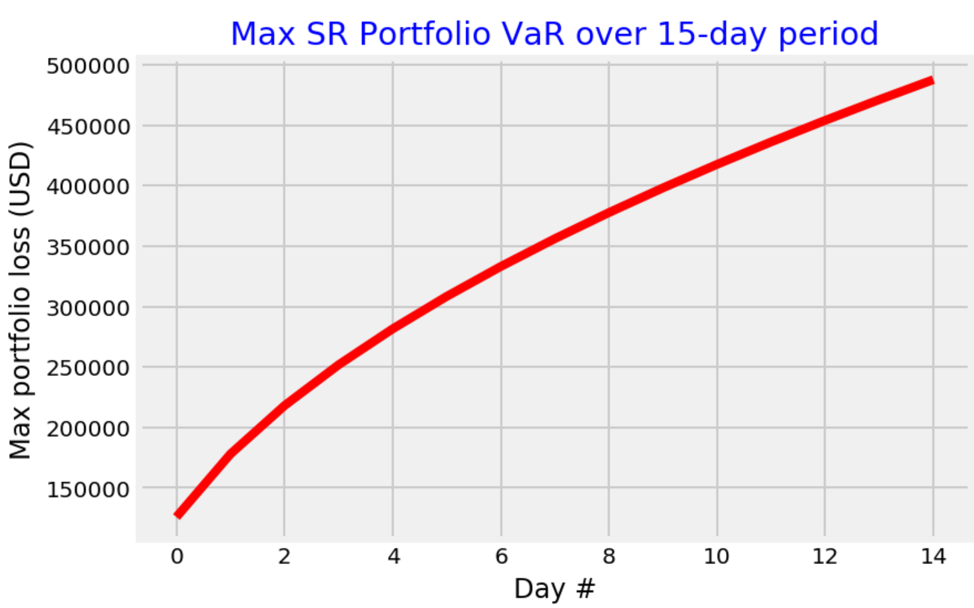


Figure 21, Value at Risk for the first 15 days MSR Portfolio. Data Source: A personal elaboration on Python.

<u>N Day VaR Results MSR:</u>

1-day VaR @ 99% confidence: 125898,87

2-day VaR @ 99% confidence: 178047,89

3- day VaR @ 99% confidence: 218063,23

4 -day VaR @ 99% confidence: 251797,73

5- day VaR @ 99% confidence: 281518,42

6 -day VaR @ 99% confidence: 308387,98

7 -day VaR @ 99% confidence: 333097,09

8 -day VaR @ 99% confidence: 356095,77

9- day VaR @ 99% confidence: 377696,6

10 -day VaR @ 99% confidence: 398127,17

11-day VaR @ 99% confidence: 417559,3

12- day VaR @ 99% confidence: 436126,47

13-day VaR @ 99% confidence: 453934,82

14- day VaR @ 99% confidence: 471070,43

15- day VaR @ 99% confidence: 487604,22

Figure 22, MSR Portfolio N-Day VaR results. Data Source: A personal elaboration on Python.

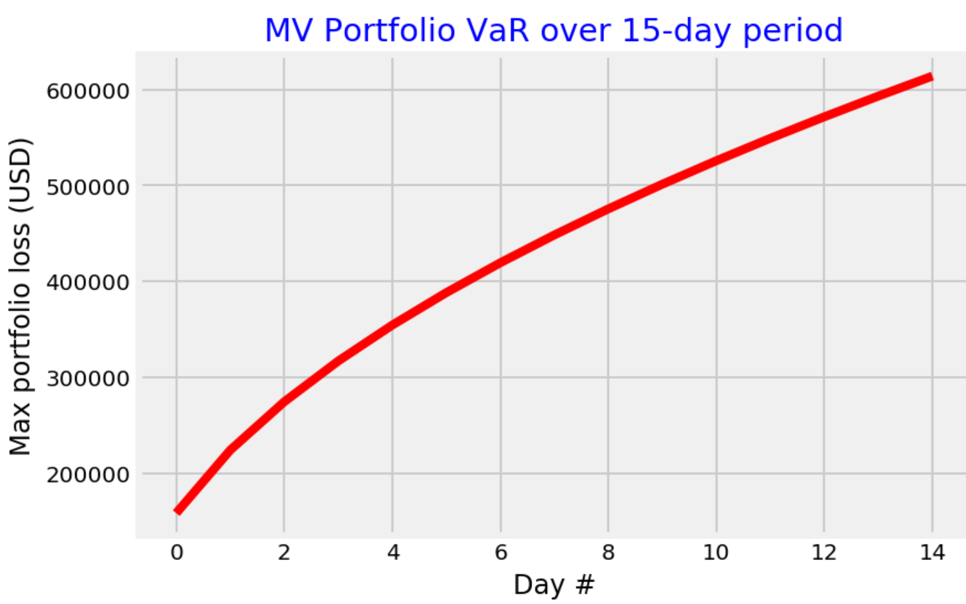


Figure 23, Value at Risk for the first 15 days MV Portfolio. Data Source: A personal elaboration on Python.

N Day VaR Results GMV:

1- day VaR @ 99% confidence: 158518,61

2 -day VaR @ 99% confidence: 224179,17

3 -day VaR @ 99% confidence: 274562,29

4 -day VaR @ 99% confidence: 317037,22

5 -day VaR @ 99% confidence: 354458,39

6 -day VaR @ 99% confidence: 388289,71

7 -day VaR @ 99% confidence: 419400,82

8 -day VaR @ 99% confidence: 448358,33

9 -day VaR @ 99% confidence: 475555,83

10 -day VaR @ 99% confidence: 501279,86

11 -day VaR @ 99% confidence: 525746,75

12 -day VaR @ 99% confidence: 549124,57

13 -day VaR @ 99% confidence: 571546,97

14 -day VaR @ 99% confidence: 593122,33

15 -day VaR @ 99% confidence: 613939,93

Figure 24, GMV Portfolio N-Day VaR results. Data Source: A personal elaboration on Python.

In the next analysis, I would like to show where the single four stocks lie on the efficient frontier curve. We can then compare their return and volatility with the ones of the Maximum Sharpe Ratio and Global Minimum Variance portfolios (*Figure 25)*. In this way, it is possible to observe the effect of diversification, and how this leads to more robust portfolios. *Snippet 5* provides the new Python code.

```
#-----------------------------------------------------------------------------------------------------------
----------------------
#Defining the desired function
def display_ef_with_selected(mean_returns, cov_matrix, risk_free_rate):
    max_sharpe = max_sharpe_ratio(mean_returns, cov_matrix, risk_free_rate)
    sdp, rp = portfolio_annualised_performance(max_sharpe['x'], mean_returns, cov_matrix)
#-----------------------------------------------------------------------------------------------------------
----------------------
#Defining the Maximum Sharpe Ratio Portoflio
    max_sharpe_allocation =
pd.DataFrame(max_sharpe.x,index=table.columns,columns=['allocation'])
    max_sharpe_allocation.allocation = [round(i*100,2)for i in max_sharpe_allocation.allocation]
    max_sharpe_allocation = max_sharpe_allocation.T
#-----------------------------------------------------------------------------------------------------------
  #Defining the Minimum Volatility portfolio
  min_vol = min_variance(mean_returns, cov_matrix)
    sdp_min, rp_min = portfolio_annualised_performance(min_vol['x'], mean_returns,
cov_matrix)
    min_vol_allocation =
pd.DataFrame(min_vol.x,index=table.columns,columns=['allocation'])
    min_vol_allocation.allocation = [round(i*100,2)for i in min_vol_allocation.allocation]
    min_vol_allocation = min_vol_allocation.T
    an_vol = np.std(returns) * np.sqrt(252)
    an_rt = mean_returns * 252
```

```
#--------------------------------------------------------------------------------------------------------------------------

#Getting the desired outcomes
    print("-"*80)
    print("Maximum Sharpe Ratio Portfolio Allocation\n")
    print("Annualised Return:", round(rp,2))
    print("Annualised Volatility:", round(sdp,2) )
    print("\n")
    print(max_sharpe_allocation)
    print("-"*80)
    print("Minimum Volatility Portfolio Allocation\n")
    print("Annualised Return:", round(rp_min,2))
    print("Annualised Volatility:", round(sdp_min,2))
    print("\n")
    print(min_vol_allocation)
    print("-"*80)
    print("Individual Stock Returns and Volatility\n")
    for i, txt in enumerate(table.columns):
        print(txt,":","annuaised return",round(an_rt[i],2),", annualised volatility:",round(an_vol[i],2))
        print("-"*80)
```

Snippet 5, Portfolio diversification effect. MSR & GMV portfolios on the efficient frontier. Data Source: A personal elaboration on Python.

```
Maximum Sharpe Ratio Portfolio Allocation

Annualised Return: 0,3
Annualised Volatility: 0,18


             AAPL   AMZN    FB     GOOGL
allocation  44,67  29,05  26,28    0,0
----------------------------------------------------------------------
------
Minimum Volatility Portfolio Allocation

Annualised Return: 0,22
Annualised Volatility: 0,16


             AAPL   AMZN    FB   GOOGL
allocation  34,02  0,73  6,98   58,26
----------------------------------------------------------------------
-----------
Individual Stock Returns and Volatility

AAPL : annuaised return 0,28 , annualised volatility: 0,21
----------------------------------------------------------------------
-----------
AMZN : annuaised return 0,34 , annualised volatility: 0,25
----------------------------------------------------------------------
-----------
FB : annuaised return 0,3 , annualised volatility: 0,23
----------------------------------------------------------------------
-----------
GOOGL : annuaised return 0,18 , annualised volatility: 0,18
----------------------------------------------------------------------
------
```

Figure 25, MSR Portfolio and GMV Portfolio % outputs-The power of diversification. Data Source: A personal elaboration on Python.

From *Figure 25* we observe that the safer stock in terms of volatility is Google, with a value of 0,18. For the same reason, it also exhibits the lowest return. However, with the Global Minimum Variance portfolio we can achieve a lower level of risk (0,16) and a higher annualized return (0,22). Furthermore, if we are more risk-seeking the Maximum Sharpe Ratio portfolio gives us a much higher annualized return (0,3) with the same annualized volatility of the safer stock (0,18). By looking at *Figure 26* in the next page, we observe that none of the 100% position in the single stocks is optimal when compared with our optimized portfolios (MSR and GMV). Diversification is one of the main reasons behind this important result. Indeed, by spreading our initial investment more evenly across the securities we can achieve a lower level of risk and a higher return.

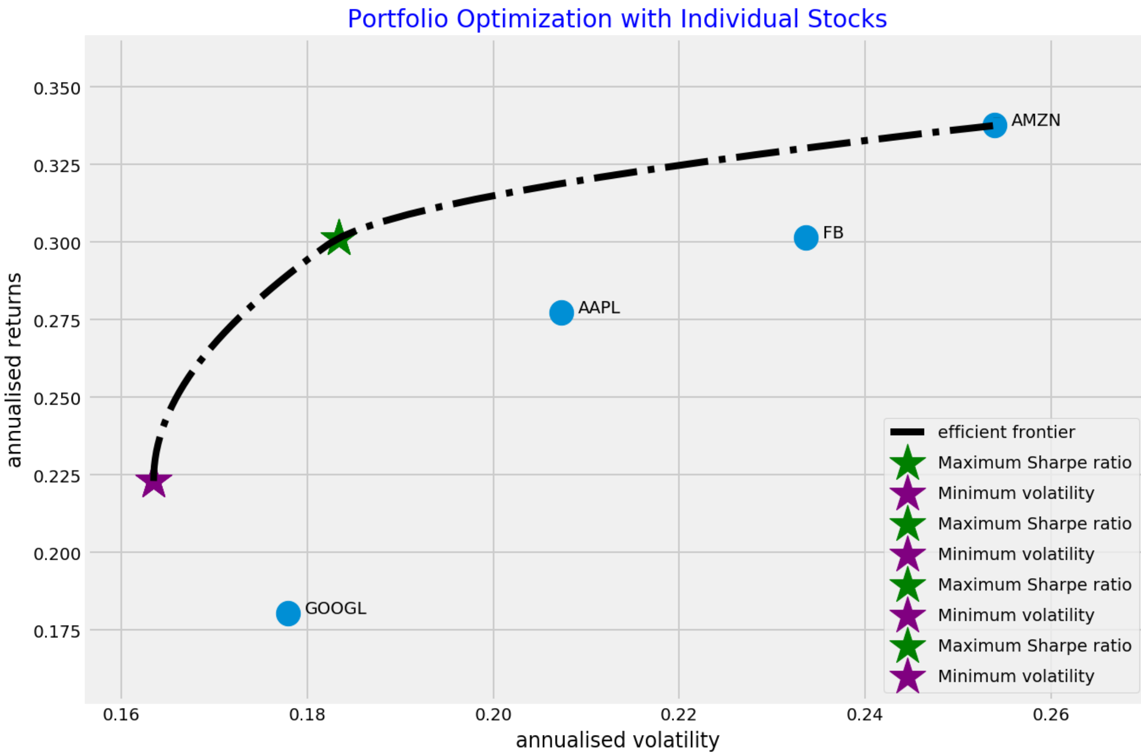


Figure 26, Portfolio Optimization with Individual Stocks. Data Source: A personal elaboration on Python.

2. THE HIERARCHICAL RISK PARITY PORTFOLIO OPTIMIZATION

2.1 The Hierarchical Structure

The hierarchical structure of complex financial systems was first investigated by the Nobel prize winner Herbert Simon in 1991. In the famous paper "*The Architecture of complexity"* the author states that "*By a complex system I mean one made up of a large number of parts that interact in a non-simple way. In such systems the whole is more than the sum of the parts, not in an ultimate, metaphysical sense but in the important pragmatic sense that, given the properties of the parts and the laws of their interaction, it is not a trivial matter to infer the properties of the whole […]"*[67]. He believes that complex financial systems exhibit a hierarchical organization whereby the whole system is decomposed into many different distinct subgroups which can be analyzed more easily: *"By a hierarchic system, or hierarchy, I mean a system that is composed of interrelated subsystems, each of the latter being, in turn, hierarchic in structure until we reach some lowest level of elementary subsystem"*[68]. Therefore, a hierarchical structure can help solving complex problems breaking them down into smaller and simpler subgroups whose solutions are then grouped together afterwards.

Nonetheless, inferring hierarchical relationships between the securities during the portfolio allocation process present many challenges. Indeed, the correlation matrices used to study the portfolio robustness, do not show a hierarchical structure. This issue is even more evident in the case of large covariance matrices. We have seen in the previous chapter, that, given the difficulty in estimating the expected returns, many scholars have developed theories and models which required only the estimation of the covariance matrix of asset returns. Among these techniques we find the most relevant risk-based portfolio allocation strategies such as: The Minimum Variance (MV), The Maximum Diversification (MD), The Maximum Sharpe Ratio (SR) and The Equal Risk Contribution (ERC). In order to predict a covariance matrix of size N, at least

[67] Simon, H. A. (1991). The architecture of complexity. In *Facets of systems science* (pp. 457-476). Springer, Boston, MA.
[68] Simon, H. A. (1991). Organizations and markets. *Journal of economic perspectives*, *5*(2), 25-44.

$\frac{N(N+1)}{2}$ independent and identically distributed (iid) returns observations are needed[69]. However, there is ample demonstration that asset returns exhibit heteroskedasticity with volatility clustering and invariant correlation structures over such long periods, resulting into serious estimation errors that may cancel out the advantages of portfolio diversification[70].

In order to overcome this problem, Marcos Lopez de Prado was the first researcher proposing a hierarchical model for the portfolio construction. The Spanish author uses graph theory and machine learning to construct a diversified portfolio with a Hierarchical Risk Parity Approach (HRP) which substantially differs from the traditional risk-based portfolio optimization models[71]. The HRP methodology avoids the inversion of the covariance matrix; the relationship of the securities in the portfolio is organized as a hierarchy where clusters of similar assets are created using the correlation coefficients. Substituting the traditional covariance structure with a hierarchical one, allows achieving three main goals: *"First, it fully utilizes the information contained in the covariance matrix. Second, it recovers the stability of the weights. And third, in contrast to most traditional risk-based asset allocation methods, it does not require the inversion of the covariance matrix"*[72].

2.2 The Problem of Quadratic Programming: The Critical Line Algorithm (CLA)

The first chapter of this thesis has already pointed out the importance of Harry Markowitz in the development of the portfolio theory and his overall contribution to economics. The mean-variance portfolio optimization approach is considered as the birth of portfolio allocation in finance. In addition to publishing his masterpiece *"Portfolio Selection"* in 1952, Markowitz, while working for the RAND Corporation, developed an algorithm for solving quadratic

[69] For instance, if we want to build a covariance matrix of asset returns for a portfolio constituted by 100 assets, we would ideally need 5,050 or at least 20 years of daily returns time series.
[70] Bai, Z. D., & Yin, Y. Q. (2008). Limit of the smallest eigenvalue of a large dimensional sample covariance matrix. In *Advances In Statistics* (pp. 108-127); de Prado, M. L. (2016). Building diversified portfolios that outperform out of sample. *The Journal of Portfolio Management*, *42*(4), 59-69; Zakamulin, V. (2015). A test of covariance-matrix forecasting methods. *The Journal of Portfolio Management*, *41*(3), 97-108.
[71] de Prado (2016), op. cit., pp.59-69.
[72] Burggraf, T., & Vyas, A. (2020). Beyond Risk Parity-A Machine Learning-based Hierarchical Risk Parity Approach on Cryptocurrencies. Available at SSRN 3534773.

problems in 1956. The mean-variance framework tries to find the value of an X that minimizes or maximizes a function *f*. As we have seen, the Markowitz optimization problem can be formulated in two different yet equal expressions: (i) Maximize the portfolio expected return for a given level of risk or (ii) Minimize the portfolio volatility (standard deviation) conditional to a given level of return. The breakthrough of Markowitz was to create an algorithm that could solve the optimization problem subject both to an equality constraint (that the weights of the holdings sum up to one) and an inequality constraint (a lower and an upper bound for the weights of each security in the portfolio). This approach, named the *Critical Line Algorithm* (CLA), allows to find the unique solution X after a known number of iterations as well as to generate the set of optimal portfolios that lie on the efficient frontier. For this reason, Harry Markowitz is widely accepted as the "*father of quadratic programming" (QP)*[73]. CLA was therefore developed by Markowitz to solve any quadratic programming problem subject to inequality constraints.

2.2.1 The framework

Given an investment universe of N= {*1,2, …, n*} assets, and a *nxn* positive covariance matrix Ω, we define the following inputs[74]:

- *w* is the (*nx1*) vector of security weights, which is the output we need to optimize.
- *L* is the (*nx1*) vector of lower bounds, with $w_i \geq L_i$, $\forall i \in N$.
- *U* is the (*nx1*) vector of upper bounds, with $w_i \geq U_i$, $\forall i \in N$.
- A subset *F* of *N* = {*1,2, …, n*}, containing the set of "free assets" whose weights respect the boundary $L_i \leq w_i \leq U_i$.
- $B \subset N$ is a subset of the holding weights that lie on one of the bounds. $B \cup F = N$.

The covariance matrix Ω, the vector of asset returns *u* and the vector of weights *w* can be therefore rewritten as:

$$\Omega = \begin{bmatrix} \Omega_F & \Omega_{FB} \\ \Omega_{BF} & \Omega_B \end{bmatrix}, u = \begin{bmatrix} u_F \\ u_B \end{bmatrix}, w = \begin{bmatrix} w_F \\ w_B \end{bmatrix} \quad (47)$$

[73] Cottle, R. W., & Infanger, G. (2010). Harry Markowitz and the Early History of Quadratic Programming. In *Handbook of Portfolio Construction* (pp. 179-211). Springer, Boston, MA.
[74] The paragraph 2.2.1 "The Framework" follows the same procedure adopted in: Bailey, D. H., & López de Prado, M. (2013). An open-source implementation of the critical-line algorithm for portfolio optimization. *Algorithms*, *6*(1), 169-196.

Where Ω_F is the *(kxk)* covariance matrix among free assets, Ω_B the *(n-k)x(n-k)* covariance matrix, and a *kx(n-k)* covariance matrix Ω_{BF} which is equal to Ω_{FB}^T. Furthermore, there are two *k* vectors u_F and w_F and two *(n-k)* vectors u_B and w_B.

When there is no constraint, the optimization problem can be solved by minimizing the Lagrange function[75] (48).

$$\zeta[w, \gamma, \lambda] = \frac{1}{2} w^T \Omega w - \gamma(w^T 1_n - 1) - \lambda\left(w^T u - u_p\right) \quad (48)$$

where γ and λ are the Lagrange multipliers and u_p is the excess return. The vector 1_n instead represents the (*nx1*) vector of ones. The Lagrange function presented above has two distinct conditions: first that the portfolio's weights sum up to one and second that the portfolio volatility is minimized at the mean return level u_p. The solution of the problem leads to a linear system of *n+2* equations, which finds the optimal variance minimizing portfolio weight vector w^*[76]. However, this solution is suboptimal since it does not respect the bounds condition and therefore will not meet the constraint: $L_i \leq w_i \leq U_i$. For this reason, the methodology of Lagrange multipliers is of difficult application, especially if, as Markowitz assumes in his famous mean-variance portfolio model, short-selling is not allowed. Indeed, in that case, the security weights must be positive. We shall include the necessary condition of an upper and lower bound; the optimal solution of the problem will then lie between these two bounds: $L_i \leq w_i \leq U_i$. In order to solve the problem under the constrained case, we should first understand the concept of *turning point*. "*A solution vector w* is a turning point if in its vicinity there is another solution vector with different free assets. This is important because in those regions of the solution space away from turning points the inequality constraints are effectively irrelevant with respect to the free assets. In other words, between any two turning points, the constrained solution reduces to solving the following unconstrained problem on the free assets*" *(49)*[77].

[75] In mathematical optimization, the method of Lagrange multipliers is a strategy for finding the local maxima and minima of a function subject to equality constraints. For a better explanation of the Lagrange multipliers see: https://en.wikipedia.org/wiki/Lagrange_multiplier#Interpretation_of_the_Lagrange_multipliers.

[76] Niedermayer, A., & Niedermayer, D. (2010). Applying Markowitz's critical line algorithm. In *Handbook of portfolio construction* (pp. 383-400). Springer, Boston, MA.

[77] de Prado (2013), op. cit., pp.169-196

$$\zeta[w,\gamma,\lambda] = \frac{1}{2}w_F^T\Omega_F w_F + \frac{1}{2}w_F^T\Omega_{FB}w_B + \frac{1}{2}w_B^T\Omega_{BF}w_F + \frac{1}{2}w_B^T\Omega_B w_B - \gamma(w_F^T 1_k + w_B^T 1_{n-k} - 1) - \lambda\left(w_F^T u_F + w_B^T u_B - u_p\right) \quad (49)$$

Where the value w_B is known and does not vary between turning points and the value w_F is the object of the minimization problem. Since: *"The efficient frontier can be simply derived as a convex combination between any two neighbor turning points",* the main challenge of the CLA algorithm proposed by Markowitz is to define each turning point and consequently, find the optimal portfolio at each turning point[78].

The two subsets *F* and *B* do not change between turning points, therefore looking for a solution between two turning points will require solving an optimization without constraint upon the subset *F* as in *(48)*. Therefore, *"Combining two neighboring turning points with a real weight w ∈ [0, 1] always leads to a constrained minimum variance portfolio"*[79].

2.2.2 A practical application of the Critical Line Algorithm

In this section I am going to present a practical application of the CLA using the Anaconda Jupiter notebook of Python 3.7[80].

The main idea of Markowitz algorithm for minimum variance optimization purpose is to first find the turning point associated with the highest expected return; after that, computing all the other turning points through an iterative process.

[78] Markowitz, H. (1955). The optimization of a quadratic function subject to linear constraints. *RAND CORP, (68). Santa Monica (CA).*
[79] Niedermayer et. Al. (2010), op.cit., pp. 383-400.
[80] The practical application is based on the same example discussed in section 5 of the paper: Bailey, D. H., & López de Prado, M. (2013). An open-source implementation of the critical-line algorithm for portfolio optimization. *Algorithms*, *6*(1), 169-196.

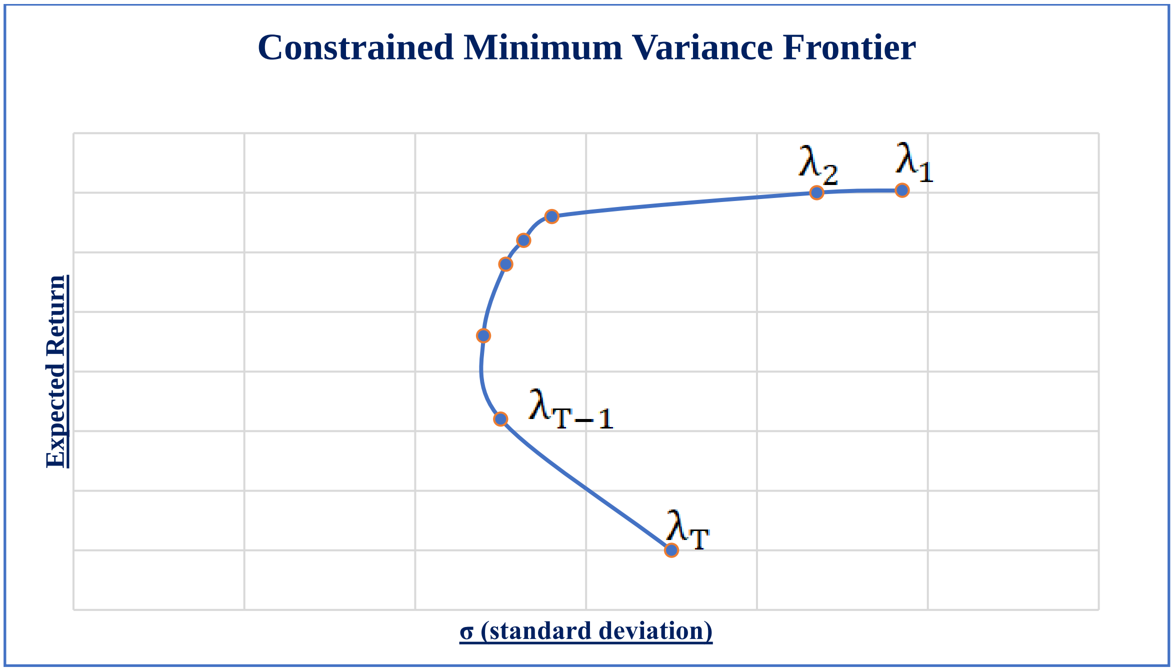


Figure 27, Constrained Minimum Variance Frontier for 10 assets. Data Source: A personal elaboration on Excel from: Niedermayer, A., & Niedermayer, D. (2010). Applying Markowitz's critical line algorithm. In Handbook of portfolio construction (pp. 383-400). Springer, Boston, MA.

Figure 27 above presents a constrained minimum variance efficient frontier application using 10 assets. Each orange dot on the line represents a turning point. λ_1 is the multiplier of the first turning point, the one that is associated with the highest expected return. The Lagrange multiplier λ decreases when moving downwards on the efficient frontier, with λ_T representing the lowest multiplier for the last turning point[81].

The goal of the algorithm is to find a solution for the constrained problem. It starts with the turning point with the highest mean return value. For this reason, it groups the assets according to their expected return, in a decreasing order. At the beginning, all the securities weights are fixed to their lower bound (w_i=L_i), then, the weight of the first asset is increased until the upper bound is reached. If $w^T 1 < 1$, the second holding weight is increased. This procedure is repeated for all the assets under examination until the sum of the weights exceeds one. The

[81] λ decreases when moving downwards because λ and $w^T u$ are linearly and positively related.

last iterated weight is then decreased so that the constraint $w^T 1=1$ is respected. *"This last weight is the first free asset, and the resulting vector of weights the first turning point"*[82].

Moving from one turning point to another one, calls for either adding or subtracting a new component (asset) to or from the subset of free assets *F*. This methodology ends when the algorithm determines the optimal expected return. All the initialization process is implemented into the Python code with the function: *def initAlgo(self)* in *Snippet 6.* Since the subset *F* cannot be an empty one, it starts with only one asset, with the option of increasing its size by adding other elements. This search of this additional element *k* is made through the function *def getB(self, f). k* has to be found in the subset *B*, which is a non-empty subset and complement to *F*. Lambda λ, is computed using the matrices created by the function *def reduceMatrix(self,matrix,listX,listY)*. The formula (*50)* for computing the multiplier is expressed through the function *def computeLambda (self, covarF_inv, covarFB, meanF, wB,i,bi).*

$$\lambda = \frac{1}{C}[(1 - 1_{n-k}^T w_B + 1_k^T \Omega_F^{-1} \Omega_{FB} w_B)(\Omega_F^{-1} 1_k)_i \\ - (1_k^T \Omega_F^{-1} 1_k)(b_i + (\Omega_F^{-1} \Omega_{FB} w_B)_i](50)$$

Where

$$C = -(1^T \Omega_F^{-1} 1_k)(\Omega_F^{-1} u_F)_i + (1_k^T \Omega_F^{-1} u_k)(\Omega_F^{-1} 1_k)_{\ i}$$

$$b_i = \begin{cases} U_i \ \ if \ C_i > 0 \\ L_i \ if \ C_i < 0 \end{cases}$$

Furthermore, the function *def computeW(self,covarF_inv,covarFB,meanF,wB)* computes the value of the free weights in the subsequent turning point, w_F. In order to accomplish that, we have to encode the value of γ, the other multiplier (*51).*

$$\gamma = -\lambda \frac{1_k^T \Omega_F^{-1} u_F}{1_k^T \Omega_F^{-1} 1_k} + \frac{1 - 1_{n-k}^T w_B + 1_k^T \Omega_F^{-1} \Omega_{FB} w_B}{1_k^T \Omega_F^{-1} 1_k} \ (51)$$

Finally, the value of the free weight can be expressed through the formula (*52).*

[82] de Prado (2013), op. cit., pp.169-196.

$$w_F = -\Omega_F^{-1}\Omega_{FB}w_B + \gamma\Omega_F^{-1}1_k + \gamma\Omega_F^{-1}u_F \quad (52)$$

```
#---------------------------------------------------------------------------------------------------
# Initialize the class
class CLA:
    def __init__(self,mean,covar,lB,uB):
        self.mean=mean
        self.covar=covar
        self.lB=lB
        self.uB=uB
        self.w=[] # solution
        self.l=[] # lambdas
        self.g=[] # gammas
        self.f=[] # free weights
# Initialize the algo
def initAlgo(self):
        #1) Form structured array
        a=np.zeros((self.mean.shape[0]),dtype=[('id',int),('mu',float)])
        b=[self.mean[i][0] for i in range(self.mean.shape[0])] # dump array into list
        a[:]=list(zip(range(self.mean.shape[0]),b)) # fill structured array

       2) Sort structured array
        b=np.sort(a,order='mu')

     #3) First free weight
        i,w=b.shape[0],np.copy(self.lB)
        while sum(w)<1:
            i-=1
```

```
          w[b[i][0]]=self.uB[b[i][0]]
      w[b[i][0]]+=1-sum(w)
      return [b[i][0]],w
#------------------------------------------------------------------------------------------------
#Defining the bounds
def computeBi(self,c,bi):
      if c>0:
          bi=bi[1][0]
      if c<0:
          bi=bi[0][0]
      return bi
#------------------------------------------------------------------------------------------------
#Defining the free weights
def computeW(self,covarF_inv,covarFB,meanF,wB):
      #1) compute gamma
      onesF=np.ones(meanF.shape)
      g1=np.dot(np.dot(onesF.T,covarF_inv),meanF)
      g2=np.dot(np.dot(onesF.T,covarF_inv),onesF)
      if wB is None:
          g,w1=float(-self.l[-1]*g1/g2+1/g2),0
      else:
          onesB=np.ones(wB.shape)
          g3=np.dot(onesB.T,wB)
          g4=np.dot(covarF_inv,covarFB)
          w1=np.dot(g4,wB)
          g4=np.dot(onesF.T,w1)
          g=float(-self.l[-1]*g1/g2+(1-g3+g4)/g2)
```

```
 #2) compute weights
                w2=np.dot(covarF_inv,onesF)
                w3=np.dot(covarF_inv,meanF)
               return -w1+g*w2+self.l[-1]*w3,g
#------------------------------------------------------------------------------------------------------------
#Computing the Lambda multiplier
def computeLambda(self,covarF_inv,covarFB,meanF,wB,i,bi):
       #1) C
       onesF=np.ones(meanF.shape)
       c1=np.dot(np.dot(onesF.T,covarF_inv),onesF)
       c2=np.dot(covarF_inv,meanF)
       c3=np.dot(np.dot(onesF.T,covarF_inv),meanF)
       c4=np.dot(covarF_inv,onesF)
       c=-c1*c2[i]+c3*c4[i]
       if c==0:return
       #2) bi
       if type(bi)==list:bi=self.computeBi(c,bi)
       #3) Lambda
       if wB is None:
          # All free assets
          return float((c4[i]-c1*bi)/c),bi
       else:
          onesB=np.ones(wB.shape)
          l1=np.dot(onesB.T,wB)
          l2=np.dot(covarF_inv,covarFB)
          l3=np.dot(l2,wB)
          l2=np.dot(onesF.T,l3)
          return float(((1-l1+l2)*c4[i]-c1*(bi+l3[i]))/c),bi
```

```
#----------------------------------------------------------------------------------------------------
# Slice covarF,covarFB,covarB,meanF,meanB,wF,wB
getMatrices(self,f):
    covarF=self.reduceMatrix(self.covar,f,f)
    meanF=self.reduceMatrix(self.mean,f,[0])
    b=self.getB(f)
    covarFB=self.reduceMatrix(self.covar,f,b)
    wB=self.reduceMatrix(self.w[-1].reshape(-1,1),b,[0])
    return covarF,covarFB,meanF,wB
#----------------------------------------------------------------------------------------------------
#Computing the subset B
getB(self,f):
    return self.diffLists(range(self.mean.shape[0]),f)
#----------------------------------------------------------------------------------------------------
# Reduce a matrix to the provided list of rows and columns
Def reduceMatrix(self,matrix,listX,listY):
    if len(listX)==0 or len(listY)==0:return
    matrix_=matrix[:,listY[0]:listY[0]+1]
    for i in listY[1:]:
        a=matrix[:,i:i+1]
        matrix_=np.append(matrix_,a,1)
    matrix__=matrix_[listX[0]:listX[0]+1,:]
    for i in listX[1:]:
        a=matrix_[i:i+1,:]
        matrix__=np.append(matrix__,a,0)
    return matrix__
```

Snippet 6, Encoding the Critical Line Algorithm (CLA). Data Source: Bailey, D. H., & López de Prado, M. (2013). An open-source implementation of the critical-line algorithm for portfolio optimization. Algorithms, 6(1), 169-196.

The algorithm computes all the turning points as well as the global minimum variance portfolio, which as explained in section 1.3.4, is the left-most point that lies on the efficient frontier. The input is encoded through the function *def getMinVar(self)* in *Snippet 7*. On the other hand, the function *def getMaxSR(self)*, finds the portfolio on the constrained efficient frontier exhibiting the highest Sharpe ratio. Lastly, every portion of the efficient frontier can be obtained as a convex combination between any two close turning pints. This is performed by the function *def efFrontier(self,points)* which returns the expected returns, standard deviations and the security's weights as outputs.

```
#---------------------------------------------------------------------------------------------------
# Get the minimum variance solution
def getMinVar(self):
    var=[]
    for w in self.w:
        a=np.dot(np.dot(w.T,self.covar),w)[0,0]
        var.append(a)
    return min(var)**.5,self.w[var.index(min(var))]
#---------------------------------------------------------------------------------------------------
# Get the max Sharpe ratio portfolio
def getMaxSR(self):
    #1) Compute the local max SR portfolio between any two neighbor turning points
    w_sr,sr=[],[]
    for i in range(len(self.w)-1):
        w0=np.copy(self.w[i])
        w1=np.copy(self.w[i+1])
        kargs={'minimum':False,'args':(w0,w1)}
        a,b=self.goldenSection(self.evalSR,0,1,**kargs)
        w_sr.append(a*w0+(1-a)*w1)
        sr.append(b)
```

```
#---------------------------------------------------------------------------------------------------
# Get the efficient frontier
def efFrontier(self,points):
    mu,sigma,weights=[],[],[]
    a=np.linspace(0,1,points/len(self.w))[:-1] # remove the 1, to avoid duplications
    b=range(len(self.w)-1)
    for i in b:
      w0,w1=self.w[i],self.w[i+1]
      if i==b[-1]:a=np.linspace(0,1,points/len(self.w)) # include the 1 in the last
iteration
      for j in a:
        w=w1*j+(1-j)*w0
        weights.append(np.copy(w))
        mu.append(np.dot(w.T,self.mean)[0,0])
        sigma.append(np.dot(np.dot(w.T,self.covar),w)[0,0]**.5)
    return mu,sigma,weights
```

Snippet 7, Getting Min Var, Max SR and the Efficient Frontier solution. Data Source*: Bailey, D. H., & López de Prado, M. (2013). An open-source implementation of the critical-line algorithm for portfolio optimization. Algorithms, 6(1), 169-196.*

Having set all the needed inputs to run the algorithm properly, we can now test it, illustrating a small numerical example that considers an investment universe of 10 securities. The lower bounds are equal to 0 while the upper ones are set equal to 1. The weights of the assets in our portfolio must satisfy the following: $\sum_{k=1}^{10} w_i = 1$. *Figure 28* below, shows the values of the bounds, mean vector as well as the covariance matrix of the 10 assets.

The CLA will return a list of 10 turning points (TP) and for each them the corresponding value of the mean return, standard deviation, weights and the multipliers λ and γ are displayed (*Figure 29*).

LB	0	0	0	0	0	0	0	0	0	0
UB	1	1	1	1	1	1	1	1	1	1
u	1,175	1,19	0,396	1,12	0,346	0,679	0,089	0,73	0,481	1,08
Cov	0,4075516									
	0,0317584	0,9063047								
	0,0518392	0,0313639	0,194909							
	0,056639	0,0268726	0,0440849	0,1952847						
	0,0330226	0,0191717	0,0300677	0,0277735	0,3405911					
	0,0082778	0,0093438	0,0132274	0,0052667	0,0077706	0,1598387				
	0,0216594	0,0249504	0,0352597	0,0137751	0,0206784	0,0210558	0,6805671			
	0,0133242	0,0076104	0,00115493	0,0078088	0,0073641	0,0051869	0,0137788	0,9552692		
	0,0343476	0,0287487	0,0427563	0,0291418	0,0254266	0,0172374	0,0462703	0,0106553	0,3168158	
	0,022499	0,0133687	0,020573	0,0164038	0,0128408	0,0072378	0,0192609	0,0076096	0,0185432	0,1107929

Figure 28, Lower Bounds, Upper Bounds, Means and Covariance. Data Source: A personal elaboration from Bailey, D. H., & López de Prado, M. (2013).

NUM	**Return**	**Risk**	**Lambda**	**X(1)**	**X(2)**	**X(3)**	**X(4)**	**X(5)**	**X(6)**	**X(7)**	**X(8)**	**X(9)**	**X(10)**
1	1,190	0,952	58,303	0,000	1,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
2	1,180	0,546	4,174	0,649	0,351	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
3	1,160	0,417	1,946	0,434	0,231	0,000	0,335	0,000	0,000	0,000	0,000	0,000	0,000
4	1,111	0,267	0,165	0,127	0,072	0,000	0,281	0,000	0,000	0,000	0,000	0,000	0,520
5	1,108	0,265	0,147	0,123	0,070	0,000	0,279	0,000	0,000	0,000	0,006	0,000	0,521
6	1,022	0,230	0,056	0,087	0,050	0,000	0,224	0,000	0,174	0,000	0,030	0,000	0,435
7	1,015	0,228	0,052	0,085	0,049	0,000	0,220	0,000	0,180	0,000	0,031	0,006	0,429
8	0,973	0,220	0,037	0,074	0,044	0,000	0,199	0,026	0,198	0,000	0,033	0,028	0.398
9	0,950	0,216	0,031	0,068	0,041	0,015	0,188	0,034	0,202	0,000	0,034	0,034	0,383
10	0,803	0,205	0,000	0,037	0,027	0,095	0,126	0,077	0,219	0,030	0,036	0,061	0,292

Figure 29, Return, Risk, Multipliers and the weights of the 10 TPs. Data source: A personal elaboration from: Bailey, D. H., & López de Prado, M. (2013).

From the figure above, we can notice that the first solution is found when the second asset of the portfolio is set as the free asset, with a weight equal to 100%. Subsequently assets 1,4,10,8,6,9,5,3 and 7 are added by lowering the value of λ until the minimum variance portfolio is reached at turning point 10.

Figure 30, shows the efficient frontier (the first 100 points). This is performed in Python by using the function *cla.efFrontier(100).* The minimum variance portfolio exhibits a risk of 0,2052 while the maximum Sharpe ratio portfolio returns a Sharpe ratio of 4,4535 for a risk of 0,2274[83].

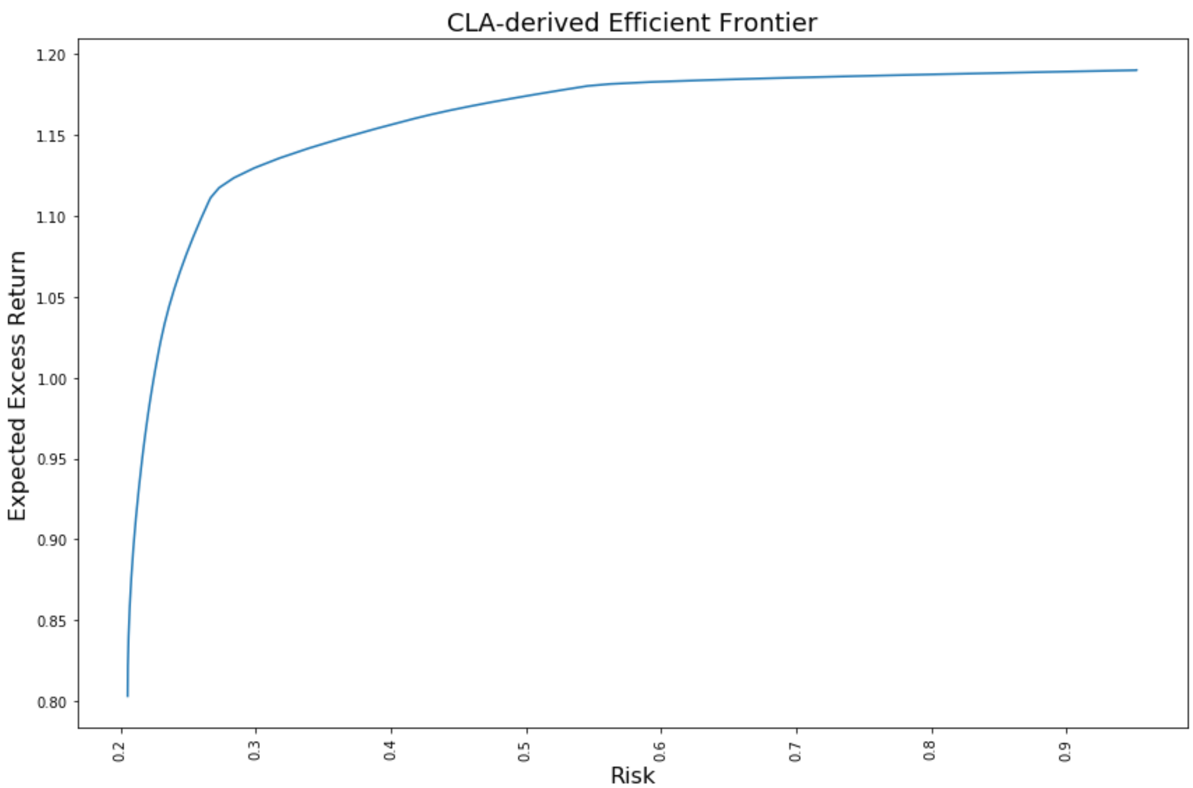


Figure 30, CLA-constrained Efficient Frontier. Data Source: A personal elaboration on Python.

[83] These results are taken from a personal implementation of the exercise using the Snippet's codes on the Anaconda Jupiter Notebook.

2.3 The Hierarchical Risk Parity Portfolio

The CLA algorithm manages to ensure an exact solution is found after a known number of iterations. It is a quadratic programming that allows to build a constrained efficient frontier and to look for the optimal portfolio in terms of mean return and variance. Nonetheless, it is subject to several drawbacks which make the solutions provided by the algorithm somewhat unstable and inaccurate. Indeed, one of the major caveats of CLA is the instability of forecasted returns: "*Small deviations in the forecasted returns will cause CLA to produce very different portfolios"*[84]. For this reason, researchers and economists have tried to develop new models and theories that would base their results on the estimation of the covariance matrix rather than the returns. This has led to the so called "Risk-based" portfolio optimization problems, which have been deeply analyzed in section 1.3. Despite their good performance and their applicability on industry-wide portfolio optimization problems, they tend to provide affected results due to their great sensitivity to the covariance matrix inversion. All these quadratic programming methodologies do require the inversion of a positive-definite covariance matrix, which is: *"prone to large errors when the covariance matrix is numerically ill-conditioned, i.e. it has a high condition number"*[85]. When this number is too high, the covariance-correlation matrix becomes too unstable. This issue becomes even larger if we add more correlated (multicollinear-investments) for diversification. "*The more correlated the investments, the greater the need for diversification, and yet the more likely we will receive unstable solutions. Therefore, the benefits of diversification often are more than offset by estimation errors"*[86]. Bailey and Lòpez de Prado, address to this problem the name of **"*Markowitz' curse*".**

Furthermore, the traditional risk-based portfolios usually have shown to provide poor results out of sample, so much that even the benchmark naïve (equally-weighted) portfolios return better risk-performance results than mean-variance and risk-based optimization techniques[87]. Therefore, because of the instability and inaccuracy of the aforementioned portfolio optimization

---

[84] Michaud, R. O., & Michaud, R. (2007). Estimation error and portfolio optimization: a resampling solution. *Available at SSRN 2658657*.
[85] Bailey, D. H., & Lopez de Prado, M. (2012). Balanced baskets: a new approach to trading and hedging risks. *Journal of Investment Strategies (Risk Journals), 1*(4).
The condition number of a covariance and correlation matrix is defined as the absolute value of the ratio between its maximal and minimal eigenvalues.
[86] de Prado (2013), op. cit., pp.59-69.
[87] DeMiguel et. Al. (2009), op. cit., pp. 1915-1953.

methodologies, Bailey and Lòpez de Prado came up with a new approach that could address the major quadratic programming limitations: the Hierarchical Risk Parity portfolio. First, it fully concentrates on the covariance matrix, hence dropping the forecasted returns. The major benefit of the Hierarchical Risk Parity approach is that it does not require the inversion of the covariance matrix, which is a highly desirable characteristic when the matrix has a high condition number. Second, it "*proposes a hierarchical implementation of an inverse-variance allocation with weights calculated between clusters of correlated asset returns*[88]".

The HRP algorithm, performs the optimization process through three distinct phases: **1) Tree Clustering, 2) Quasi- diagonalization** and **3) Recursive Bisection**.

2.3.1 Tree clustering

The main rationale behind this procedure is straightforward. Imagine you are an investor who decides to diversify his financial resources into different asset classes, hence creating a portfolio with stocks, bonds, real estate, hedge funds etc. Among the securities he has invested in, some are substitutes of one another. Stocks may be divided according to the industry, size, liquidity. For instance, when deciding the portfolio's weight of a stock like Facebook, we should consider either decreasing or adding the allocation to another multinational company in the social network industry like Twitter rather than a small firm operating locally. Therefore, if in your portfolio there are securities which exhibit a greater correlation between themselves, you should first diversify the weights among them, and then consider the rest of the portfolio. Suppose you decide the weights for Facebook and Twitter in your portfolio are respectively 60/40. However, if you consider them as a unique entity for further diversification, you may realize that in the whole portfolio Facebook and Twitter represent only the 10%, therefore the real weights are 6% to Facebook and 4% to Twitter.

One of the main problems of the CLA algorithm is that it does not consider the hierarchical structure. Indeed: "*to a correlation matrix, all investments are potential substitutes to each other. In another words, correlation matrices lack the notion of hierarchy. This lack of*

[88] Pfitzinger, J., & Katzke, N. (2019). *A constrained hierarchical risk parity algorithm with cluster-based capital allocation* (No. 14/2019).

hierarchical structure allows weights to vary freely in unintended ways, which is a root cause of CLA’s instability’’[89].

A typical example of a hierarchical structure is a tree structure. A tree structure is suitable when analyzing the portfolio composition. Indeed, it has only N-1 edges to connect N nodes, allowing weights to re-adjust among the securities of the same hierarchy. Furthermore, in a tree structure, the weights are distributed top-down. *Figures 31* and *32* show the passage from geometric to hierarchical relationship, through a tree structure.

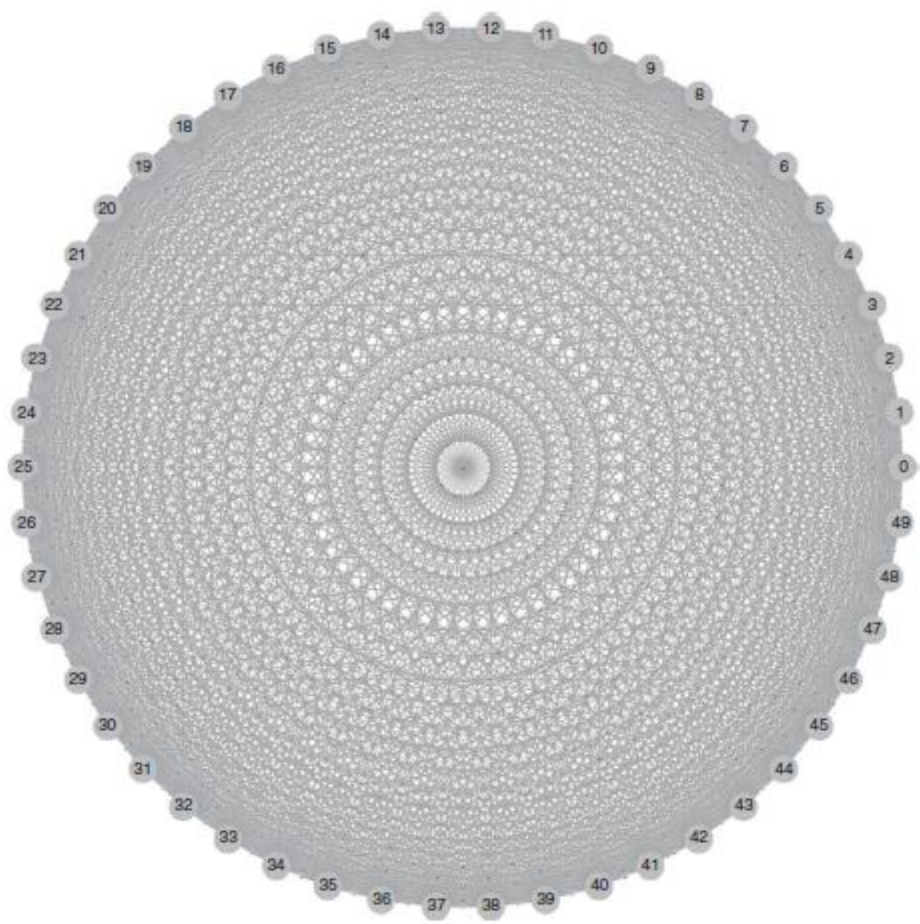


Figure 31, The complete top-graph without the notion of Hierarchy. Data Source: A personal elaboration on Python from: Bailey, D. H., & López de Prado, M. (2016).

[89] De Prado, M. L. (2018). *Advances in financial machine learning*. John Wiley & Sons.

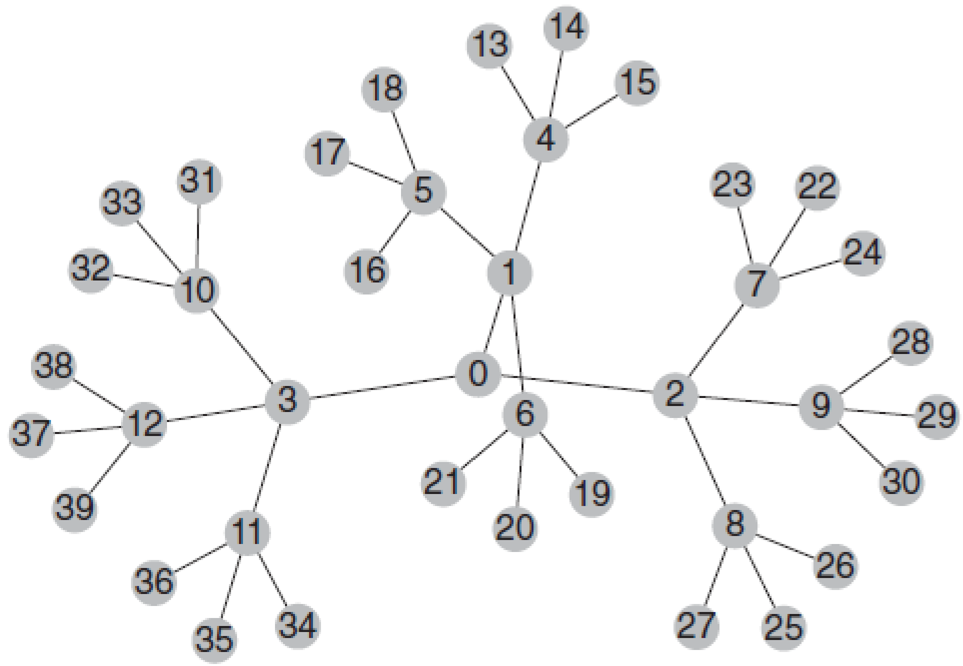


Figure 32, Representation of a tree structure. Data Source: A personal elaboration on Python from: Bailey, D. H., & López de Prado, M. (2016).

Let's consider a *TxN* matrix of observations X (N assets with returns series over time T). Next we can compute the corresponding correlation and covariance matrices. The first goal is to "*combine these N column-vectors into a hierarchical structure of clusters, so that allocations can flow downstream through a tree graph"*[90]. Therefore, we calculate a *NxN* correlation matrix with entries:

$$\rho = \left\{\rho_{i,j}{}_{i,j=1,\dots,N}\right\} \text{ where } \rho_{i,j} = \rho[X_i, X_j]$$

Then, the distance matrix *D*: (X_i, X_j) is defined as[91]:

$$D(X_i, X_j) = \sqrt{\frac{1}{2}\left(1 - \rho_{i,j}\right)} \quad (53)$$

[90] de Prado, (2018), Op. cit., pp. 224-226.
[91] D is defined in such a way that is a proper metric space. For a detailed proof see: de Prado (2013), op. cit., pp.59-69.

The second step of the tree-clustering stage consists in calculating the Euclidean distance between any two columns vector of *D*, which gives us the augmented distance matrix $\bar{D}$.

$$\bar{D}(i,j) = \sqrt{\sum_{k=1}^{N}\left(D(k,i) - D(k,j)\right)^2} \quad (54)$$

The main difference between equations *53* and *54* is that the first computes the distance between any two securities *i* and *j* in the portfolio, while the second is the distance of those two assets and the remaining part of the portfolio[92]. $\bar{D}(i,j)$ is therefore a function of the whole correlation matrix. The next step consists in forming the first cluster (*i*, j**). This can be done by taking the pair that returns the least distance:

$$U[1] = \arg\min_{i,j} \bar{D}(i,j) \quad (55)$$

Where *U* is the set of clusters. After that, we need to update the distance matrix $\bar{D}$, trough a passage known as the "*linkage criterion*"[93]. The distance between the first clustered item *U[1]* and any other asset *i* is therefore computed as follows:

$$\bar{D}(i, U[1]) = \min(\bar{D}(i,i^*), \bar{D}(i,j^*)) \quad (56)$$

This step is repeated for each security in the portfolio; each time a new cluster of assets is formed, the algorithm updates the distance matrix, until only one cluster is left[94].

Figure 33 reported below, reports a typical hierarchical portfolio visualization: a dendrogram graph. The image shows how the clusters are created for similar investments. For the specific case, I have analyzed a portfolio composed by ten securities, namely: ['AAPL' (Apple), 'AMZN' (Amazon), 'AXP' (American Express), 'BA' (Bank of America), 'CSCO' (Cisco), 'FB' (Facebook), 'IBM' (IBM), 'JPM' (JPMorgan Chase)]. The x-axis reports the name of the assets in the analyzed portfolio, while the y-axis measures the distance between the two merging leaves. The figure shows how clusters are formed at each iteration: for instance, the

---

92 For a deeper reading on the matter, see: Jaeger, M., Krügel, S., Marinelli, D., Papenbrock, J., & Schwendner, P. (2020). Understanding Machine Learning for Diversified Portfolio Construction by Explainable AI. *Available at SSRN*.
93 de Prado, (2018), Op. cit., pp. 224-226.
94 For a deeper analysis of each step, see: de Prado, (2016), Op. cit., pp.59-69.

securities Facebook, Google and Microsoft are grouped together representing a cluster; the same happens with JPMorgan and American Express, given their similar industry and sector. The tree-clustering process is encoded in the Python computer programming language and reported in *Snippet 8*[95].

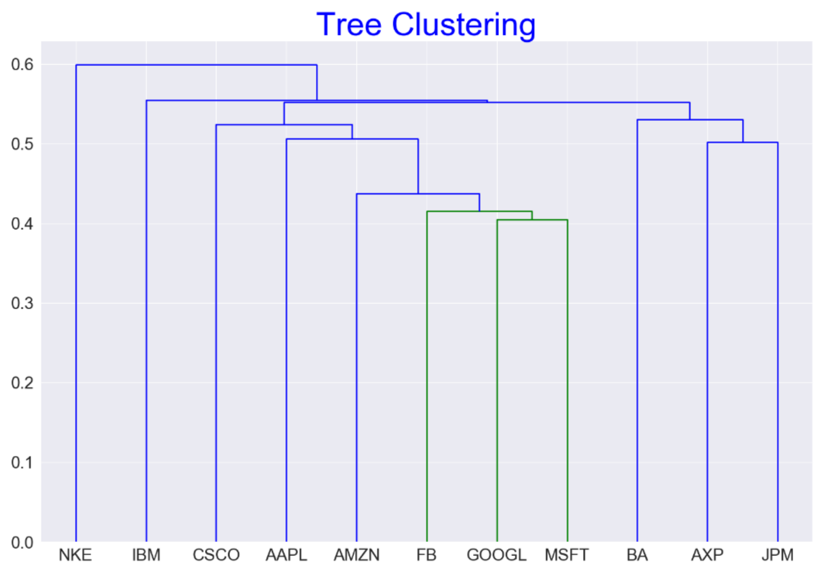


Figure 33, Tree Clustering & Dendrogram Graph. Data Source: A personal elaboration on Python.

```
# correlation matrix
corr = returns.corr()
# distance matrix
d_corr = np.sqrt(0.5*(1-corr))
#Tree Clustering
def tree_clustering(dist_mat, method="single", metric =  'eculidean'):
    flat_dist_mat = squareform(dist_mat) #distance array
    res_linkage = linkage(flat_dist_mat, method=method, metric = metric)
    return res_linkage
```

Snippet 8, Encoding Tree Clustering. Data Source: A personal elaboration on Python from: Bailey, D. H., & López de Prado, M. (2016).

[95] The codes are taken from de Prado, (2016), and reported into the Python 3.7 Anaconda Jupyter notebook.

2.3.2 Quasi-diagonalization

The second part of the HRP algorithm consists in the quasi-diagonalization stage. The approach allows a reorganization of the columns and rows of the covariance matrix using the information of the formed clusters, so that the largest entries are placed along the diagonal. Therefore, high correlations lie adjacently and along the matrix diagonal. The main goal of the quasi-diagonalization algorithm is to group similar holdings together while dissimilar ones lie around the matrix: *"We know that each row of the linkage matrix merges two branches into one. We replace clusters in ($y_{N-1,1}$,$y_{N-1,2}$) with their constituents recursively until no clusters remain. These replacements preserve the order of the clustering. The output is a sorted list of original nodes"*[96] *(Snippet 9).*

Figures 34 and *35* respectively report the correlation matrix before and after the quasi-diagonalization algorithm is applied[97]. As we can notice from *Figure 35* the darker-colored squares (representing a higher correlation coefficient) are all concentrated around the diagonal matrix.

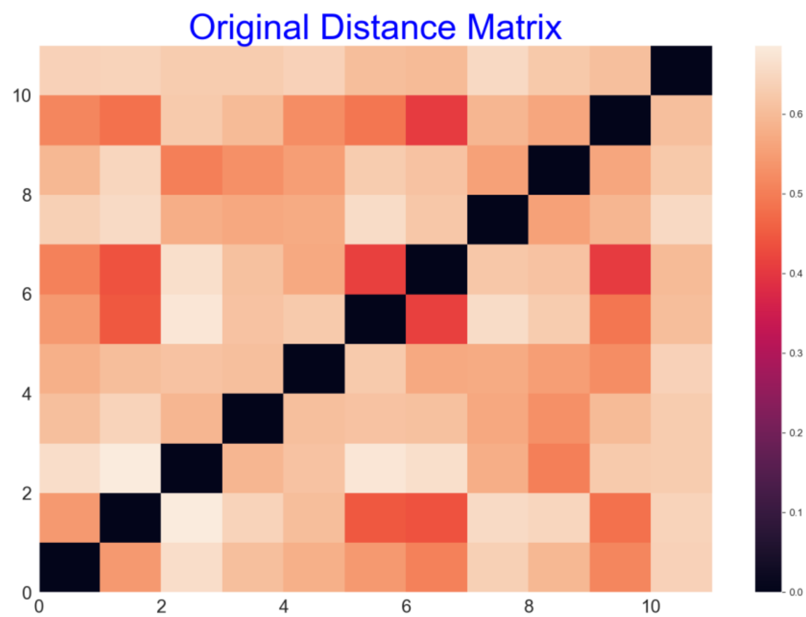


Figure 34, Original Distance Matrix. Data Source: A personal elaboration on Python.

[96] de Prado, (2018), Op. cit., pp. 224-226.
[97] The Figures are the result of an empirical application on Python. The portfolio analyzed is the same as the one reported for the Tree-Clustering Dendrogram study in section 2.3.1.

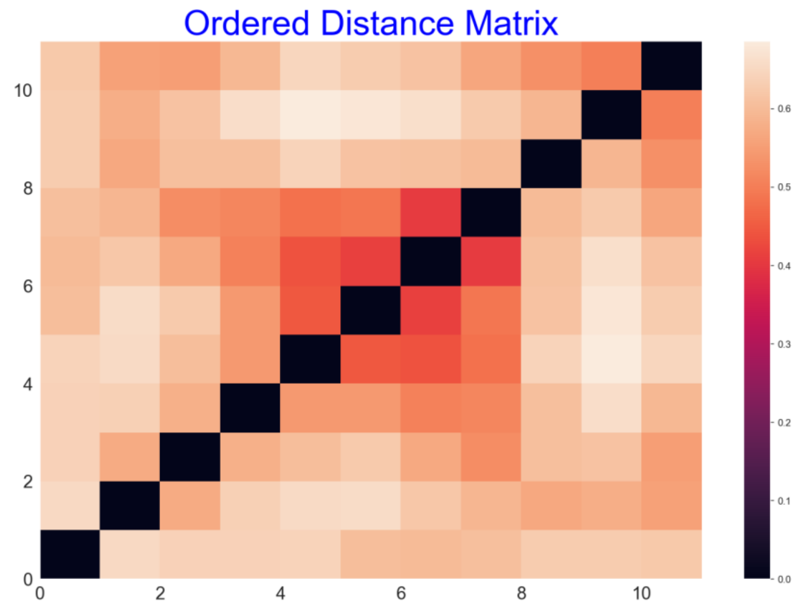


Figure 35, Ordered Distance Matrix following the Quasi-Diagonalization process. Data Source: a personal elaboration on Python.

```
#Returns the order implied by a hierarchical tree (dendrogram).
def seriation(Z, N, cur_index):

    :param Z: A hierarchical tree (dendrogram).
    :param N: The number of points given to the clustering process.
    :param cur_index: The position in the tree for the recursive traversal.
    :return: The order implied by the hierarchical tree Z.
  """
  if cur_index < N:
      return [cur_index]
  else:
      left = int(Z[cur_index - N, 0])
      right = int(Z[cur_index - N, 1])
      return (seriation(Z, N, left) + seriation(Z, N, right)
```

```
# Returns a sorted distance matrix.
def compute_serial_matrix(dist_mat, method="single"):
    N = len(dist_mat)
    flat_dist_mat = squareform(dist_mat)

    res_linkage = linkage(flat_dist_mat, method=method)
    res_order = seriation(res_linkage, N, N + N - 2)

    seriated_dist = np.zeros((N, N)) #
    a,b = np.triu_indices(N, k=1)

    seriated_dist[a,b] = dist_mat[[res_order[i] for i in a], [res_order[j] for j in b]]
    seriated_dist[b,a] = seriated_dist[a,b]
    return seriated_dist, res_order, res_linkage

ordered_dist_mat, res_order, res_linkage = compute_serial_matrix(d_corr.values)
```

Snippet 9, Encoding the Quasi-Diagonalization Process. Data Source: A personal elaboration on Python from: Bailey, D. H., & López de Prado, M. (2016).

2.3.3 Recursive bisection

The recursive bisection is the last stage of the HRP algorithm and the most important one since it defines the optimal allocation by assigning the final weights to the securities in the portfolio. Here, the algorithm takes advantage of the portfolio's feature that: "*the inverse-variance allocation is optimal for a diagonal covariance matrix*"[98].

[98] For a deeper study, see: https://hudsonthames.org/an-introduction-to-the-hierarchical-risk-parity-algorithm/.

Following the tree-clustering procedure, the algorithm further divides each cluster into two other sub-clusters V_1 and V_2, starting from the final cluster *U[N]*. Given the portfolio weights $w_i = 1, \forall i = 1,..,N$, the variance of each sub-cluster is computed as follows[99]:

$$V_{1,2} = w^T V w \ (57)$$

Where

$$w = \frac{diag(V)^{-1}}{trace(diag(V)^{-1})} \ (58)$$

Then, the HRP computes two weighting factors which are calculated respectively:

$$\alpha_1 = 1 - \frac{V_1}{V_1 + V_2}, \alpha_2 = 1 - \alpha_1 \ (59)$$

Given these two weighting factors, the algorithm runs the updated portfolio weights for each sub-cluster. Therefore, only the holdings within each cluster are considered for the final portfolio allocation; the weights w_1 and w_2 for the two sub-clusters are thus:

$$w_1 = \alpha_1 * w_1, \ \ w_2 = \alpha_2 * \ w_2 \ (60)$$

"This top-down assignment of weights is an advantage of HRP over other allocation algorithms - only assets within the same group compete for allocation with each other rather than competing with all the assets in the portfolio"[100]. The whole algorithm guarantees that $0 \leq w_i \leq 1, \forall i = 1,..,N \ and \ \sum_{i=1}^{N} w_i = 1$. *Figures 36* and *37* report the Hierarchical Risk Parity portfolio weights after running the recursive bisection stage, while *Snippet 10* reports its implementation on Python[101].

[99] The Recursive Bisection procedure follows the same explanation reported in: Burggraf et. Al., (2020), Op. cit., Available at SSRN 3534773.
[100] Burggraf et. Al., (2020), Op. cit., Available at SSRN 3534773.
[101] The Figures are the result of an empirical application on Python. The portfolio analyzed is the same as the one reported for the Tree-Clustering Dendrogram study in section 2.3.1.

Assets	Weights
AAPL	0,074009
AMZN	0,049352
AXP	0,078185
BA	0,102499
CSCO	0,126099
FB	0,075914
GOOGL	0,078333
IBM	0,168443
JPM	0,070235
MSFT	0,067648
NKE	0,109283

Figure 36, HRP portfolio's weights. Data Source: A personal elaboration on Python.

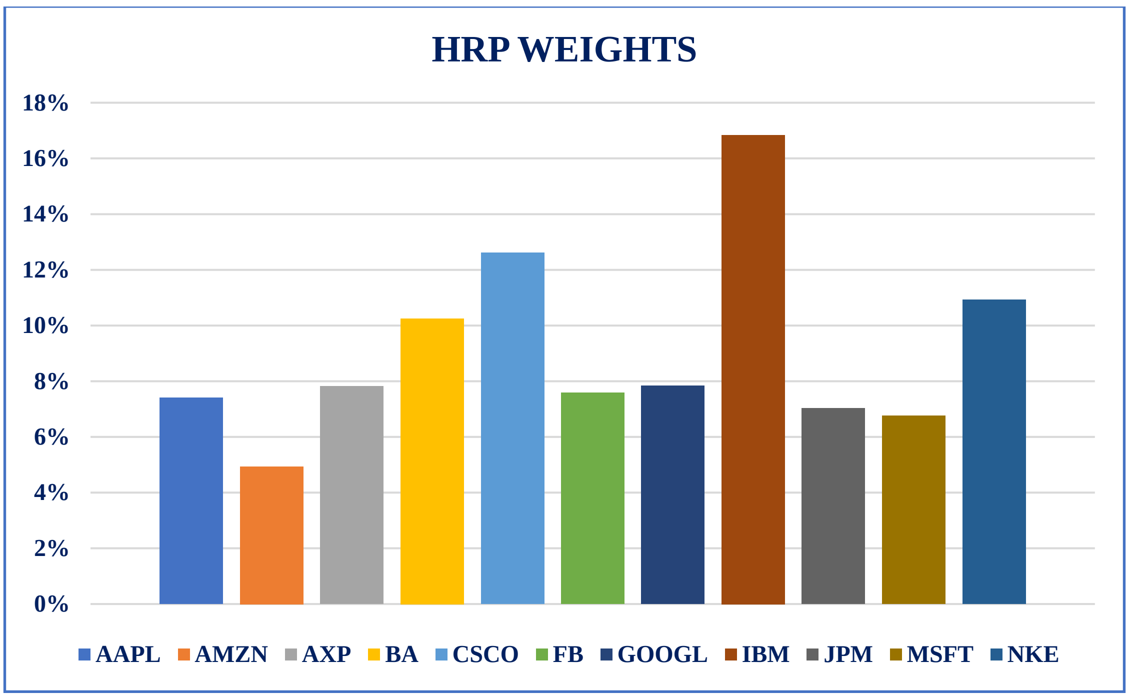


Figure 37, Graphical Representation of the portfolio's weights. Data Source: A personal elaboration on Python.

```
#Encoding the Recursive Bisection Stage
def compute_HRP_weights(covariances, res_order):
    weights = pd.Series(1, index=res_order)
    clustered_lists = [res_order]
    while len(clustered_lists) > 0:

        clustered_lists = [cluster[start:end] for cluster in clustered_lists
                           for start, end in ((0, len(cluster) // 2),
                                              (len(cluster) // 2, len(cluster)))
                           if len(cluster) > 1]

        for subcluster in range(0, len(clustered_lists), 2):
            left_cluster = clustered_lists[subcluster] # divide into groups every two lists; take the left cluster (list)
            right_cluster = clustered_lists[subcluster + 1] # take the right cluster (list)
            left_subcovar = covariances.iloc[left_cluster, left_cluster] # the covariance matrix of the indexes in left clusters

            inv_diag = 1 / np.diag(left_subcovar.values)

            parity_w = inv_diag * (1 / np.sum(inv_diag))
            left_cluster_var = np.dot(parity_w, np.dot(left_subcovar, parity_w))

            right_subcovar = covariances.iloc[right_cluster, right_cluster] # the covariance matrix of the indexes in right clusters
            inv_diag = 1 / np.diag(right_subcovar.values)
            parity_w = inv_diag * (1 / np.sum(inv_diag))

```

```
            right_cluster_var = np.dot(parity_w, np.dot(right_subcovar, parity_w))
            alloc_factor = 1 - left_cluster_var / (left_cluster_var + right_cluster_var)

            weights[left_cluster] *= alloc_factor
            weights[right_cluster] *= 1 - alloc_factor
    return weights
compute_HRP_weights(cov, res_order)
```

Snippet 10, Encoding the Recursive Bisection Process. Data Source: A personal elaboration on Python from: Bailey, D. H., & López de Prado, M. (2016).

2.4 The HRP Portfolio Optimization in Python: A practical Application

The HRP portfolio proposed by de Prado in his famous paper "*Building Diversified Portfolios that outperform out-of-sample"* in 2016, presents outstanding results in terms of robustness. Indeed, one of the main discoveries of de Prado, is that the machine learning based portfolio provides better risk performance indicators out-of-sample, thus outperforming the traditional portfolio allocations methodologies.

The following empirical application has the objective to replicate de Prado's findings by building on Python, an artificial correlation matrix *C* with several hierarchical clusters (*Snippet 11)*. After that, I sample time series from a normal distribution and implement the Hierarchical Risk Parity algorithm explained in the previous section on these time series[102]. The next analysis requires to compare the different portfolio optimization techniques' in-sample and out-of-sample volatilities. I will study the following portfolios respectively: *Hierarchical Risk Parity, 1/N uniform weighting*, *Naïve Risk Parity* and *Minimum Variance*. Besides that, following the same approach used by de Prado, I carry out an analysis on the out-of-sample Monte Carlo

[102] This practical application is based on the codes provided in: http://gautier.marti.ai/qfin/2018/10/02/hierarchical-risk-parity-part-1.html.

Simulations[103]. This further test is necessary since: *"However, the portfolio with minimum variance in-sample is not necessarily the one with minimum variance out-of-sample [….] instead, in this section we evaluate via Monte Carlo the performance out-of-sample of HRP against CLA's minimum-variance and traditional risk parity's IVP allocations. This will also help us understand what features make a method preferable to the rest, regardless of anecdotal counter-examples"*[104]. *Figures 38-40* show the different correlation and covariance matrices.

```
# build a hierarchical block diagonal correlation matrix
nb_alphas = 250
nb_observations = int(0.3 * 252)
quality = 0.6 * np.ones((nb_alphas // 6, nb_alphas // 6))
value = 2.4 * np.ones((nb_alphas // 2, nb_alphas // 2))
momentum = 2.6 * np.ones((int(nb_alphas * (1 - 1/6 - 1/2) + 1),
                    int(nb_alphas * (1 - 1/6 - 1/2) + 1)))
correl_mom_value = -1.2 * np.ones((int(nb_alphas * (1 - 1/6)) + 1,
                          int(nb_alphas * (1 - 1/6)) + 1))

correl = (block_diag(quality, correl_mom_value) +
        block_diag(quality, momentum, value)) / 3
np.fill_diagonal(correl, 1)
mean_returns = np.zeros(nb_alphas)
volatilities = ([np.sqrt(0.1 / np.sqrt(252))] * (nb_alphas // 3) +
            [np.sqrt(0.3 / np.sqrt(252))] * (nb_alphas - nb_alphas // 3 - nb_alphas // 6)
+
            [np.sqrt(0.5 / np.sqrt(252))] * (nb_alphas // 6))
```

[103] Monte Carlo methods, or Monte Carlo experiments, are a large class of mathematical algorithms that are based on repeated random sampling o get numerical results. The underlying rationale is to use randomness to solve problems that might be deterministic in principle. For a deeper study, see: https://en.wikipedia.org/wiki/Monte_Carlo_method.
[104] de Prado, (2016), Op. cit., pp. 159-169.

```
covar = np.multiply(correl,
                    np.outer(np.array(volatilities),
                             np.array(volatilities)))
covar = pd.DataFrame(covar)
```

Snippet 11, Building an artificial correlation matrix C with several hierarchical clusters. Data Source: A personal elaboration on Python based on: http://gautier.marti.ai/qfin/2018/10/02/hierarchical-risk-parity-part-1.html.

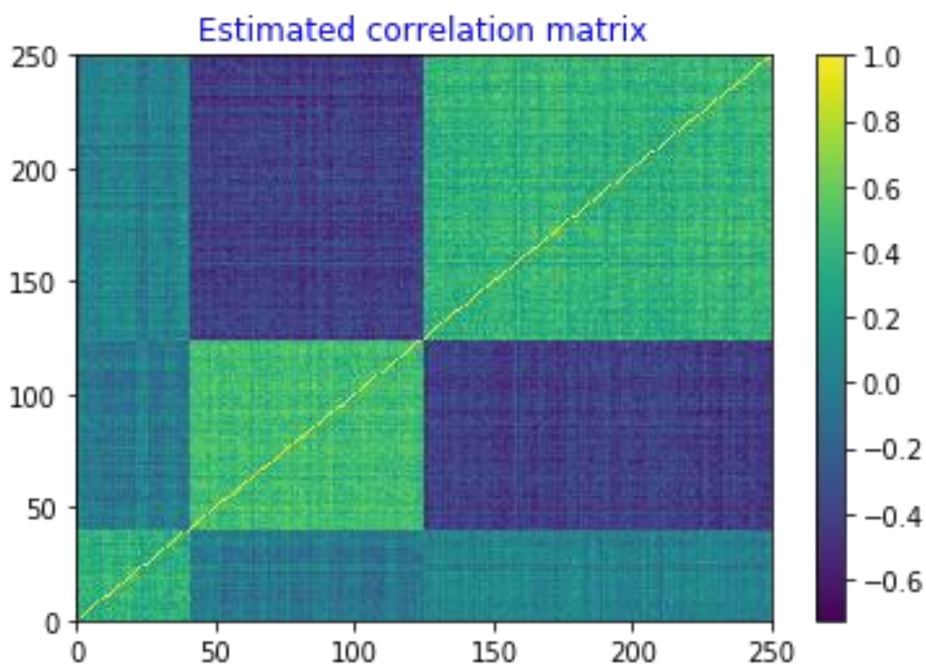


Figure 38, Estimated HRP correlation matrix. Data Source: A personal elaboration on Python based on: http://gautier.marti.ai/qfin/2018/10/02/hierarchical-risk-parity-part-1.html.

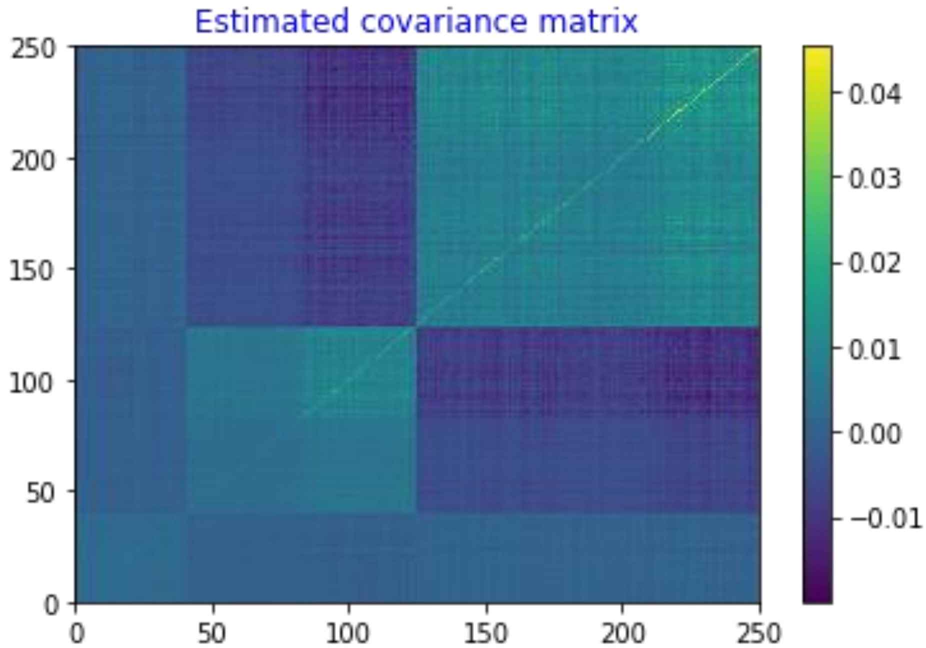


Figure 39, Estimated HRP covariance matrix. Data Source: A personal elaboration on Python based on: http://gautier.marti.ai/qfin/2018/10/02/hierarchical-risk-parity-part-1.html.

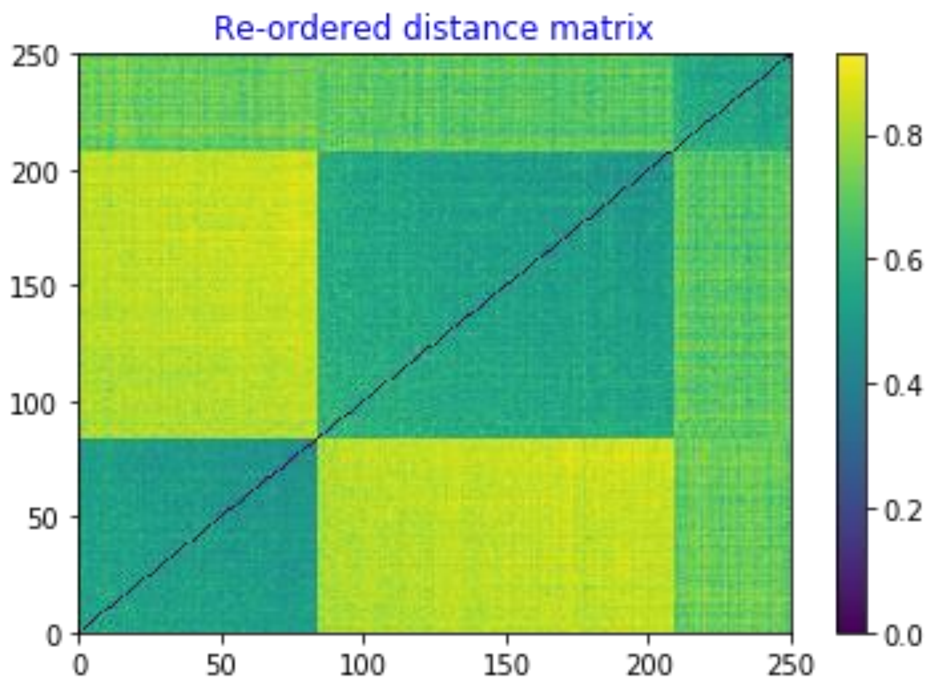


Figure 40, Quasi-Diagonalization. Data Source: A personal elaboration on Python based on: http://gautier.marti.ai/qfin/2018/10/02/hierarchical-risk-parity-part-1.html.

The comparative risk analysis is then reported in *Figure 41*, which show both the in-sample and out-of-sample volatilities for each portfolio optimization model taken into consideration[105]. While the validation in-sample shows that the Minimum Variance portfolio outperforms the others in terms of volatility minimization, the validation out-of-sample returns more robust results for the HRP portfolio. Indeed, "*the Minimum Variance portfolio yields overfitted solutions that do not produce out-of-sample performing portfolios*"[106]. However, a deeper Monte Carlo simulation study is required; indeed the naïve Risk Parity portfolio still exhibits a lower standard deviation than the HRP portfolio. For the Monte Carlo analysis: *"synthetic returns are drawn from a centered Gaussian parameterized by a random covariance matrix, where the variances are sampled from a multimodal distribution and the underlying correlation matrix from the uniform distribution over the space of correlation matrices using the onion method"*[107]. The simulations allow to better understand whether the different portfolio optimization techniques provide stable results in and out-of-sample. *Figures 42-44* at the end of the chapter show, how the HRP volatilities in and out-of-sample distribution is pretty stable compared with the Minimum Variance Portfolio. Although the Risk Parity portfolio seems to provide robust results, *"it is very likely that the sampled correlation matrices do not verify the stylized facts of empirical financial correlations matrices"*[108]. *Snippet 12* reported below, reports the Python codes used to perform the Monte Carlo simulations.

Portfolio Models	In-Sample	Out-of-Sample
HRP	0,41	0,38
1/N uniform weighting	0,44	0,45
Naïve Risk Parity	0,32	0,29
Minimum Variance	0,00	7,71

Figure 41, In-sample & out-of-sample volatility analysis. Data Source: A personal elaboration on Python.

[105] The out-of-sample analysis is based simulating the time series of returns over a two-year horizon.
[106] For a deeper analysis see: http://gautier.marti.ai/qfin/2018/10/02/hierarchical-risk-parity-part-1.html.
[107] For a deeper analysis see: https://gmarti.gitlab.io/qfin/2018/10/15/hierarchical-risk-parity-part-2.html.
[108] For a deeper analysis see: https://gmarti.gitlab.io/qfin/2018/10/15/hierarchical-risk-parity-part-3.html.

```
#Setting the desired methods
methods = {
    'Minimum Variance': compute_MV_weights,
    'Risk Parity': compute_RP_weights,
    'Hierarchical Risk Parity': compute_HRP_weights,
}
empirical_volatilities = {method: {'in-sample' : [], 'out-sample': []}
                          for method in methods.keys()}
#Performing the Monte Carlo experiments
nb_experiments = 2000
for experiment in tqdm(range(nb_experiments)):
    true_covariances = sample_cov_matrix(500)
    in_sample = generate_returns_sample(
        true_covariances, horizon=3 * 252)
    out_sample = generate_returns_sample(
        true_covariances, horizon=3 * 252)

    for name, method in methods.items():
        in_sample_weights = method(in_sample.cov())

        in_sample_vol = compute_portfolio_volatility(
            in_sample_weights, in_sample)

        out_sample_vol = compute_portfolio_volatility(
            in_sample_weights, out_sample)
```

Snippet 12, Encoding the Monte Carlo simulations. Data Source: A personal elaboration on Python based on: http://gautier.marti.ai/qfin/2018/10/02/hierarchical-risk-parity-part-2.html.

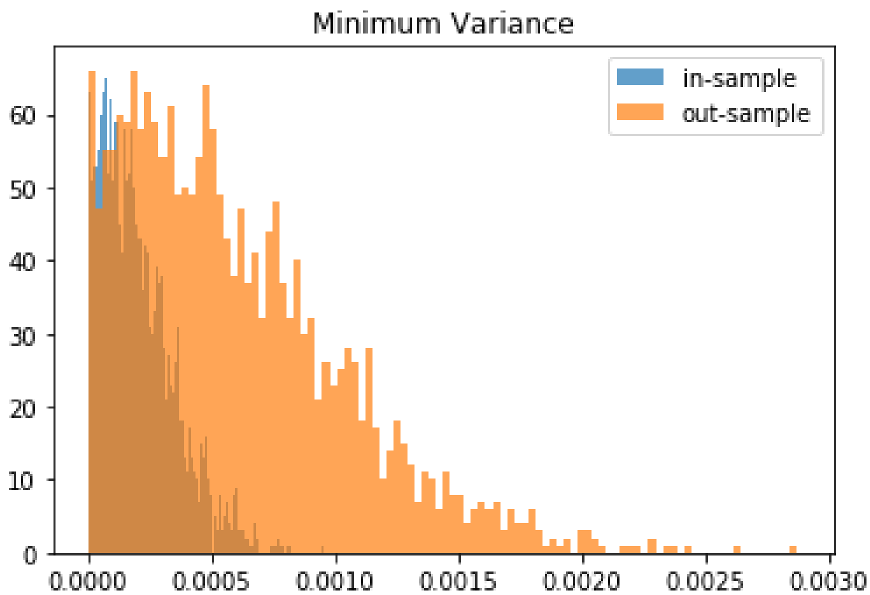


Figure 42, MV volatilities distribution. Data Source: A personal elaboration on Python based on: [http://gautier.marti.ai/qfin/2018/10/02/hierarchical-risk-parity-part-2.html.](http://gautier.marti.ai/qfin/2018/10/02/hierarchical-risk-parity-part-2.html)

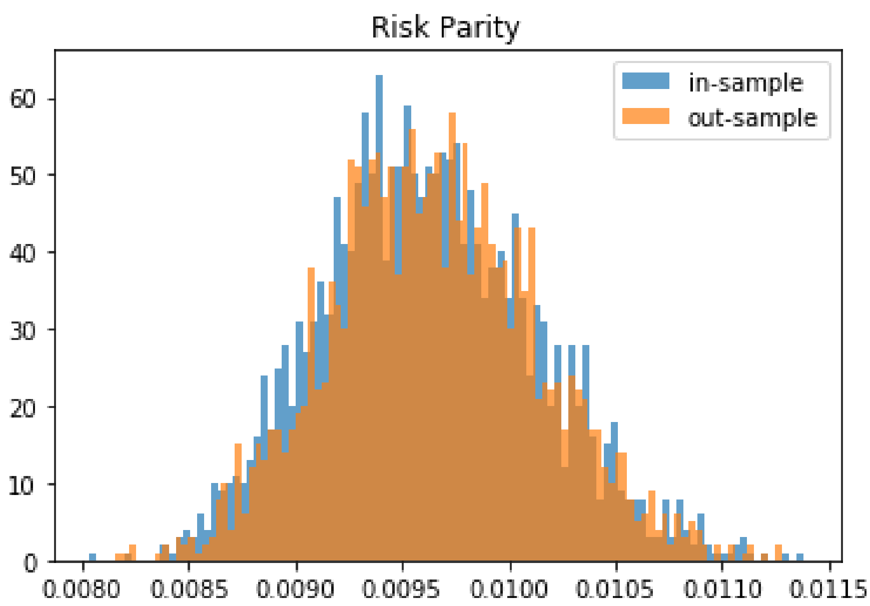


Figure 43, RP volatilities distribution. Data Source: A personal elaboration on Python based on: [http://gautier.marti.ai/qfin/2018/10/02/hierarchical-risk-parity-part-2.html.](http://gautier.marti.ai/qfin/2018/10/02/hierarchical-risk-parity-part-2.html)

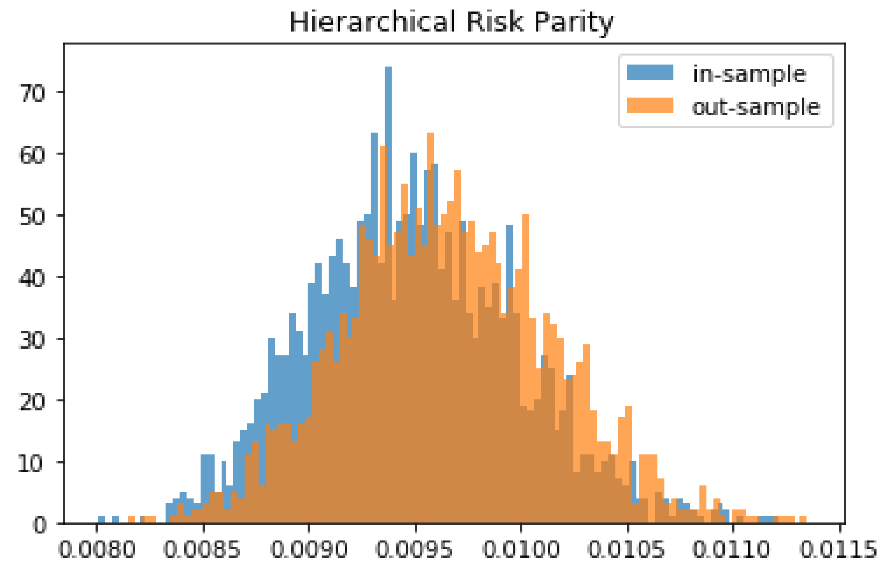


Figure 44, HRP volatilities distribution. Data Source: A personal elaboration on Python based on: [http://gautier.marti.ai/qfin/2018/10/02/hierarchical-risk-parity-part-2.html.](http://gautier.marti.ai/qfin/2018/10/02/hierarchical-risk-parity-part-2.html)

3. EMPIRICAL HIERARCHICAL RISK PARITY PORTFOLIO ANALYSIS

3.1 Data Description

For the empirical analysis I have decided to build a market index-based portfolio, the Dow Jones Industrial Average Stocks portfolio (DJIA) and another one made of the most liquid ETFs tracking the major index in US markets in the last decade. The rationale behind my choice is to investigate whether a machine learning portfolio can achieve a better result in terms of risk and performance metrics than the traditional portfolio risk models; the DJIA index portfolio can be used as a good benchmarking tool for the analysis.

3.1.1 The choice of the index: the Dow Jones Industrial Average

When the average investor thinks about stock markets indices, the first most prominent names that come up to her mind are the biggest U.S. market indices, namely Dow Jones, S&P 500, Nasdaq 100 and few others.

In the analysis, I have decided to look in depth of one of those indices given their overall reliability of data, and their impact on the global stock market rather than just the US one.

The S&P 500 is the broadest measure of the US economy; the index value is calculated by weighting each company according to its market capitalization and then a divisor, which is set by S&P, is applied to produce the final value.

The Nasdaq 100 is the youngest of the three abovementioned indices having begun trading in 1985. It represents the largest non-financial companies listed on the Nasdaq exchange and is generally regarded as a technology index due the heavy weighting given to tech-based companies. Similarly, to the S&P 500 index, the Nasdaq 100 is based on the market capitalization of its components.

On the other hand, The Dow Jones Industrial Average, often referred in short as the ‘Dow’, is the oldest index, dating back to 1896 and is the most globally well known. The Dow represents 30 large cap stocks as determined by the Wall Street Journal[109]. Unlike the S&P 500 and the

[109] The DJIA covers only companies with a large capitalization and high liquidity.

Nasdaq 100, the weighting for each component in the Dow Jones Industrial Average is ranked according to the share price, and then a divisor applies to create the final value[110].

Therefore, the choice of the Dow 30 index for the analysis is motivated both because it is computed in a different way and also because it presents a diversified composition across several sectors which makes it more interesting for creating different scenarios.

Figure 45-47 report the figures and pie charts that illustrate the different Dow Jones 30 constituents, their market capitalization as of today, their reference sectors and industries.

The adjusted close prices, the cumulative returns, the portfolio return distribution, the return correlation matrix and the descriptive statistics are reported in *Figures 48-52*. The time period considered, is the one going from 01/01/2012, to 01/01/2019[111].

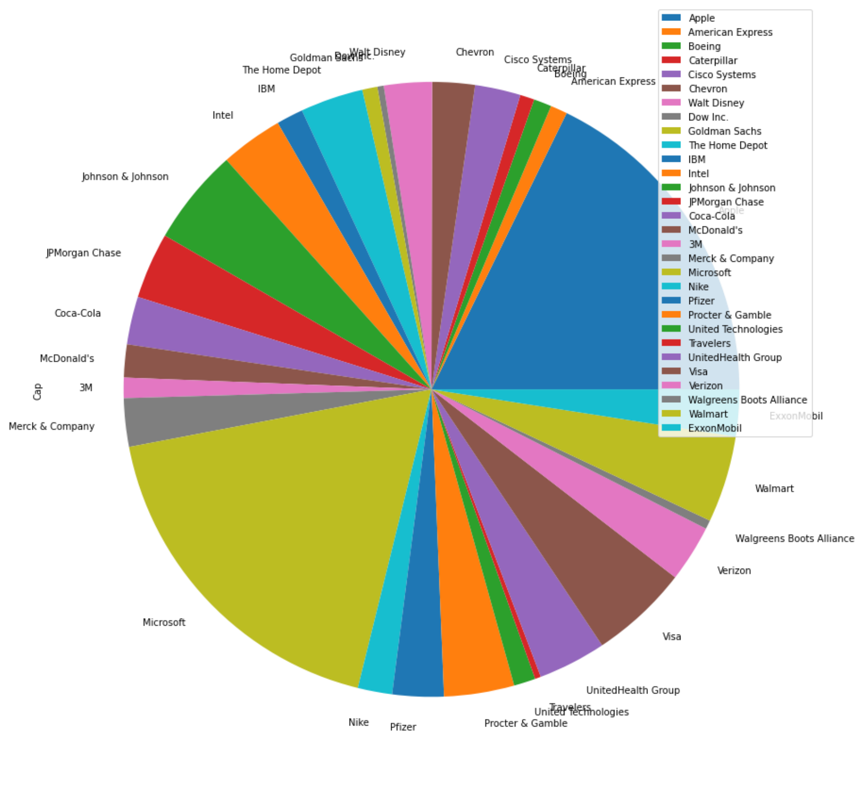


Figure 45, Pie Chart Market Cap Analysis. Data Source: A personal elaboration on Python.

[110] The value of the index is the sum of the price of one share of stock for each component company divided by a factor which changes whenever one of the component stocks has a stock split or a stock dividend, so as to generate a consistent value for the index. For a deeper study, see: https://en.wikipedia.org/wiki/Dow_Jones_Industrial_Average.

[111] This period covers the US business cycle expansions and contractions as defined by the National Bureau of Economic Research. For a deeper study, see: https.://www.nber.org/cycles.html.

Ticker	**Company**	**Sector**	**Industry**
AAPL	Apple	Consumer Goods	Electronic Equipment
AXP	American Express	Financial	Consumer Financial Services
BA	Boeing	Capital Goods	Aerospace & Defense
CAT	Caterpillar	Capital Goods	Construction & Agriculture Machinery
CSCO	Cisco Systems	Technology	Networking & Communication Devices
CVX	Chevron	Energy	Oil & Gas - Integrated
DIS	Disney	Services	Broadcasting
DOW	Dow	Basic Materials	Chemicals
GS	Goldman Sachs	Financial	Investment Brokerage - National
HD	Home Depot	Services	Retail (Home Improvement)
IBM	International Business Machines	Technology	Computer Hardware
INTC	Intel	Technology	Semiconductors
JNJ	Johnson & Johnson	Healthcare	Major Drugs
JPM	JP Morgan Chase	Financial	Money Center Banks
KO	Coca-Cola	Consumer/Non-Cyclical	Beverages (Non-Alcoholic)
MCD	McDonald's	Services	Restaurants
MMM	3M	Conglomerates	Conglomerates
MRK	Merck	Healthcare	Major Drugs
MSFT	Microsoft	Technology	Software & Programming
NKE	Nike	Consumer Goods	Textile - Apparel Footwear & Accessories
PFE	Pfizer	Health Care	Major Drugs
PG	Procter & Gamble	Consumer/Non-Cyclical	Personal & Household Products
TRV	The Travelers Companies	Financial	Property & Casualty Insurance
UNH	Unitedhealth Group	Healthcare	Health Care Plans
UTX	United Technologies	Conglomerates	Conglomerates
V	Visa	Financial	Credit Services
VZ	Verizon	Services	Communications Services
WBA	Walgreens	Services	Pharmaceutical Retailers
WMT	Wal-Mart	Services	Retail (Department & Discount)
XOM	ExxonMobil	Energy	Oil & Gas - Integrated

Figure 46, Dow Jones Industrial Average Stocks composition. Data Source: A personal elaboration on Excel of yahoo-Finance data.

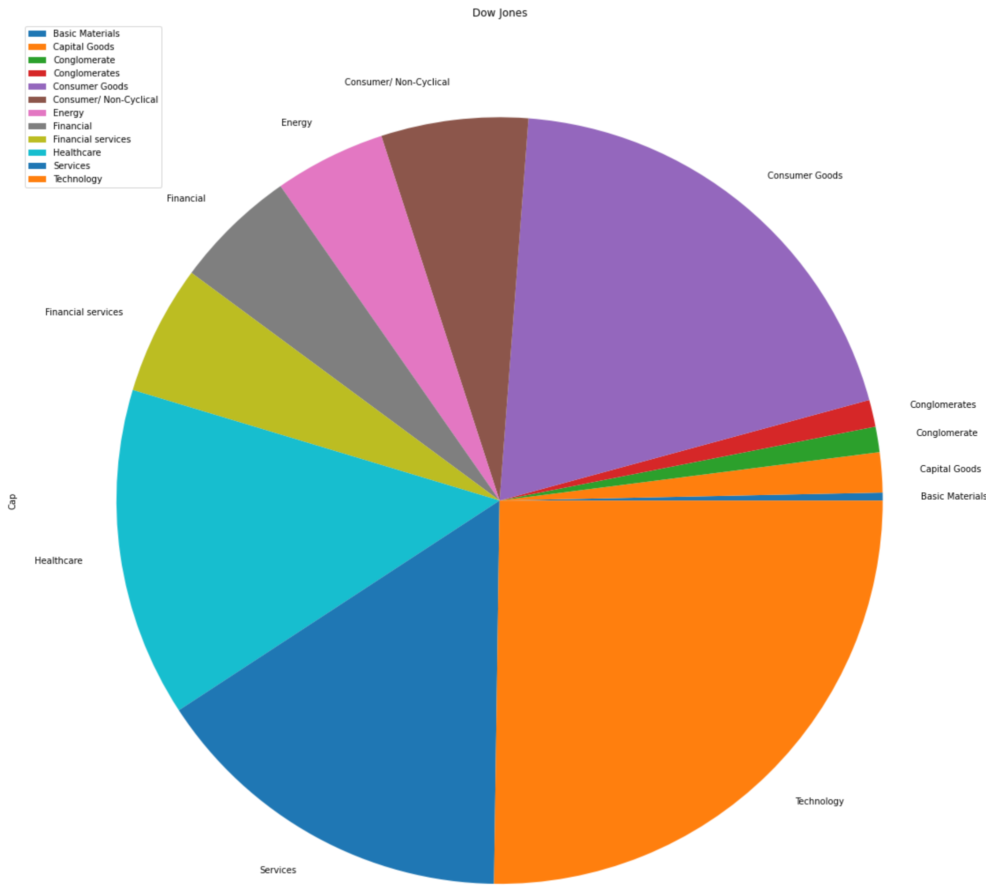


Figure 47, Dow Jones Industrial Average Sector composition. Data Source: A personal application on Python from Bloomberg.

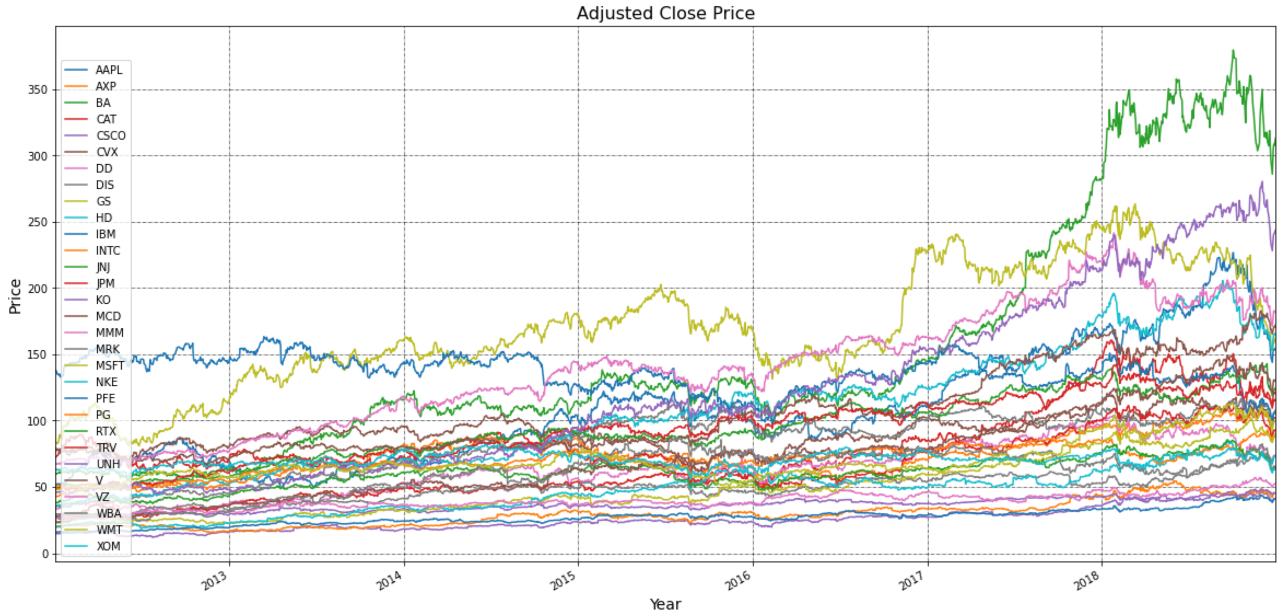


Figure 48, Dow Jones Industrial Average Adjusted Close Price. Data Source: A personal elaboration on Python from Yahoo-Finance.

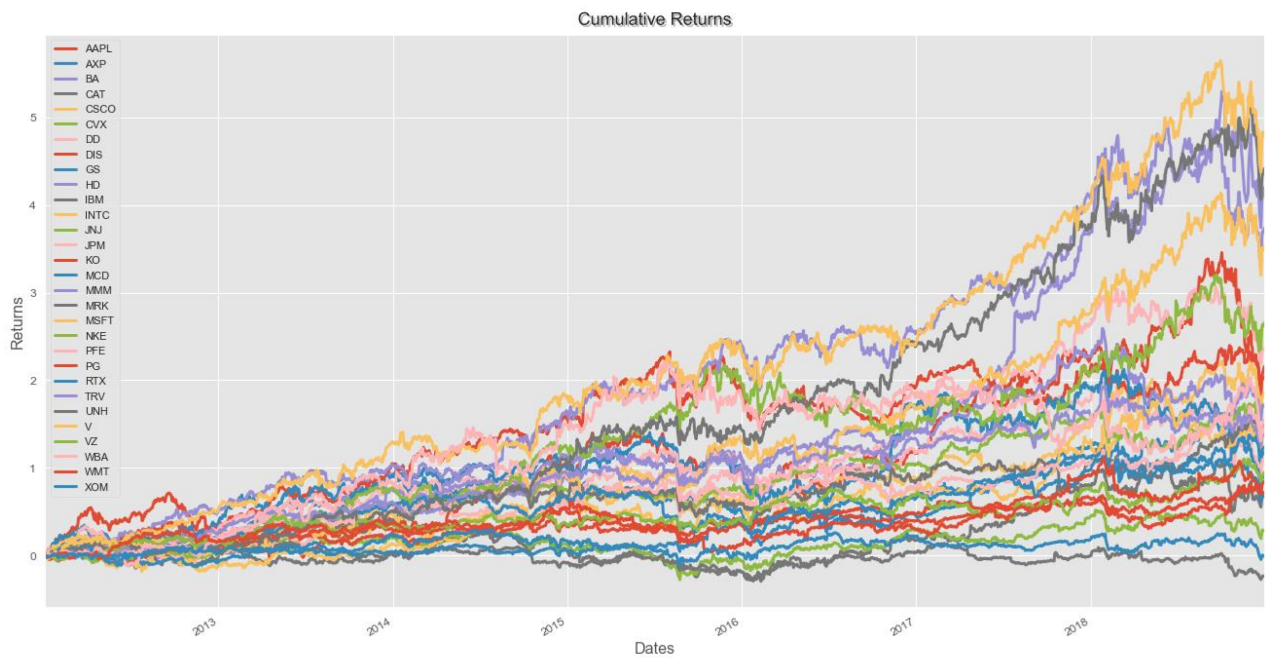


Figure 49, Dow Jones Industrial Average Cumulative Returns. Data Source: A personal elaboration on Python from Yahoo-Finance.

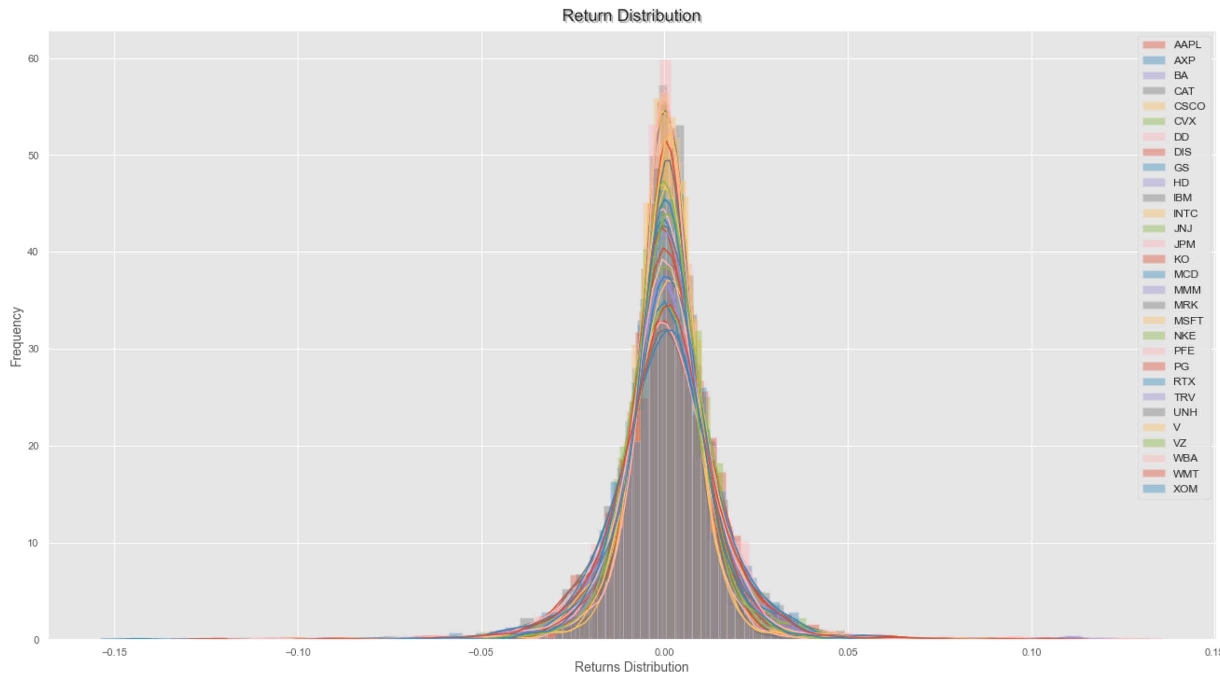


Figure 50, Dow Jones Industrial Average Return Distribution. Data Source: A personal elaboration on Python from Yahoo-Finance.

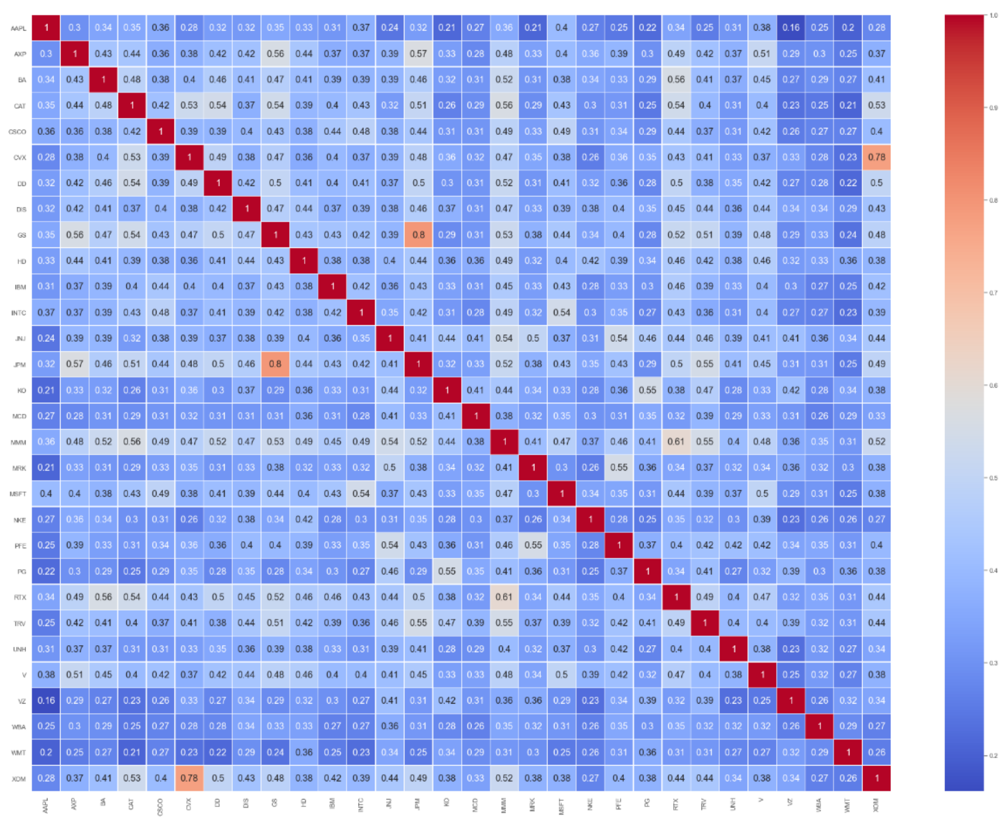


Figure 51, Dow Jones Industrial Average Correlation Matrix. Data Source: A personal elaboration on Python.

Index	Annualized Return (%)	Annualized Volatility (%)	Sharpe Ratio
V	27,461	21,062	1,304
UNH	26,315	20,573	1,279
BA	26,126	22,408	1,166
MSFT	24,306	23,071	1,054
HD	23,969	18,407	1,302
NKE	21,225	23,161	0,916
JPM	19,802	21,802	0,908
AAPL	19,171	25,459	0,753
DIS	18,210	18,626	0,978
CSCO	17,554	22,422	0,783
MMM	15,688	16,520	0,950
WBA	15,602	24,308	0,642
INTC	15,219	23,646	0,644
MRK	14,954	18,638	0,802
PFE	14,794	16,689	0,886
TRV	13,821	16,089	0,859
JNJ	13,631	14,746	0,924
DD	13,601	24,413	0,557
AXP	13,212	20,251	0,652
MCD	12,764	15,578	0,819
GS	12,152	23,353	0,520
VZ	10,935	16,561	0,660
CAT	10,532	24,623	0,428
WMT	10,417	18,129	0,575
RTX	9,023	17,742	0,509
PG	8,866	14,722	0,602
KO	8,584	13,909	0,617
CVX	5,658	20,330	0,278
XOM	1,443	17,462	0,083
IBM	-1,933	19,246	-0,100

Figure 52, Dow Jones Industrial Average Descriptive Statistics. Data Source: A personal elaboration on Excel.

3.1.2 The all ETFs portfolio

The second portfolio I have constructed for the final analysis, consists of the 15 most liquid ETFs tracking the major index in the US markets. I have concentrated my attention on an all ETFs portfolio due to the great importance these index-based funds cover in the financial markets nowadays. Indeed, introduced 39 years ago, ETFS are now one of the fastest-growing asset types in the financial markets, reaching the total assets under management to almost US$6 trillion by the end of 2019[112]. Furthermore, these instruments are widely used by financial advisors and retail investors to equitize cash, undertake diversified investments and implement tactical adjustments to portfolios[113].

As for the previous section, the following figures, report the tickers, asset class and the corresponding tracked index for each ETF, as well as the returns analysis and the constituent's descriptive statistics (*Figures 53)*. The daily adjusted close prices are downloaded from the Yahoo-Finance database from 01/01/2012 to 01/01/2019.

Ticker	Name	Asset Class	Index
QQQ	Invesco QQQ Trust, Series 1	Equity	NASDAQ-100 Index
IWF	iShares Russell 1000 Growth ETF	Equity	Russell 1000 Growth Index
SPY	SPDR S&P 500	Equity	S&P 500 Index
VTI	Vanguard Total Stock Market ETF	Equity	CRSP US Total Market Index
OEF	iShares S&P 100 ETF	Equity	S&P 100 Index
DIA	SPDR Dow Jones Industrial Avera	Equity	Dow Jones Index
IWO	iShares Russell 2000 Growth ETF	Equity	Russell 2000 Growth Index
IJH	iShares Core S&P Mid-Cap ETF	Equity	S&P MidCap 400 Index
MDY	SPDR MidCap Trust Series I	Equity	S&P MidCap 400 Index
IWM	iShares Russell 2000 ETF	Equity	Russell 2000 Index
IWD	iShares Russell 1000 Value ETF	Equity	Russell 1000 Value Index
IWN	iShares Russell 2000 Value ETF	Equity	Russell 2000 Value Index
FEZ	SPDR DJ Euro STOXX 50 Etf	Equity	EURO STOXX 50 Index
PFF	iShares US Preferred Stock ETF	Fixed Income	CE Exchange-Listed Preferred & Hybrid Securities Index
DBC	Invesco DB Commodity Index Trac	Commodity	DBIQ Optimum Yield Diversified Commodity Index

Figure 53, ETFs Ticker list, Asset Class & Tracking Index. Data Source: A personal elaboration on Excel.

[112] For a deeper study, see: https://amers2.apps.cp.thomsonreuters.com/web/cms/?navid=45050.

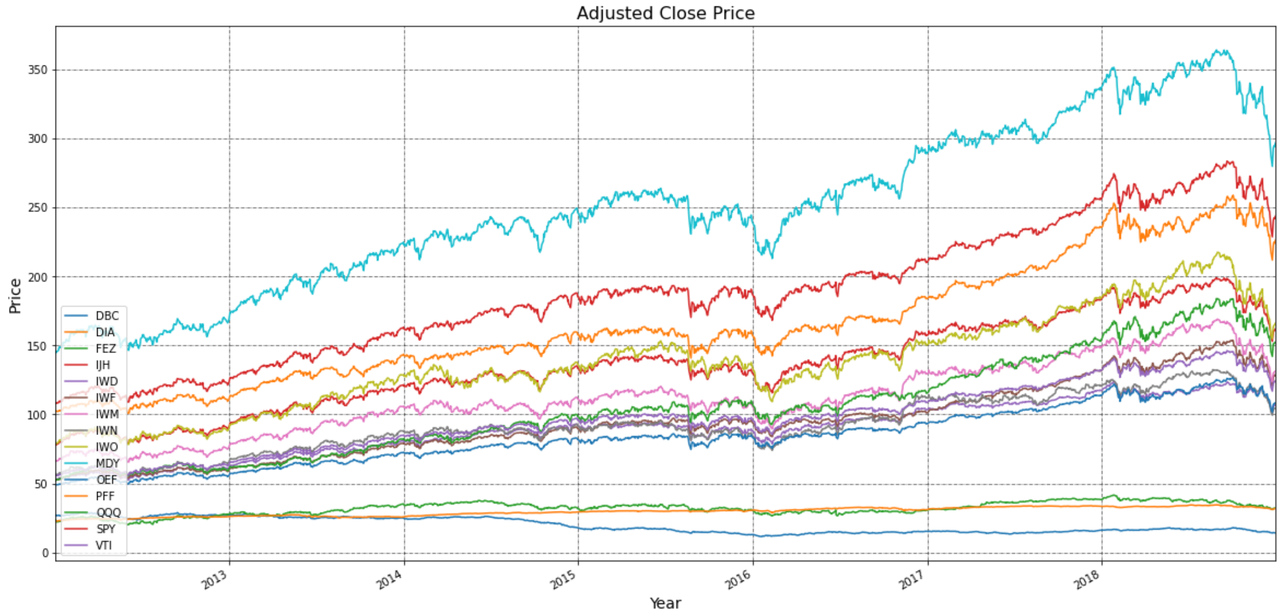


Figure 54, ETFs Adjusted Close Price. Data Source: A personal elaboration on Python from Yahoo-Finance.

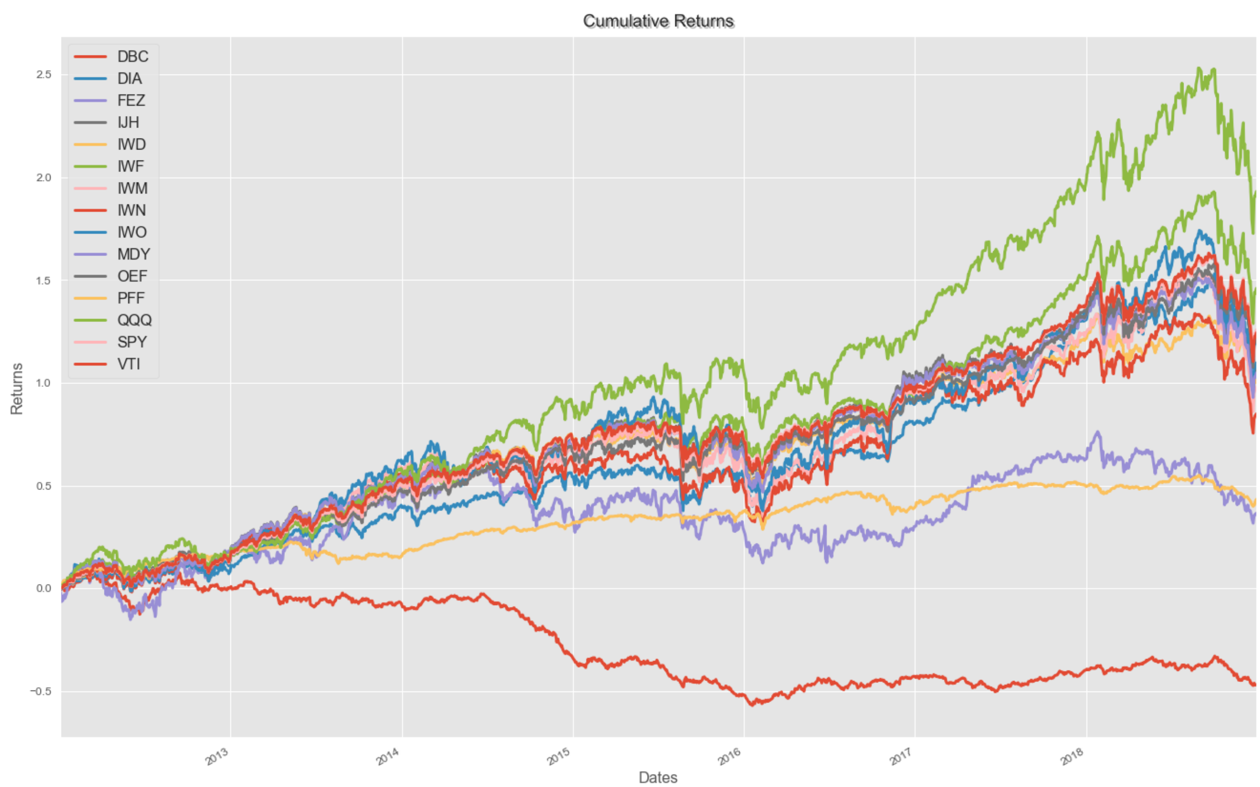


Figure 55, ETFs Cumulative Returns. Data Source: A personal elaboration on Python from Yahoo-Finance.

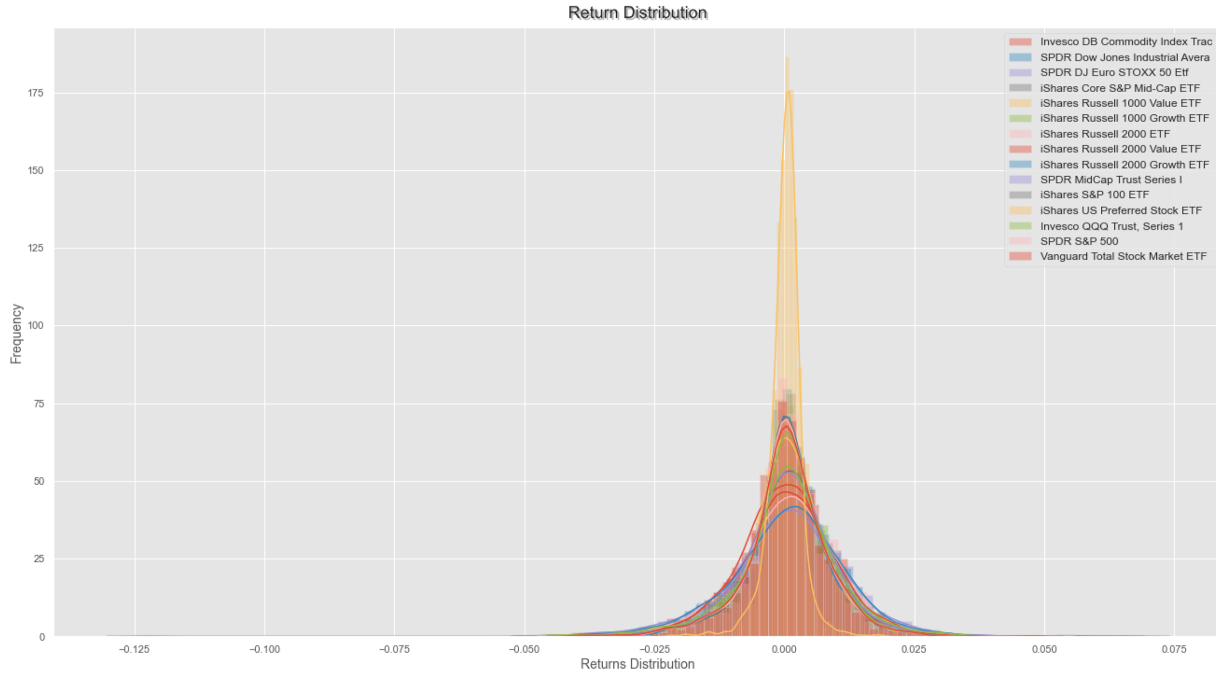


Figure 56, ETFs Return Distribution. Data Source: A personal elaboration on Python from Yahoo-Finance.

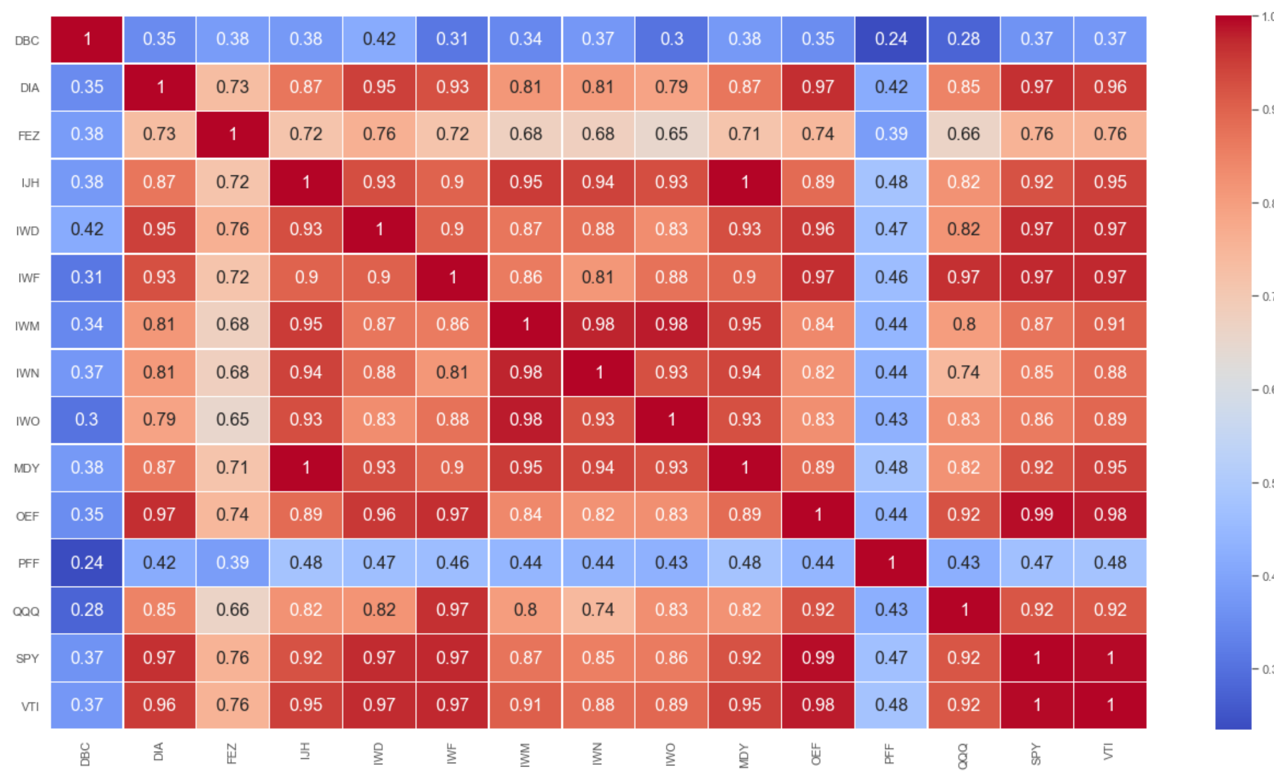


Figure 57, ETFs Correlation Matrix. Data Source: A personal elaboration on Python from Yahoo-Finance.

Index	Annualised Return (%)	Annualised Volatility (%)	Sharpe Ratio
Invesco QQQ Trust, Series 1	16,683	15,953	1,046
iShares Russell 1000 Growth ETF	13,832	13,657	1,013
SPDR S&P 500	12,489	12,833	0,973
Vanguard Total Stock Market ETF	12,418	12,973	0,957
iShares S&P 100 ETF	12,243	12,857	0,952
SPDR Dow Jones Industrial Avera	12,190	12,581	0,969
iShares Russell 2000 Growth ETF	12,136	17,437	0,696
iShares Core S&P Mid-Cap ETF	11,473	14,127	0,812
SPDR MidCap Trust Series I	11,287	14,218	0,794
iShares Russell 2000 ETF	11,069	16,071	0,689
iShares Russell 1000 Value ETF	10,879	12,666	0,859
iShares Russell 2000 Value ETF	9,973	15,255	0,654
SPDR DJ Euro STOXX 50 Etf	6,360	19,415	0,328
iShares US Preferred Stock ETF	5,290	5,125	1,032
Invesco DB Commodity Index Trac	-8,021	14,453	-0,555

Figure 58, ETFs Descriptive Statistics. Data Source: A personal elaboration on Excel.

3.2 Methodology

I set the stable period testing for a period of about 8 years, from 01/01/2012 to 01/01/2019, which is the usual time horizon of a business cycle. The main goal of the final analysis is to compare the two Hierarchical Risk Parity portfolios (the Dow Jones Index one and the one built with the fifteen ETFs), with other three traditional portfolio allocation approaches in terms of profitability, diversification and risk minimization. The three strategies used as a good benchmarking tool for the final test are respectively: The Minimum Variance portfolio (MV) or CLA, the Inverse Variance portfolio (IV) and the Equal Weighted portfolio (EW) [114]. I have also included a random weighted portfolio (RDM) where the weights are randomly chosen according to the total number of simulations, which I set equal to 10000. According to several studies indeed, random portfolios have shown to outperform their benchmarks, exhibiting higher returns and Sharpe ratios[115].

I will conduct the final test both in-sample and out-of-sample, because I want to investigate whether the robustness of the HRP portfolio changes in the two testing periods. This is in line with de Prado's findings, according to which the HRP portfolio usually outperforms the

[114] For a deeper understanding of the strategies refer to: Chapter 1 (1.3.3, 1.3.7) and Chapter 2 (2.2).

[115] Burns, P. (2007). Random portfolios for performance measurement. In *Optimisation, Econometric and Financial Analysis* (pp. 227-249). Springer, Berlin, Heidelberg.

traditional methodologies out-of-sample[116]. I set the in-sample period from 01/01/2012 to 01/01/2016, while the out-of-sample testing period goes from 01/01/2016 to 01/01/2019. In the latter period, I adjust the portfolio allocation on the first trading day every month[117].

I have personally encoded all the analysis on the Anaconda Jupyter notebook of Python (vv.3.7)[118]. *Snippet 13* reports the codes for initializing the different allocation methods and finding their weights. *Snippet 14* shows the in-sample and out-of-sample tests.

```
#-----------------------------------------------------------------------------------------------------
#Initializing the allocation methods and getting the weights
def get_weight(self, data, model):
        if model == 'HRP':
            model_weight = self.get_HRP_weights(data)
        elif model == 'EW':
            model_weight = self.get_EW_weights(data)
        elif model == 'MVP':
            model_weight = self.get_MVP_weights(data)
        elif model == 'IVP':
            model_weight = self.get_IVP_weights(data)
        elif model == 'RDM':
            model_weight = self.get_RDM_weights(data)

    return model_weight
```

---

116 de Prado, (2016), Op. cit., pp.59-69.

117 The out-of-sample approach follows: https://github.com/KennnnyZhou/Hierarchical_Risk_Parity/blob/master/Paper%20for%20803%20Project.pdf.

118 The codes for the test have been personally modified from: https://github.com/KennnnyZhou/Hierarchical_Risk_Parity/blob/master/Paper%20for%20803%20Project.pdf; https://github.com/hudson-and-thames/research/blob/master/Chapter16/Chapter16.ipynb, https://github.com/robertmartin8/PyPortfolioOpt/blob/master/examples.py.

```
#-----------------------------------------------------------------------------------------------------------
#Calculating the weights for each portfolio method

def get_HRP_weights(self, price_data):
        HRP_result = get_HRP_result(price_data)
        return HRP_result[0]

    def get_EW_weights(self, price_data):
        N = price_data.shape[1]
        EW_weights = [1 / N] * N
        return pd.Series(EW_weights)

    def get_IVP_weights(self, price_data):
        return_data = price_data.pct_change().dropna()
        cov = return_data.cov().values
        ivp_weights = 1. / np.diag(cov)
        ivp_weights /= ivp_weights.sum()
        return pd.Series(ivp_weights)

    def get_MVP_weights(self, price_data):
        mvp_weights = MVP(price_data)[0]
        return mvp_weights

   def generateWgts(num):
       wgts = np.random.random(num)
      wgts /= wgts.sum()
      return wgts
```

Snippet 13, Initializing the portfolio allocation strategies. Data Source: A personal elaboration on Python.

```
#-------------------------------------------------------------------------------------------------------
    file = 'ETF_Index.csv'
    price_data = pd.read_csv(file, index_col=0)
    price_data.index = pd.to_datetime(price_data.index, format="%Y-%m-%d")
    models_list = ['IVP', 'HRP', 'EW', 'MVP', 'UMVP', 'RDM']

  ########### in-sample ############
    in_start_date = '2012-01-01'
    in_end_date = '2016-01-01'
    in_sample = price_data[(price_data.index >= in_start_date ) & (price_data.index < 
in_end_date)]
    in_test = in_test(in_sample, models_list)
    r, vol, weights = in_test.run_test(models_list)
    in_test.plot_SR()
    in_test.plot_r_vol()
    in_test.plot_frontier()

    ########### out-of-sample ##############
    out_start_date = '2012-01-01'
    out_end_date = '2019-01-01'
    out_sample = price_data[(price_data.index >= out_start_date ) & (price_data.index < 
out_end_date)]
    period = 60
    out_test = out_test(out_sample, period)
    out_r, out_weights = out_test.run_test(models_list)
    out_test.plot_cum_return()
    out_test.plot_SR()
    out_test.plot_ann_vol()
```

Snippet 14, In-Sample & Out-of-Sample Tests. Data Source: A personal elaboration on Python.

3.3 Empirical Results

In this section I will present the empirical results and descriptive statistics for all the different allocation approaches, using the adjusted close price of both the Dow Jones Industrial Average Index and the created all ETFs portfolio. For ease of reading, for the ETFs portfolio I will report only the in-sample and out-of-sample tests.

The Dow Jones Index Portfolio in-sample test results, reported below, show that the Random Weighted (RDM) portfolio is the one exhibiting the highest Sharpe ratio and annualized return (*Figure 59* and *Figure 60)*. The Markowitz Minimum Variance (MVP) portfolio on the other hand, is the one having the lowest annualized volatility. The Hierarchical Risk Parity (HRP) portfolio notably manages to strengthen its robustness in the out-of-sample test. Not only, in line with de Prado's findings, it achieves the lowest annualized volatility outperforming even the CLA solution, but it also achieves a very good Sharpe ratio result (*Figure 62* and *Figure 63*). In the out-of-sample test, the MV portfolio seems to be subject to random shocks, and therefore it is much more volatile as demonstrated by its cumulative return and annualized volatility's results.

As we can understand by analyzing the different portfolios weight allocation, the HRP one provides a more stable and robust algorithm in optimizing the portfolio construction (*Figure 66*). The CLA portfolio concentrates almost the 75% of the holdings on the top-4 investments and assigns a zero weight to most of the assets (19 out of 30) (*Figure 71*). The main rationale behind this result, is that the algorithm has the goal of minimizing the whole portfolio volatility, therefore it focuses on those assets exhibiting the lowest risk. The IV portfolio instead, has allocated the weights more uniformly across the portfolio, with a greater proportion assigned to the first 4-5 securities (*Figure 77*). The HRP portfolio seems to find a balance between the before mentioned portfolios. Indeed, it focuses its attention more on the top 4-5 assets, assigning the greater share to the Johnson & Johnson security. Nonetheless, unlike the MV portfolio, it allots its resources more evenly, giving a non-zero weight to all the constituents in the portfolio.

The HRP portfolio presents a very good Value at Risk (VaR) indicator for a $1M USD investment (*Figure 68* and *Figure 69*). The maximum portfolio loss the HRP portfolio can bear with 99% level of confidence, over a one-day period is 263961,11 USD, a level below both the IV and EW portfolios values. Only the MV one, consistent with its objective, achieves a better result, exhibiting a VaR of 249258,76 USD. Furthermore, from *Figure 70*, we can notice that the HRP portfolio has a very good drawdown measure. Compared to the other allocation methods, it recovers more easily from a negative trend of the stocks returns over a specified period of time. Since the MV portfolio has concentrated its attention on only a few securities, it is subject to much more negative impact by random shocks than HRP (*Figure 76*).

The all-ETFs portfolio (3.3.2) provides interesting result as well. For what concerns the weight distribution, the results are consistent with the Dow Jones Index portfolio. However, the in-sample and out-of-sample tests give different outputs. From the in-sample results, we notice that the MV portfolio exhibits the highest Sharpe ratio and the lowest annualized volatility (*Figure 89* and *Figure 90*). *Figure 91* reports the in-sample efficient frontier, that shows the HRP portfolio is the only one, together with the MV portfolio, to lie on the line. In the out-of-sample test, the HRP portfolio does not manage to beat the MV portfolio in terms of risk minimization, though it still achieves a very low level of volatility, but this time, it presents the highest level of Sharpe ratio compared to the other allocation approaches (*Figure 93* and *Figure 94*). The EW and RDM portfolios are the ones exhibiting the worst indicators out of sample.

3.3.1 The Dow Jones Index Portfolio Results

In-Sample Test

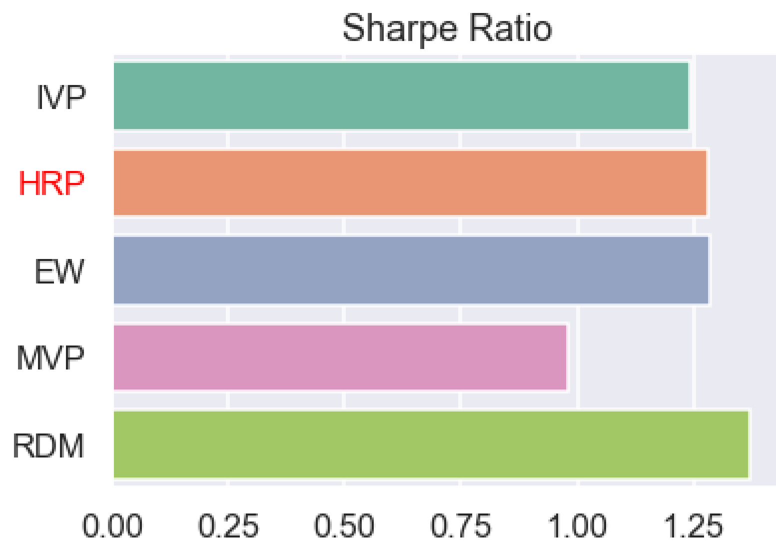


Figure 59, In-Sample Sharpe Ratio. Data Source: A personal elaboration on Python.

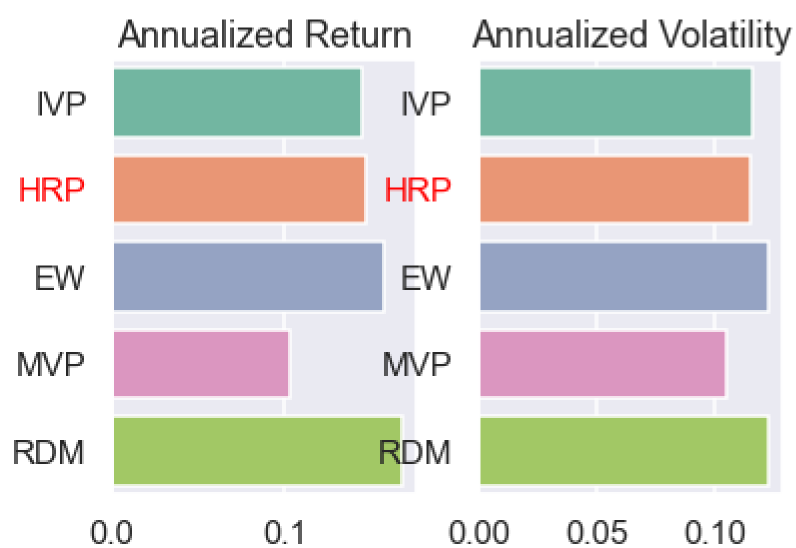


Figure 60, In-Sample Annualized Return & Volatility. Data Source: A personal elaboration on Python.

Out-of-Sample Test

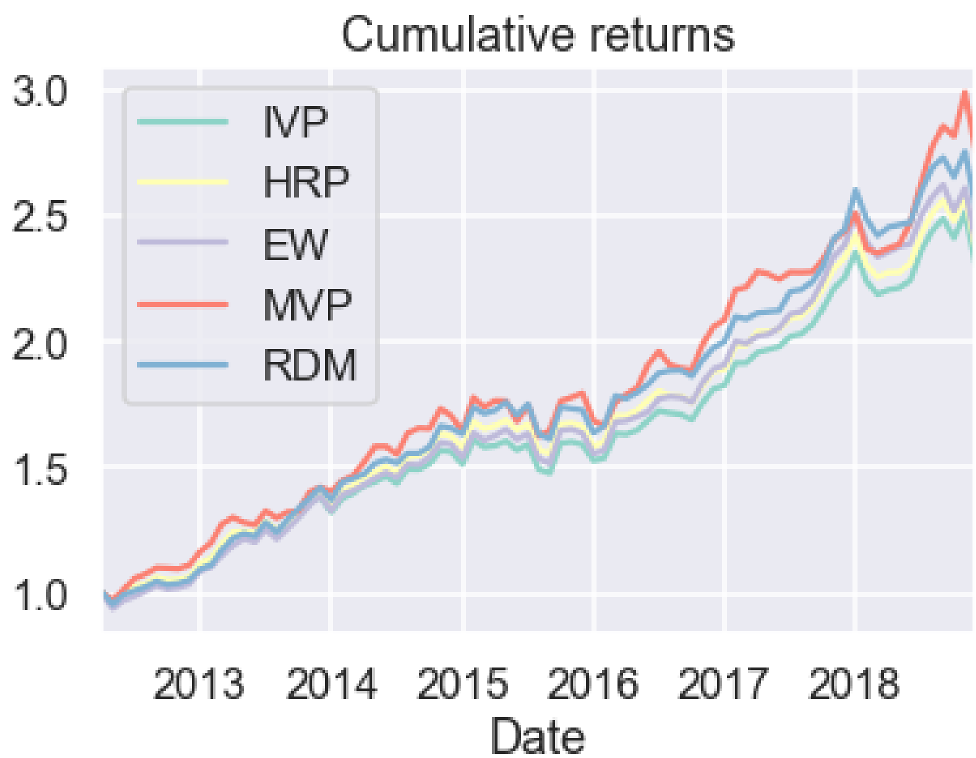


Figure 61, Out-of-Sample Cumulative Returns. Data Source: A personal elaboration on Python.

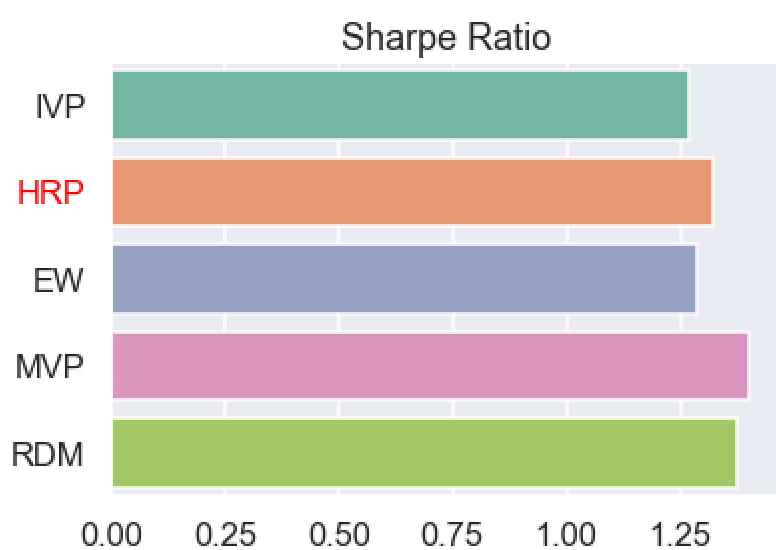


Figure 62, Out-of-Sample Sharpe Ratio. Data Source: A personal elaboration on Python.

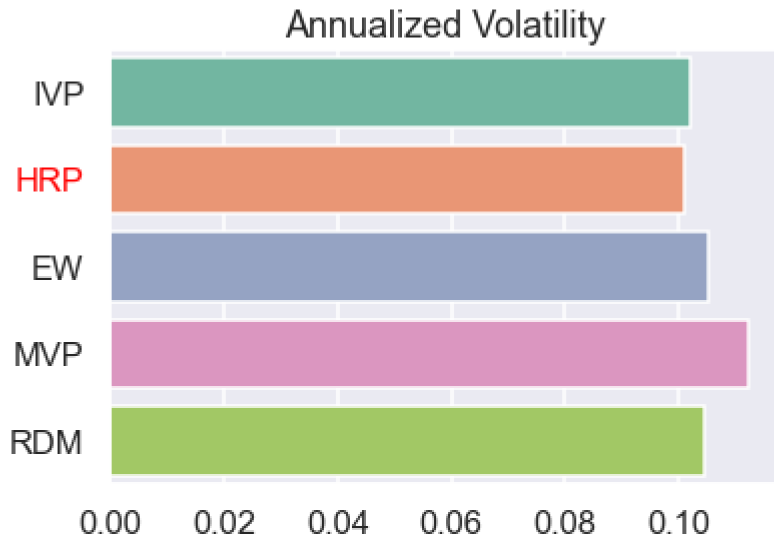


Figure 63, Out-of-Sample Annualized Volatility. Data Source: A personal elaboration on Python.

The HRP Portfolio

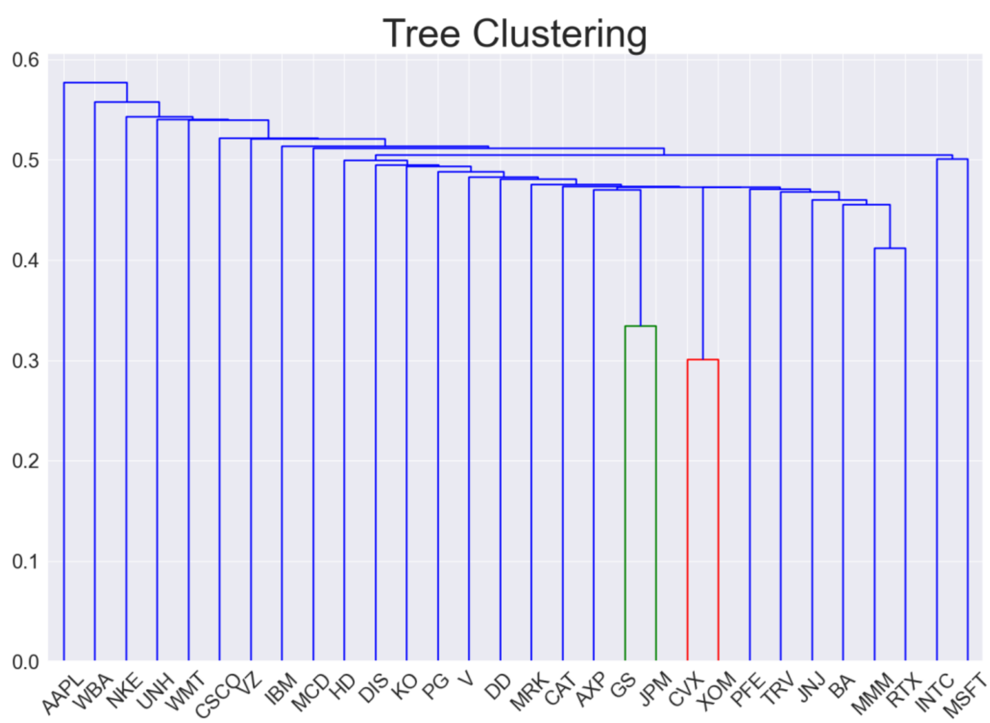


Figure 64, HRP Tree Clustering Process. Data Source: A personal elaboration on Python.

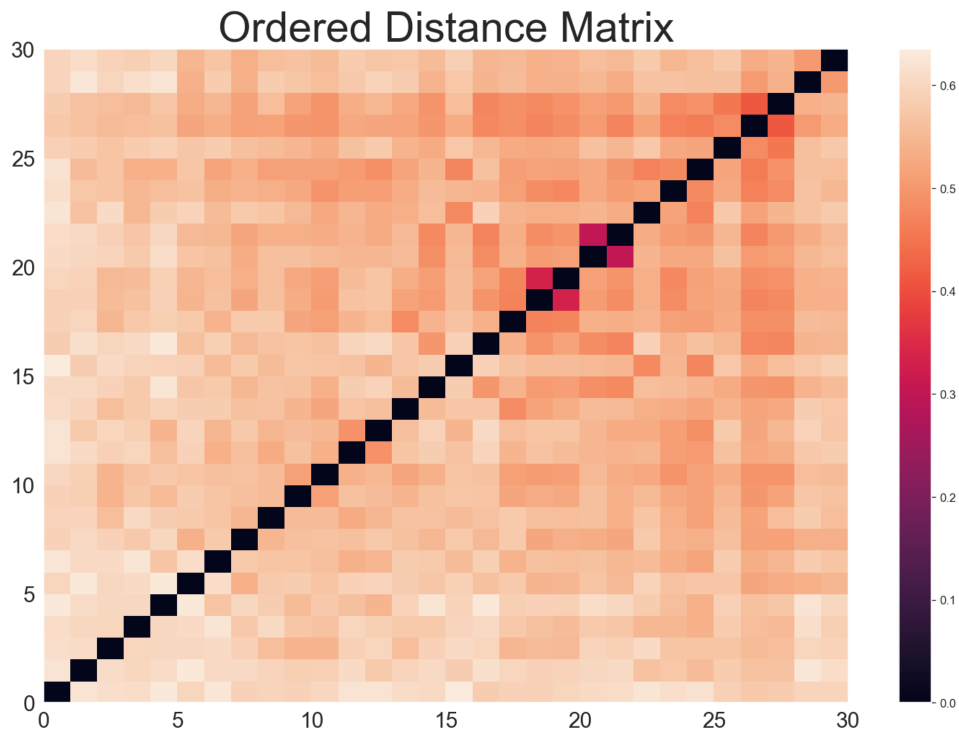


Figure 65, HRP Ordered Distance Matrix. Data Source: A personal elaboration on Python.

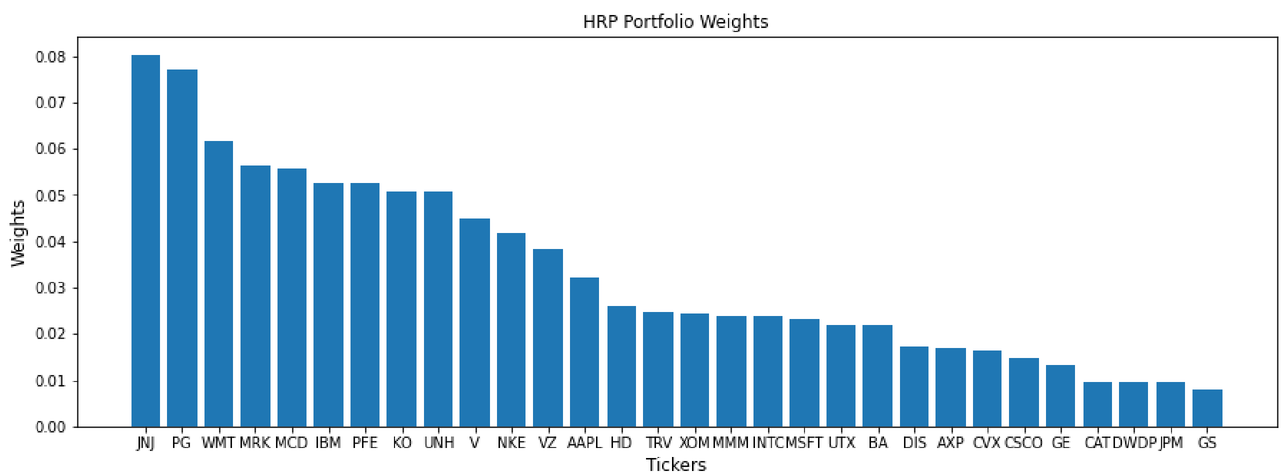


Figure 66, HRP Portfolio Weights. Data Source: A personal elaboration on Python.

Skew: -0,3260888447524245
Kurtosis: 3,922758068181932

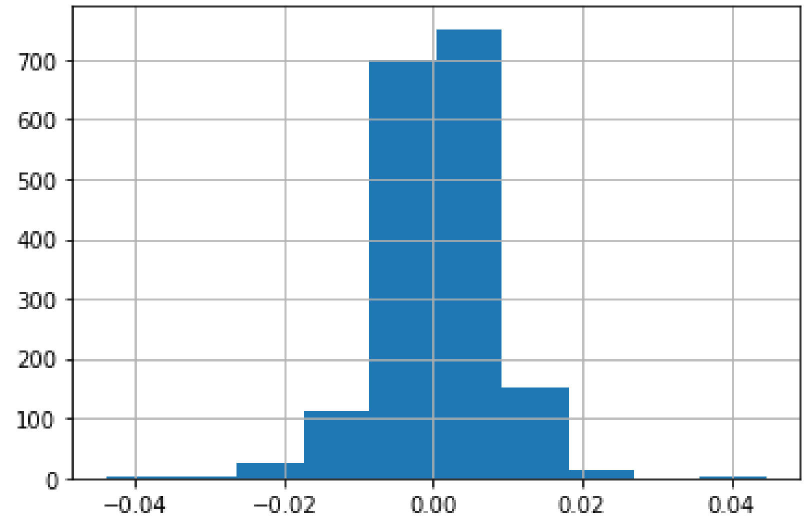


Figure 67, HRP Portfolio Returns Distribution. Data Source: A personal elaboration on Python.

N Day VaR Results:

1-day VaR @ 99% confidence: 263961,11

2-day VaR @ 99% confidence: 373297,38

3-day VaR @ 99% confidence: 457194,05

4-day VaR @ 99% confidence: 527922,22

5-day VaR @ 99% confidence: 590234,99

6-day VaR @ 99% confidence: 646570,03

7-day VaR @ 99% confidence: 698375,45

8-day VaR @ 99% confidence: 746594,76

9-day VaR @ 99% confidence: 791883,33

10-day VaR @ 99% confidence: 834718,32

11-day VaR @ 99% confidence: 875459,96

12-day VaR @ 99% confidence: 914388,11

13-day VaR @ 99% confidence: 951725,32

14-day VaR @ 99% confidence: 987652,04

15-day VaR @ 99% confidence: 1022316,98

Figure 68, N Day VaR Results. Data Source: A personal elaboration on Python.

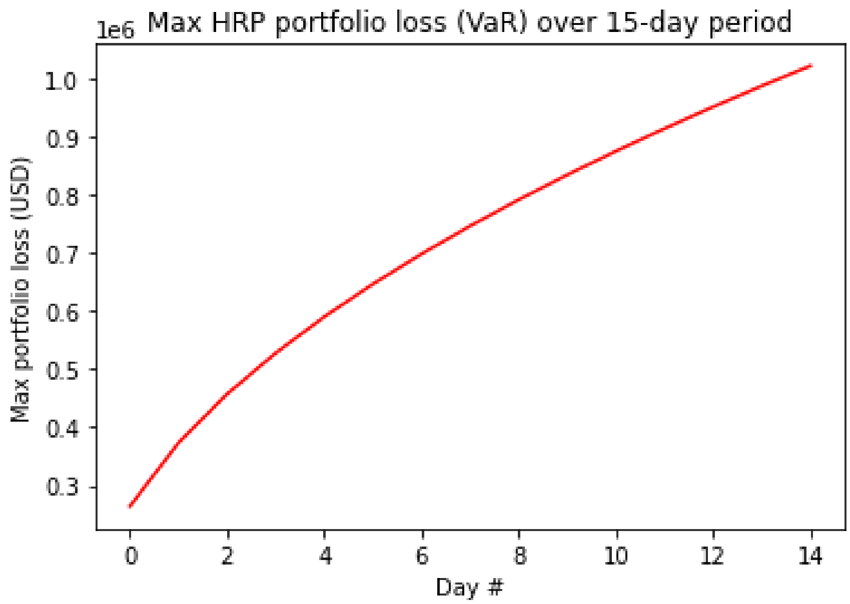


Figure 69, HRP VaR over 15-days. Data Source: A personal elaboration on Python.

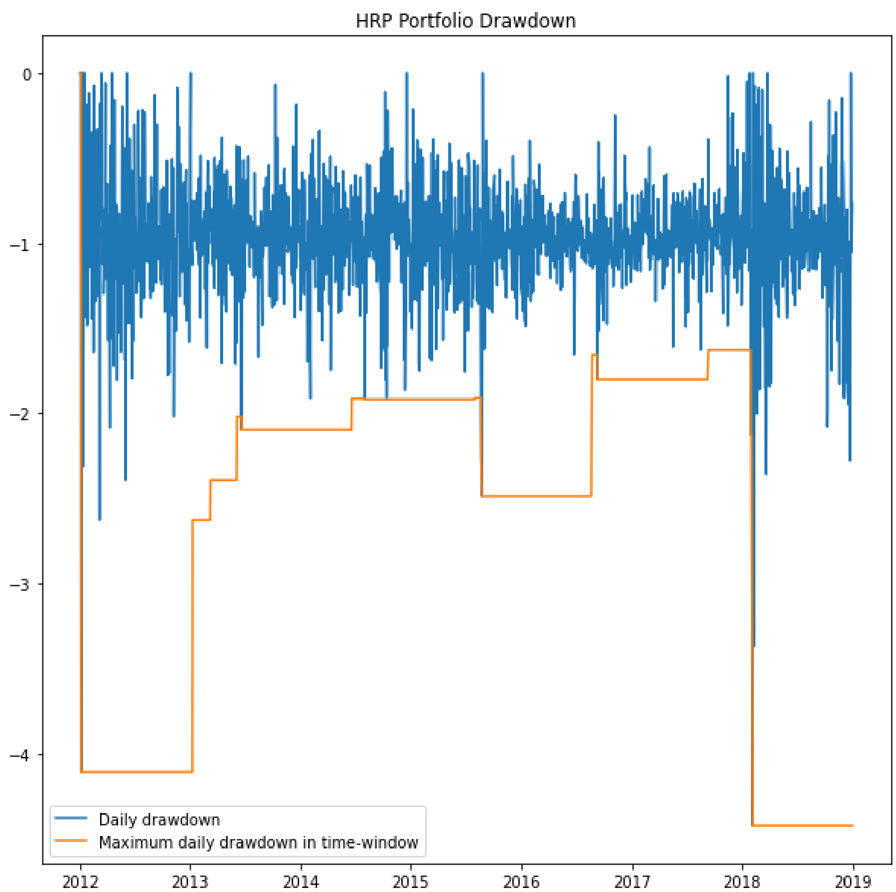


Figure 70, HRP Portfolio Drawdown. Data Source: A personal elaboration on Python.

<u>The CLA Portfolio</u>

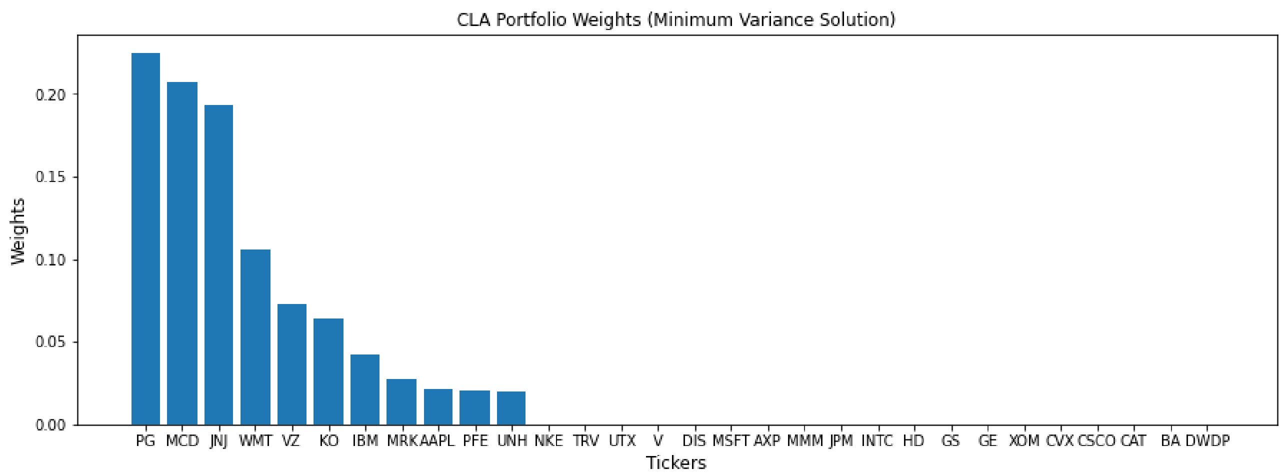


Figure 71, CLA Portfolio weights. Data Source: A personal elaboration on Python.

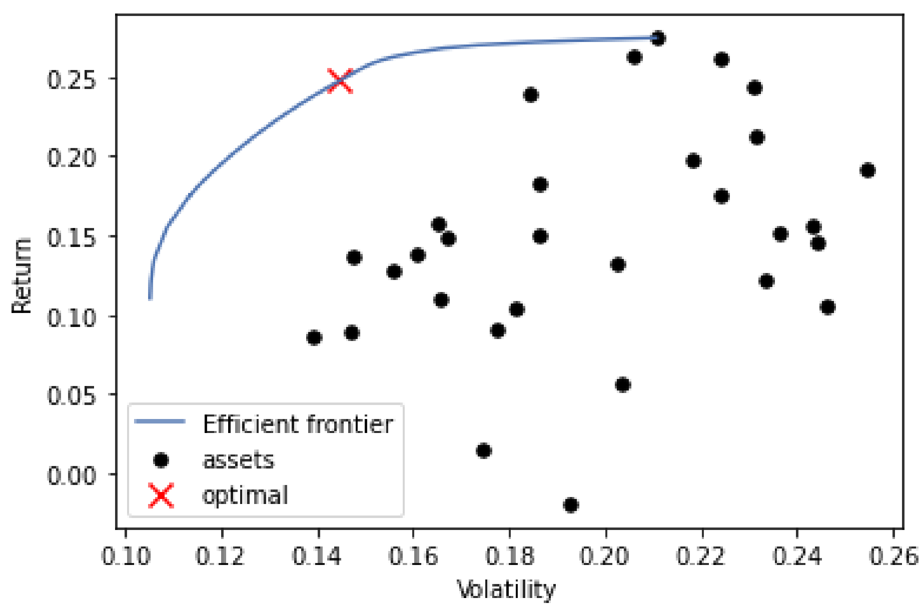


Figure 72, CLA Efficient Frontier. Data Source: A personal elaboration on Python.

Skew: -0,2875361042783768
Kurtosis: 3,4698266816637124

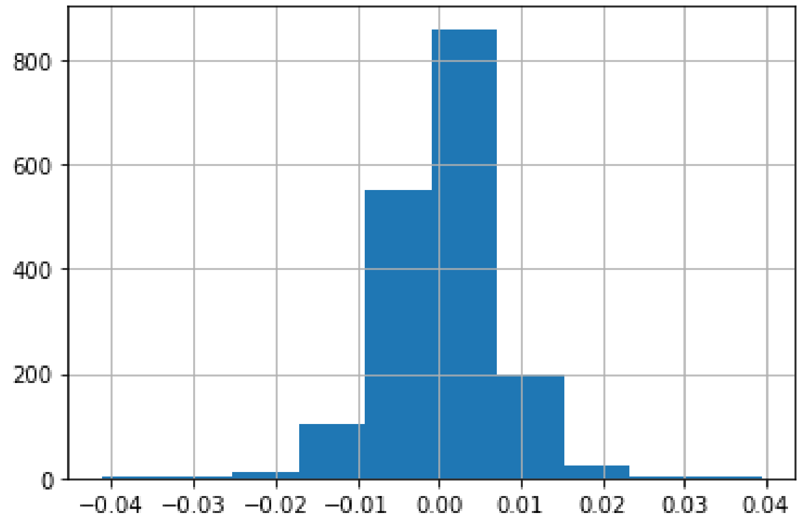


Figure 73, CLA Portfolio Returns Distribution. Data Source: A personal elaboration on Python.

N Day VaR Results:

1-day VaR @ 99% confidence: 249258,76

2-day VaR @ 99% confidence: 352505,11

3-day VaR @ 99% confidence: 431728,83

4-day VaR @ 99% confidence: 498517,51

5-day VaR @ 99% confidence: 557359,52

6-day VaR @ 99% confidence: 610556,77

7-day VaR @ 99% confidence: 659476,68

8-day VaR @ 99% confidence: 705010,23

9-day VaR @ 99% confidence: 747776,27

10-day VaR @ 99% confidence: 788225,39

11-day VaR @ 99% confidence: 826697,77

12-day VaR @ 99% confidence: 863457,66

13-day VaR @ 99% confidence: 898715,22

14-day VaR @ 99% confidence: 932640,86

15-day VaR @ 99% confidence: 965375,01

Figure 74, N Day VaR Results. Data Source: A personal elaboration on Python.

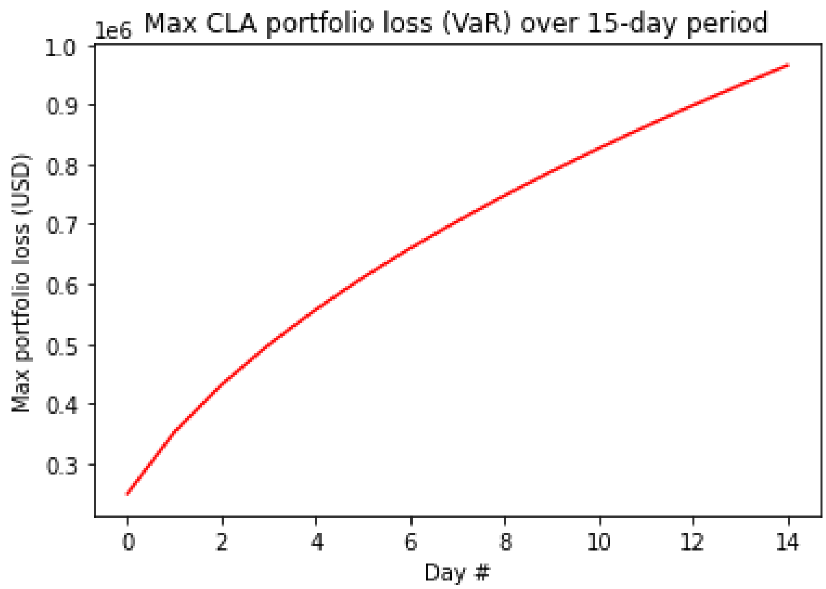


Figure 75, CLA VaR over 15-days. Data Source: A personal elaboration on Python.

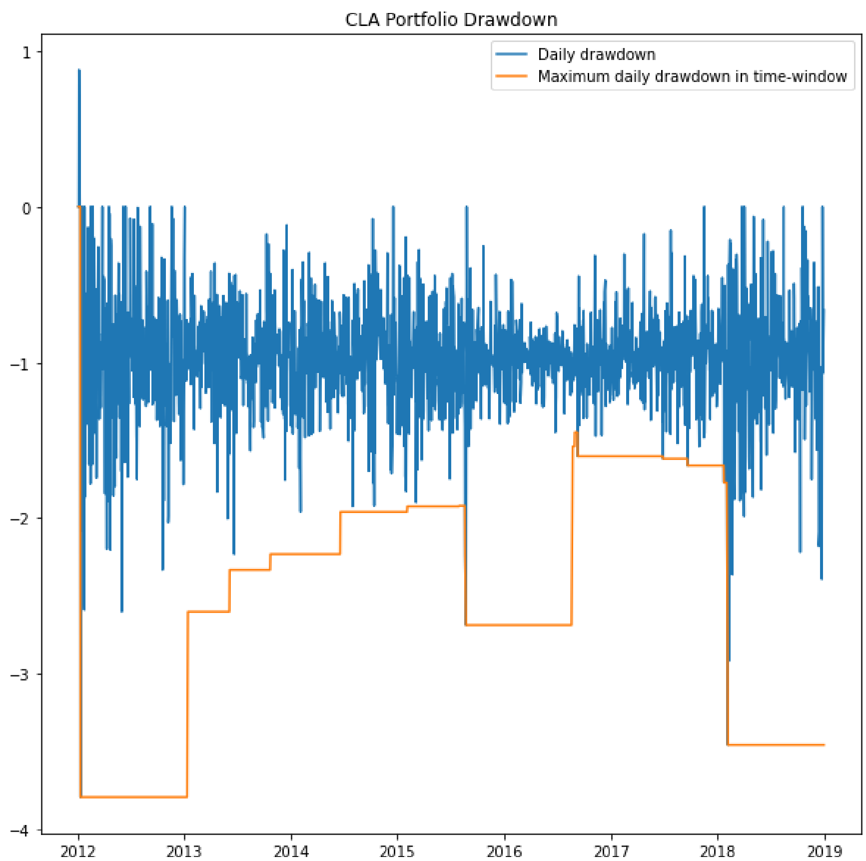


Figure 76, CLA Portfolio Drawdown. Data Source: A personal elaboration on Python.

The Inverse Variance (IV) Portfolio

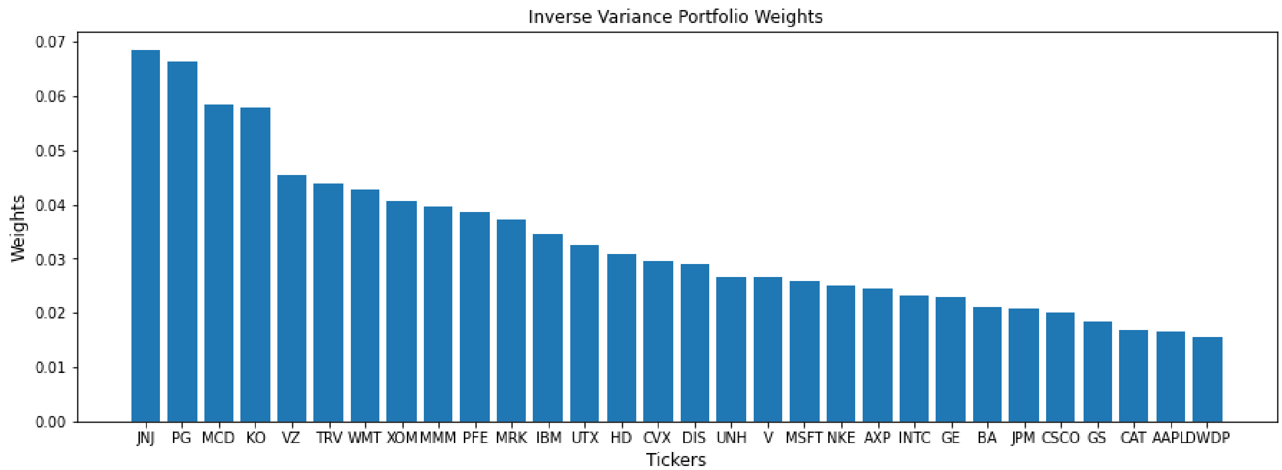


Figure 77, IV Portfolio weights. Data Source: A personal elaboration on Python.

Skew: -0,14808740533053372
Kurtosis: 5,257294574696461

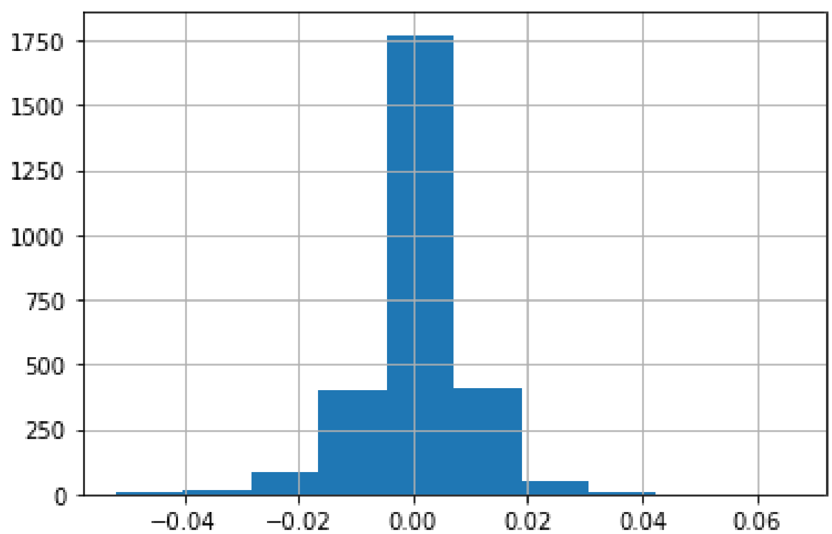


Figure 78, IV Portfolio Returns Distribution. Data Source: A personal elaboration on Python.

N Day VaR Results:

1-day VaR @ 99% confidence: 331826,14

2-day VaR @ 99% confidence: 469273,03

3-day VaR @ 99% confidence: 574739,74

4-day VaR @ 99% confidence: 663652,29

5-day VaR @ 99% confidence: 741985,81

6-day VaR @ 99% confidence: 812804,73

7-day VaR @ 99% confidence: 877929,45

8-day VaR @ 99% confidence: 938546,06

9-day VaR @ 99% confidence: 995478,43

10-day VaR @ 99% confidence: 1049326,4

11-day VaR @ 99% confidence: 1100542,81

12-day VaR @ 99% confidence: 1149479,48

13-day VaR @ 99% confidence: 1196416,17

14-day VaR @ 99% confidence: 1241579,74

15-day VaR @ 99% confidence: 1285157,12

Figure 79, N Day VaR Results. Data Source: A personal elaboration on Python.

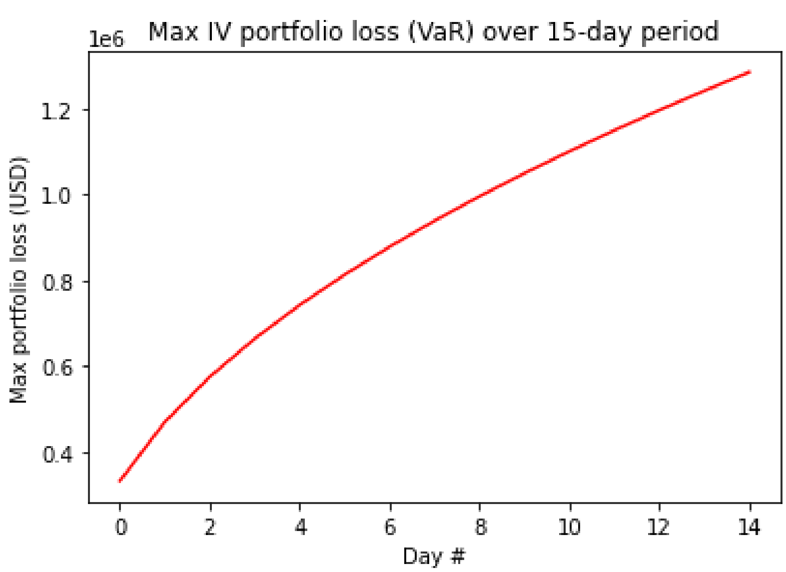


Figure 80, IV VaR over 15-days. Data Source: A personal elaboration on Python.

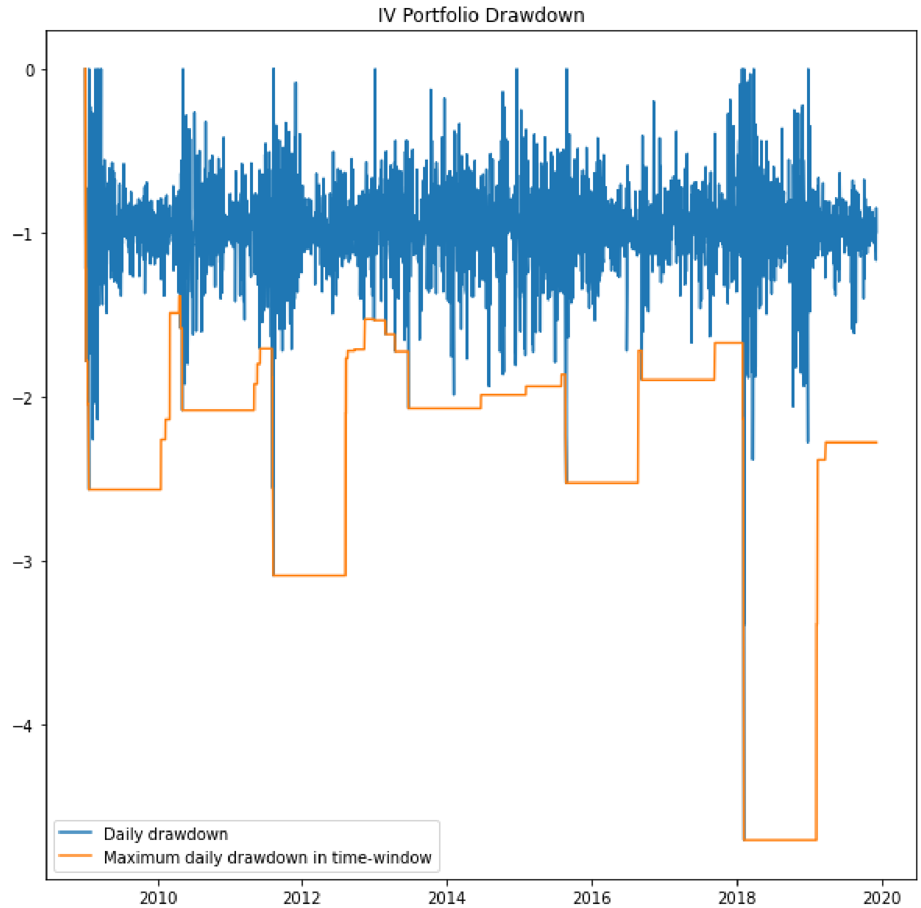


Figure 81, CLA Portfolio Drawdown. Data Source: A personal elaboration on Python.

The Equal-Weighted (EW) Portfolio

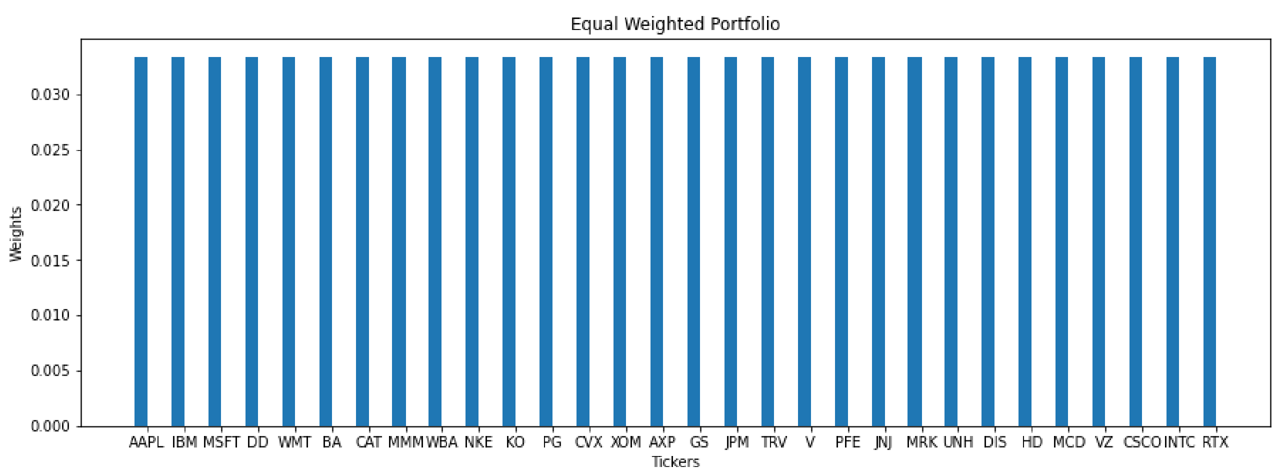


Figure 82, EW Portfolio weights. Data Source: A personal elaboration on Python.

Skew: -0,07168131420992294
Kurtosis: 5,906773749543939

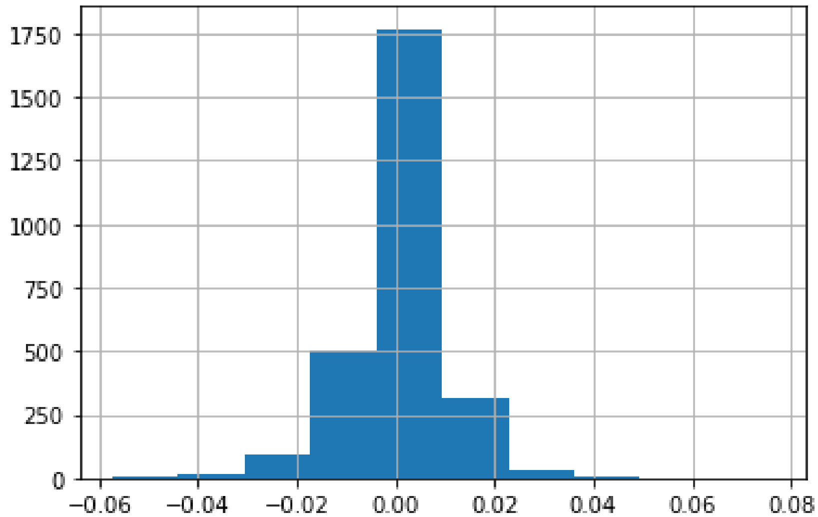


Figure 83, EW Portfolio Returns Distribution. Data Source: A personal elaboration on Python

N Day VaR Results:

1-day VaR @ 99% confidence: 363062,37

2-day VaR @ 99% confidence: 513447,73

3-day VaR @ 99% confidence: 628842,48

4-day VaR @ 99% confidence: 726124,75

5-day VaR @ 99% confidence: 811832,15

6-day VaR @ 99% confidence: 889317,56

7-day VaR @ 99% confidence: 960572,75

8-day VaR @ 99% confidence: 1026895,46

9-day VaR @ 99% confidence: 1089187,12

10-day VaR @ 99% confidence: 1148104,03

11-day VaR @ 99% confidence: 1204141,67

12-day VaR @ 99% confidence: 1257684,95

13-day VaR @ 99% confidence: 1309040,0

14-day VaR @ 99% confidence: 1358455,01

15-day VaR @ 99% confidence: 1406134,52,

Figure 84, N Day VaR Results. Data Source: A personal elaboration on Python.

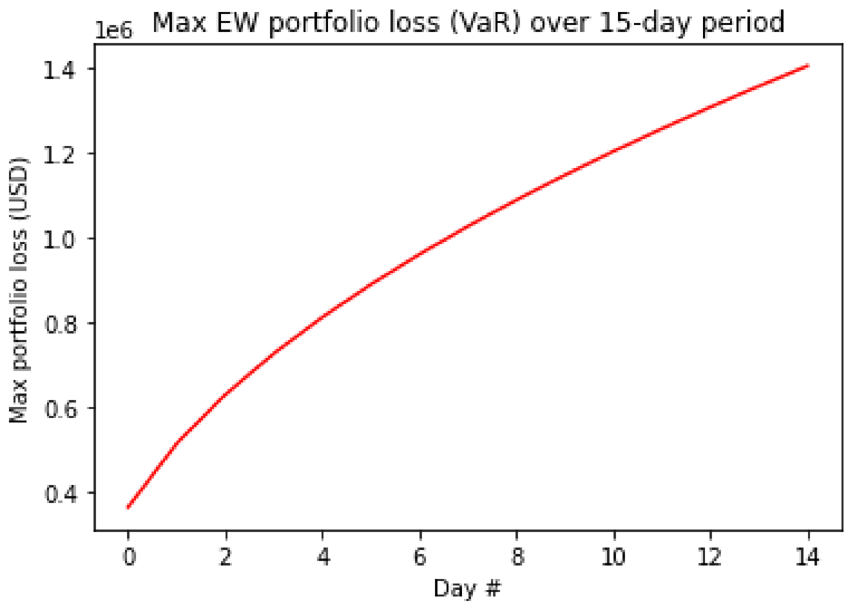


Figure 85, EW VaR over 15-days. Data Source: A personal elaboration on Python.

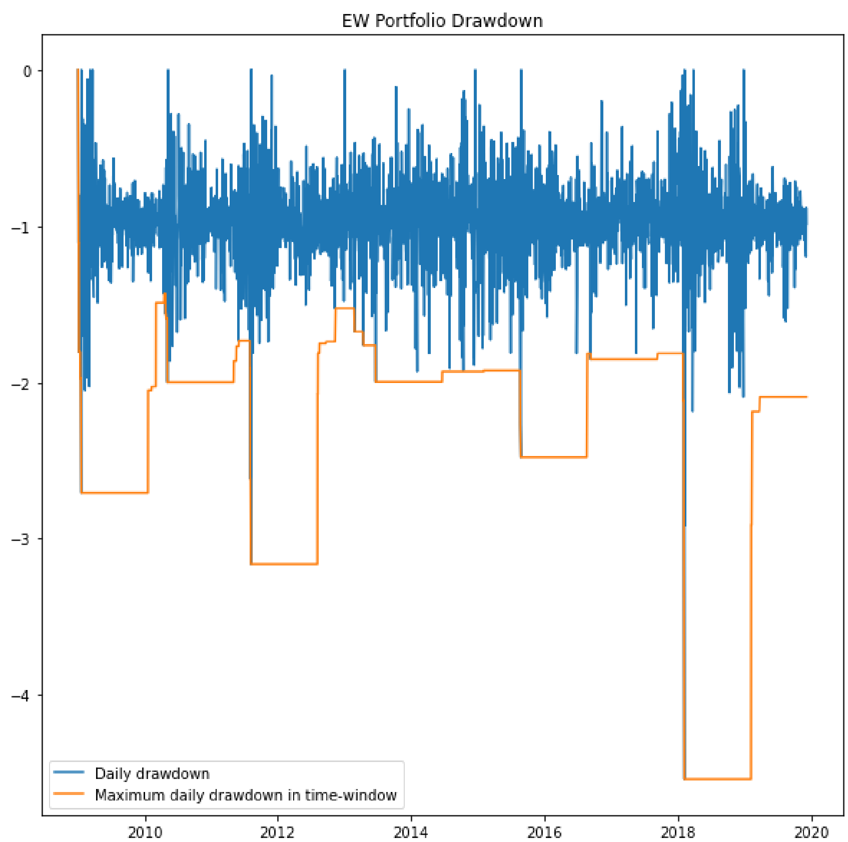


Figure 86, EW Portfolio Drawdown. Data Source: A personal elaboration on Python.

The Random-Weighted (RDM) Portfolio

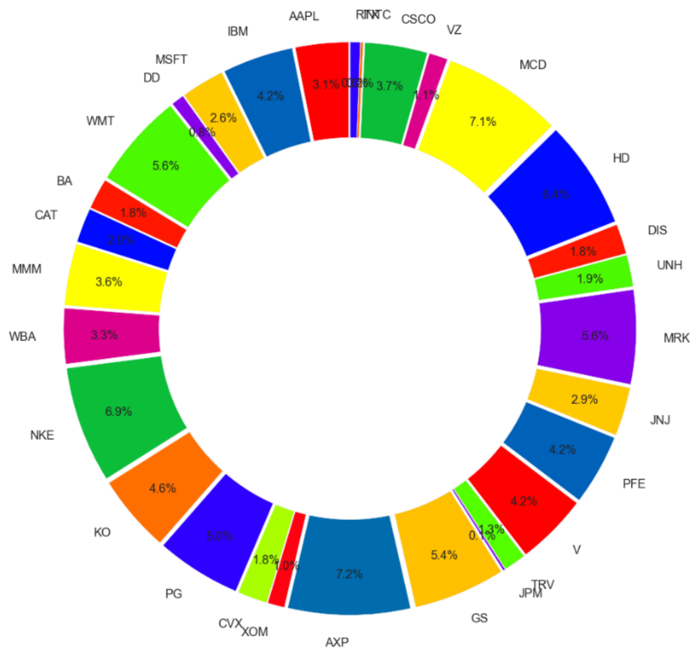


Figure 87, EW Portfolio weights. Data Source: A personal elaboration on Python.

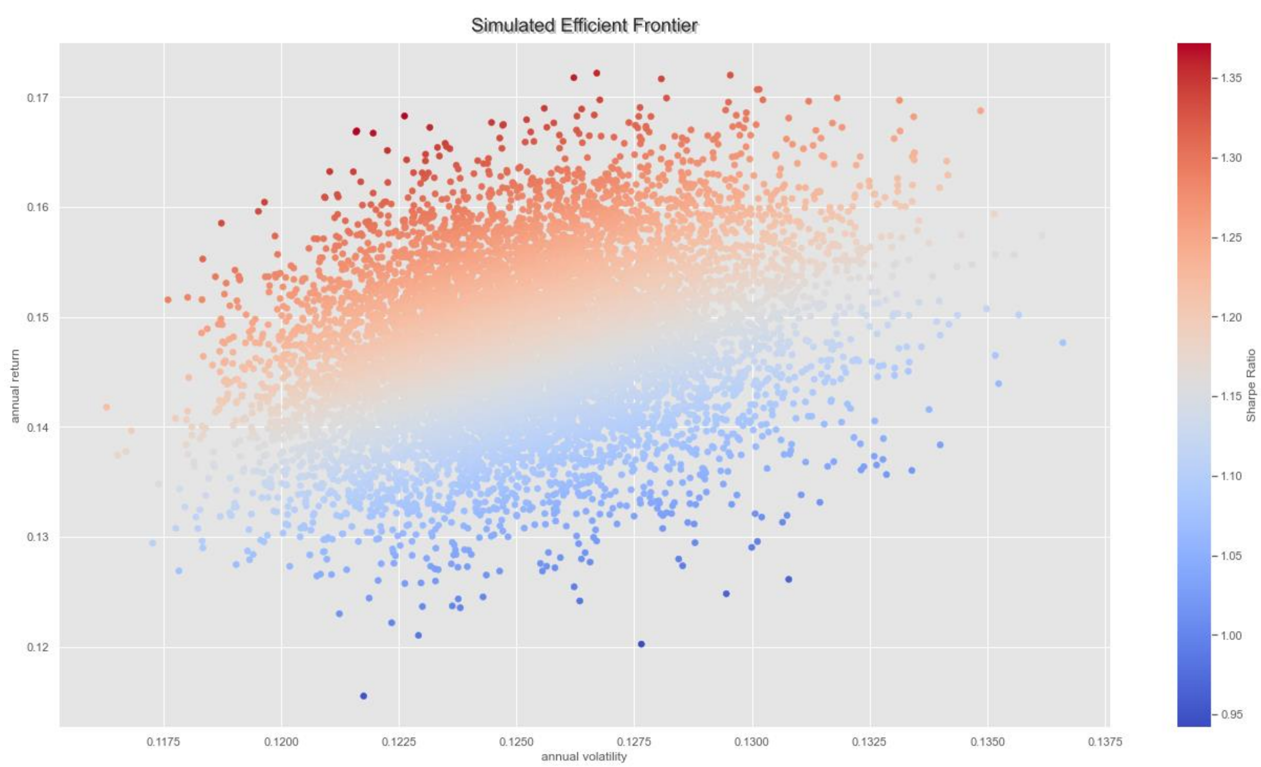


Figure 88, RDM Portfolio Simulated Efficient Frontier. Data Source: A personal elaboration on Python.

3.3.2 The ETF Portfolio Results

In-Sample Test

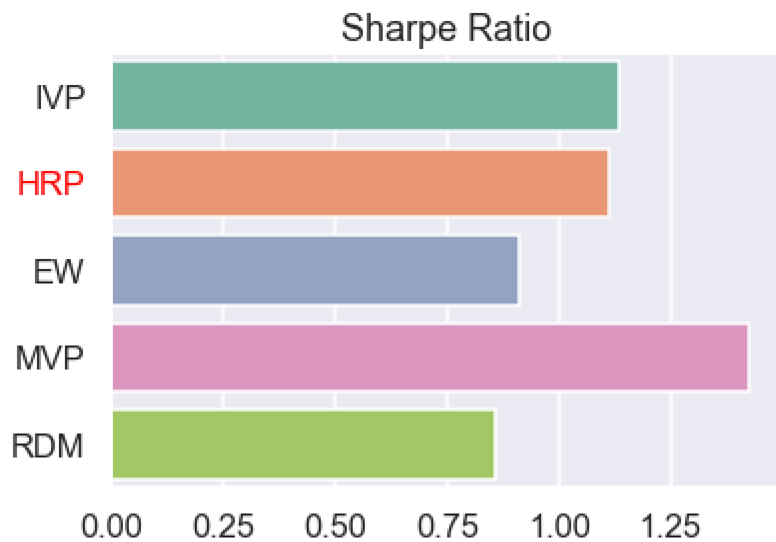


Figure 89, In-Sample Sharpe Ratio. Data Source: A personal elaboration on Python.

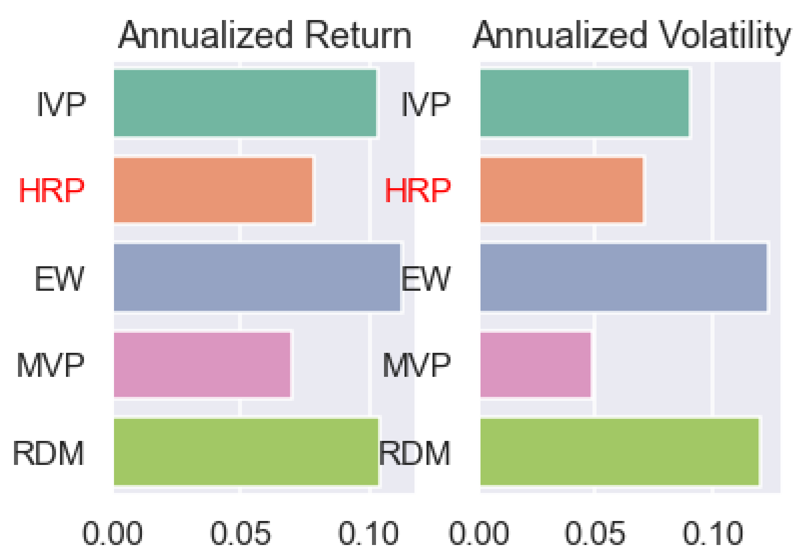


Figure 90, In-Sample Annualized Return & Volatility. Data Source: A personal elaboration on Python.

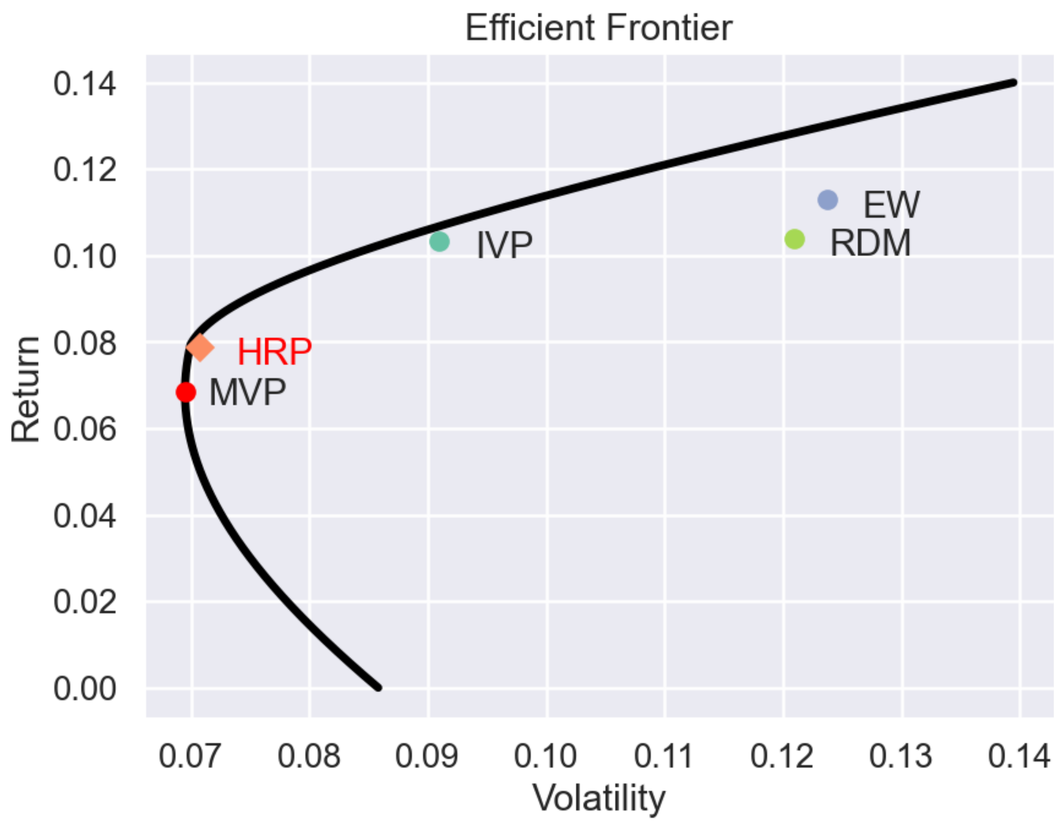


Figure 91, In-Sample Efficient Frontier. Data Source: A personal elaboration on Python.

Out-of-Sample Test

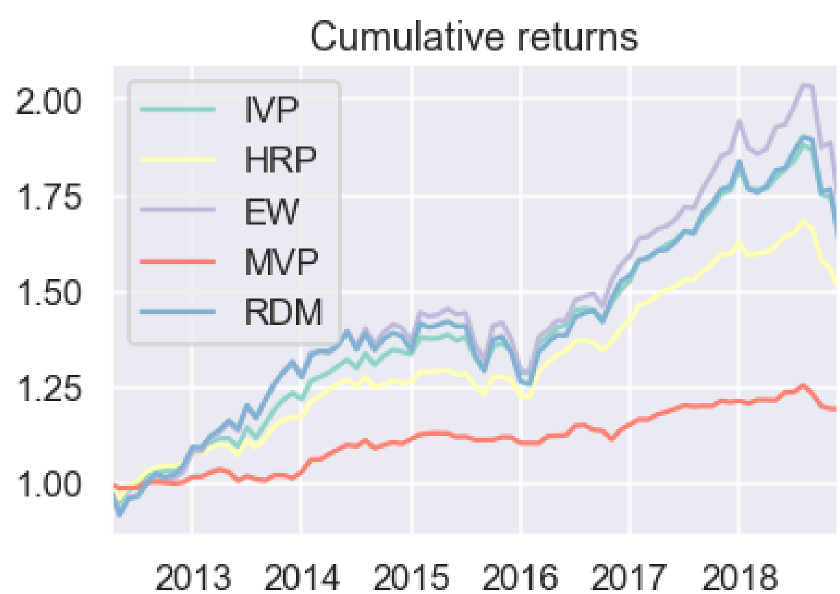


Figure 92, Out-of-Sample Cumulative Returns. Data Source: A personal elaboration on Python.

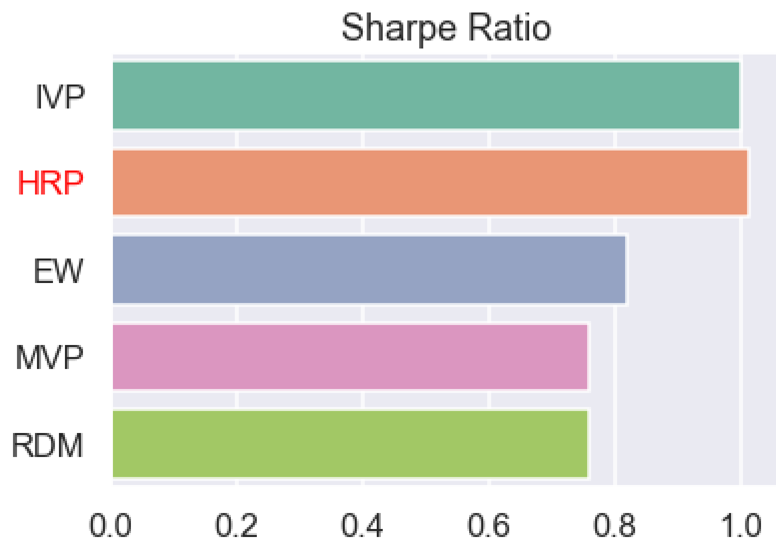


Figure 93, Out-of-Sample Sharpe Ratio. Data Source: A personal elaboration on Python.

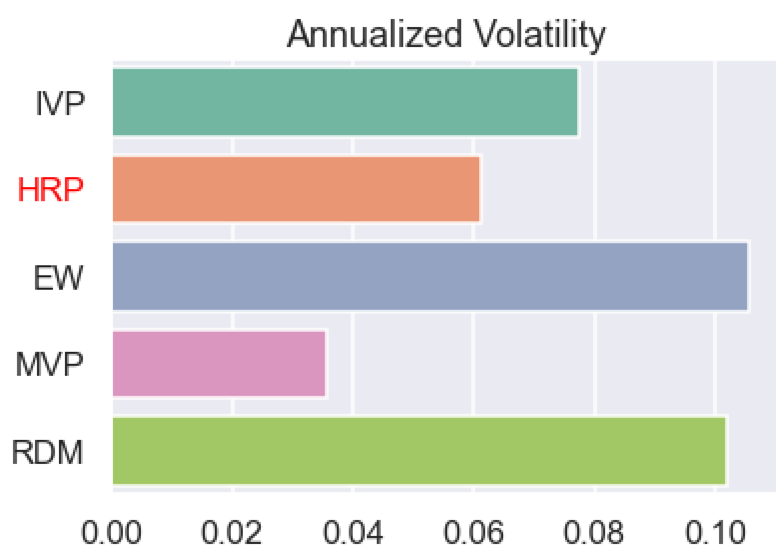


Figure 94, Out-of-Sample Annualized Volatility. Data Source: A personal elaboration on Python.

CONCLUSIONS

Since an investor doesn't have the option of not risking, the only option he has is to choose the amount and type of risk he wants to be exposed to. In order to do this, it is necessary to manage the portfolio risks on a continuous basis. This is the main goal of risk management, which is the set of actions aimed at configuring the portfolio to maintain risk exposure within set limits and to optimize risks with respect to the conditions imposed. Only after this meticulous and comprehensive assessment it is possible to identify the most efficient portfolio that provides the best returns for the accepted level of risks. For this reason, I was intrigued by the possibilities of the de Prado's Hierarchical Risk Parity model since it offers an excellent allocation tool for minimizing the portfolio overall volatility.

The main object of this thesis was to deeply investigate and strengthen the understanding of the main portfolio allocation strategies and compare their performance and risk metrics with the de Prado's Hierarchical Risk Parity portfolio. It is particularly interesting to analyze whether machine learning based portfolios outperform both the Markowitz minimum variance or CLA and the traditional risk-based portfolios. This purpose was carried out by examining the developments of the main portfolio optimization academic literature along with the recent theories and new algorithms used in portfolio construction.

Although representing an intuitive quantitative framework, the Markowitz mean-variance approach is known to deliver unstable and inconsistent solutions. The main root of the problem is related to the difficulties in estimating the expected return. For this reason, researchers and economists have tried to develop new models and theories that would base their results on the estimation of the covariance matrix rather than the returns. This has led to the so called "Risk-based" portfolio optimization problems, portfolio strategies that mainly focus their attention on the risk factors. Despite their good performance and their applicability on industry-wide portfolio optimization problems, they tend to provide affected results due to their great sensitivity to the covariance matrix inversion. Furthermore, the traditional risk-based portfolios usually have shown to provide poor results out-of-sample. Therefore, because of the instability and inaccuracy of the afore-mentioned portfolio optimization methodologies, Bailey and Lòpez de Prado came up with the Hierarchical Risk Parity portfolio, an approach that could address the major quadratic programming limitations.

The HRP methodology avoids the condition of the invertibility of the covariance matrix and offers an efficient approach that helps constructing more robust portfolios in terms of weights diversification, risk minimization and performance metrics.

In the empirical and final chapter of the thesis I have compared five different portfolio allocation approaches for two distinct portfolios: the Dow Jones Industrial Average portfolio and all ETFs portfolio. The analysis is carried out over the period that goes from 01/01/2012 to 01/01/2019. I apply the optimization models on the performance of the two studied portfolios from 01/01/2012 to 01/01/2016 for the in-sample test and out-of-sample from the 01/01/2016 to 01/01/2019. The main purpose of this empirical research is to investigate whether HRP based portfolios outperform the traditional strategies out-of-sample, as noted by de Prado.

I have constructed and analyzed 4 long-only asset allocation strategies that are used as a good benchmarking tool for the HRP approach: The Minimum Variance portfolio (MV) or CLA, the Inverse Variance portfolio (IV), the Equal Weighted portfolio (EW) and a Random Weighted portfolio (RDM). Subsequently I have applied these strategies on the two constructed portfolios. Even if the results are subject to the investment universe (an index-based portfolio and an all ETFs portfolio), some interesting conclusion can be pronounced.

The Dow Jones Index Hierarchical Risk Parity (HRP) portfolio notably manages to strengthen its robustness in the out-of-sample test. Not only, in line with de Prado’s findings, it achieves the lowest annualized volatility outperforming even the CLA solution, but it also achieves a very good Sharpe ratio result. In the out-of-sample test, the MV portfolio seems to be subject to random shocks, and therefore it is much more volatile as demonstrated by its cumulative return and annualized volatility’s results. For what concerns the all ETFs HRP portfolio, though it does not manage to beat the MV portfolio in terms of risk minimization out-of-sample, it still achieves a very low level of volatility, and it presents the highest level of Sharpe ratio compared to the other allocation approaches. Both the Dow Jones Index and the all ETFs HRP portfolios exhibit the lowest level of maximum drawdown, a very low level of Value at Risk (even if higher than the one achieved with the minimum variance approach) and a very good return distribution as well. Furthermore, the HRP algorithm, allows for a much more uniform weight allocation especially if compared with the portfolios built with the CLA algorithm, which literally concentrate 75% of their holdings on the top 4-5 investments. This means the HRP methodology allows for a better risk diversification.

From the empirical results, therefore, it is possible to conclude the HRP strategy seems a good approach to obtain well diversified portfolios. In addition to that, it leads to an optimization of the risk and performance metrics, especially in the out-of-sample test.

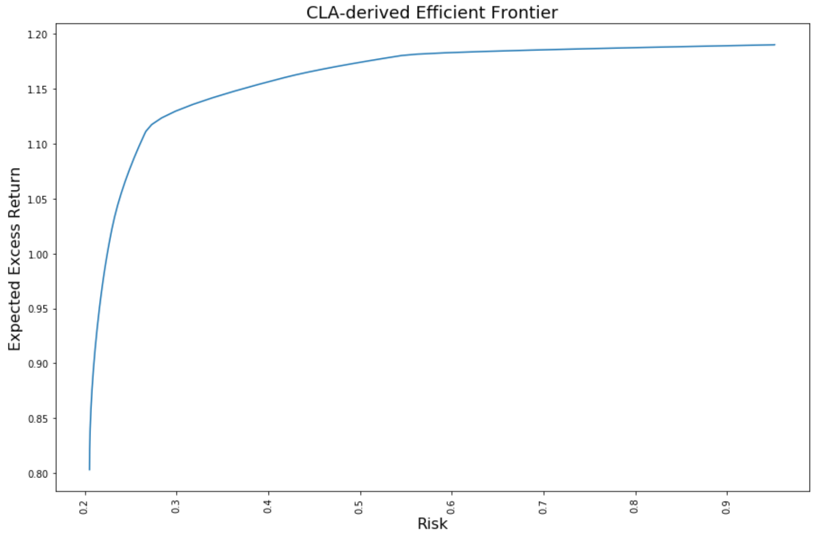
Nonetheless, there are few features that might be implemented to improve the model presented in this thesis I leave as a suggestion for further research. First, the HRP portfolio allocation approach allows only for long positions. Therefore, short sales and capital gains taxes can be added in the analysis. Secondly, new portfolio allocation strategies can be proposed for benchmarking the HRP portfolio. For instance, a Maximum Diversification portfolio (MD) and an Equal Risk Contribution portfolio (ERC) can be used as a good comparison tool. Then, the HRP method still does not produce fully diversified portfolios, with certain securities exhibiting the greatest weight for an extended period of time. Lastly, it would be interesting to evaluate the HRP performance out-of-sample via Monte Carlo as done by de Prado in his paper "*Building Diversified Portfolios that Outperform Out-of-Sample*". Indeed, a Monte Carlo experiment might provide even a better result in terms of variance minimization when the model is tested out-of-sample.

EXECUTIVE SUMMARY

The paper aims at highlighting and analyzing the development of the main portfolio allocation approaches. The de Prado discovery of the Hierarchical Risk Parity (HRP) algorithm for optimizing a financial portfolio, deeply questions the efficiency of the Markowitz efficient frontier theory as well as the traditional portfolio allocation strategies. Adopting a machine learning technique in the asset allocation process, allows to develop a more robust portfolio in terms of risk minimization and performance metrics. The HRP methodology helps building a diversified portfolio based on the information contained in the covariance matrix. In the empirical analysis, the HRP portfolio stands out in the out-of-sample test, achieving lower risk indicators compared with other traditional portfolio construction models.

One of the most important and difficult challenges in the asset management industry is related to the optimal asset allocation choice. Investment portfolio theories govern the way an individual investor or firm allocates his/its wealth and assets within an investing portfolio. Investment managers try to achieve the highest return adjusted for risk by analyzing the way the risk and return evolves over time. In recent years, a growing capital flow has poured onto the financial markets by means of increasingly diverse and complex portfolio techniques with the goal of achieving a satisfactory level of return. However, a portfolio is always and anyhow exposed to some level and type of risk, which is directly and strongly correlated with the portfolio return level. Therefore, the person in charge of managing the portfolio needs to set the risk/return profile in compliance with his/her preferences and objectives. The pioneer of the theories governing this process is the American economist Harry Markowitz, who in 1952 published the paper “Portfolio Selection” introducing the Mean-Variance portfolio allocation approach, which would have marked a milestone in the portfolio theory. Considered the father of the Modern Portfolio Theory (MPT), Markowitz was the first scholar developing a meticulous mathematical structure for portfolio optimization that could recognize the subtle yet strong relation existing between risk and return. He rejected the common idea that the optimal portfolio is necessarily the one achieving the highest return. His mean-variance approach is indeed an investment theory which tries to maximize the portfolio expected return for a defined and accepted level of risk, or equivalently minimize risk for a given level of expected return, by carefully deciding the various securities allocation. Markowitz managed to formulate the "optimal" approach for allocating resources across risky securities in a static world, where people are only interested in the mean and variance of the portfolio's return. According to the mean-variance environment, everyone faces a trade-off when constructing his/her optimal portfolio. Risk-averse investors would be willing to give up a bit of return in change of safer portfolios, while risk-seeking people want to maximize the expected return no matter the variance. The only efficient portfolios are those that for any defined amount of variance, have the highest possible expected return. The set of all these portfolios would construct the so called "efficient frontier" defined as the bundle of optimal portfolios that show the highest expected return for a given level of risk or the lowest risk for a given level of expected return. In addition to publishing his masterpiece *"Portfolio Selection"* in 1952, Markowitz, while working for the RAND Corporation, developed an algorithm for solving quadratic problems in 1956. The brilliant and young economist implemented his groundbreaking mean-variance approach into an optimization algorithm, the Critical Line Algorithm (CLA). The Markowitz efficient frontier solution requires both an equality constraint (that the portfolio's weights sum up to one) and an inequality constraint (a lower and upper bound for the weights, which are 0 and 1 respectively), in order to be solved. As there is no analytic solution to this problem, the breakthrough of the young American scholar was to develop an open-source algorithm that could solve inequality-constrained portfolio optimization problems and compute the optimal set of efficient portfolios lying on the curve[119].

[119] CLA-derived Efficient Frontier. Data Source: A personal elaboration on Python.

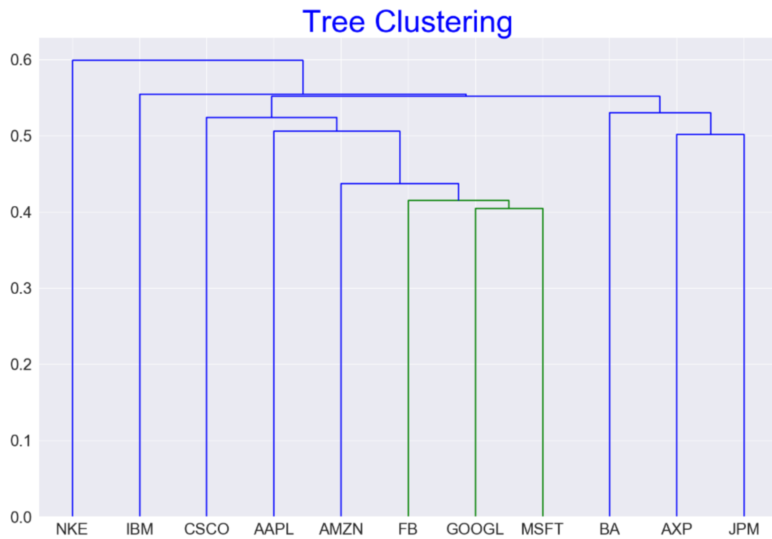


The next larger step in the modern portfolio theory was the development of the Capital-Asset Pricing Model (CAPM) by William Sharpe in 1964, which became another evidence for researchers and investors of the interconnections between the asset risk and asset return. Unlike the mean-variance approach, the CAPM introduces the distinction of two types of risks: the systematic risk (that portion of risk that cannot be diversified away and affects all the securities) and the idiosyncratic or unsystematic risk (which depends on the company/industry specific characteristics, and can be eliminated through diversification).

Since Markowitz work, asset managers as well as academics around all the world have been focusing on carrying out theories and new approaches to build robust portfolios. The financial industry, however, is a very volatile one: a constant critical analysis of the rules governing the risk-return relationship is therefore always required. The global financial crisis of 2008 has displayed all the limitations and drawbacks of the traditional portfolio allocation methodologies. The Markowitz efficient frontier theory has demonstrated to lead to inconsistent outcomes especially due to the challenges in estimating the expected returns and the covariances for the different asset classes. Furthermore, even the CLA solution somewhat produced unstable results. Indeed, small deviations in the forecasted returns lead the algorithm to develop very different portfolios. The CAPM instead, although widely used in the academic financial world due to its simplicity and well-defined framework for estimating the cost of capital, was subject to criticism due to its unrealistic and oversimplified assumptions. Therefore, academics in the asset management world felt the need to develop new theoretical frameworks that could lead to an optimal asset allocation. Among the new portfolio allocation approaches, the ones that grasped the most attention from practitioners and researchers are the risk-based methodologies. Since the portfolio expected return is considered unpredictable, these new strategies are risk-based ones because they try to estimate the risk factors affecting each asset in the portfolio. The new portfolio weights depend only on the specific risk factors affecting each security in the portfolio. Some of the most relevant risk-based models include: Equal Risk Contribution portfolio (ERC) where each security contributes the same amount to the overall portfolio volatility; the Equally Weighted portfolio or Naïve portfolio strategy (EW) that evenly allocates the same weight on each asset; the Global Minimum Variance portfolio (GMV), which indicates the efficient frontier portfolio exhibiting the lowest possible volatility ; the Maximum Diversification Portfolio (MDP) that maximizes the diversification ratio; the Maximum Sharpe Ratio Portfolio (MSP) finding the optimal capital allocation in the presence of a riskless asset; the Inverse Volatility Strategy or Naïve Risk Parity (IV), a portfolio allocation strategy which allocates to each component a weight equal to the inverse of their volatilities, measured by the standard deviation; the Market-Capitalization-Weighted Portfolio (MCWP) which computes the weights as the average of the market capitalizations of the portfolio constituents over the sum of the average of the same capitalizations.

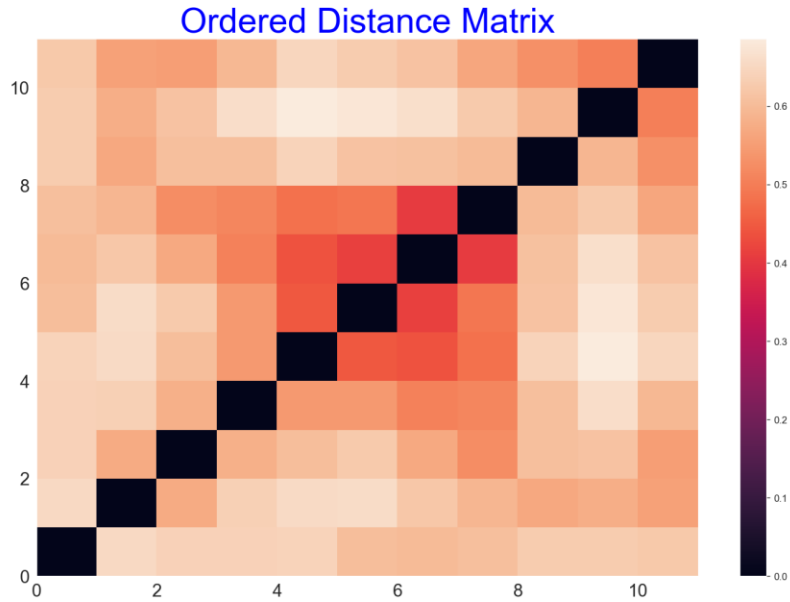
All these new portfolio allocation strategies prevented the estimation of the unpredictable asset returns by focusing exclusively on the covariance matrix. However, over the course of the years, a large empirical evidence has demonstrated that risk-based allocation methodologies, requiring the covariance matrix inversion, result into serious estimation errors that may cancel out the advantages of portfolio diversification.

In order to overcome this problem, the Spanish economist Marcos Lopez de Prado was the first researcher proposing a hierarchical model for the portfolio construction. A hierarchical structure can indeed help solving complex problems breaking them down into smaller and simpler subgroups whose solutions are then grouped together afterwards. The breakthrough of de Prado was to develop an algorithm, the Hierarchical Risk Parity portfolio (HRP), that could solve CLA instability issues and at the same time produce portfolios that could outperform the traditional risk-based allocation strategies. The HRP model, by avoiding the inversion of the covariance matrix and identifying a hierarchical structure in the portfolio weights, applies graph theory and machine learning techniques to construct a diversified portfolio based on the information contained in the covariance matrix. The HRP algorithm fully concentrates on the covariance matrix, hence dropping the forecasted returns. It operates through three distinct phases: 1) Tree Clustering, 2) Quasi-diagonalization and 3) Recursive Bisection. The first step of the HRP approach allows to group the securities in the portfolio under a hierarchical structure, where clusters of similar assets are created using the correlation coefficients. A typical hierarchical portfolio visualization is a dendrogram graph[120].

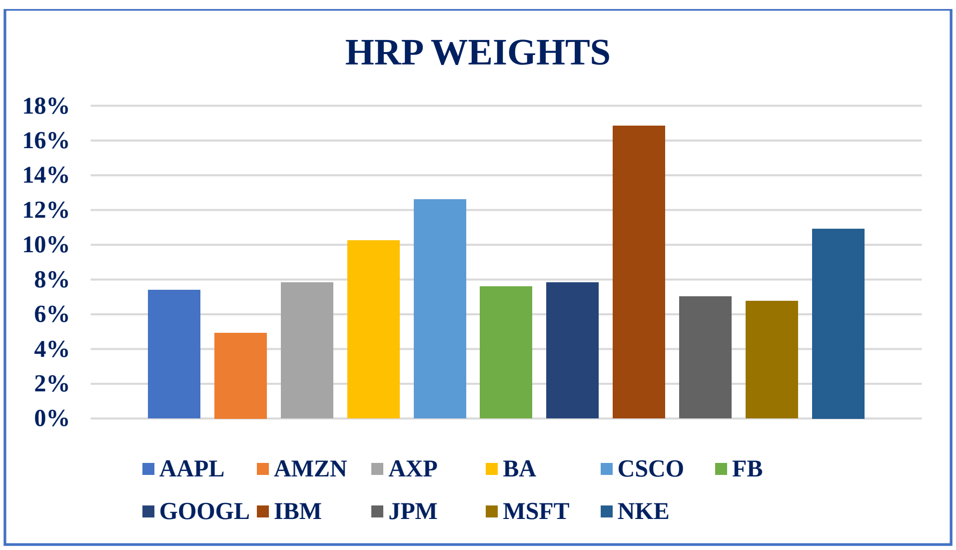


The quasi-diagonalization process instead, allows a reorganization of the columns and rows of the covariance matrix using the information of the formed clusters, so that the largest entries are placed along the diagonal. The main goal of the quasi-diagonalization algorithm is to group similar holdings together while dissimilar ones lie around the matrix[121].

[120] Tree Clustering Dendrogram graph. Data Source: A personal elaboration on Python.
[121] Quasi-diagonalization stage of the HRP algorithm. Data Source: A personal elaboration on Python.



Finally, the last stage of the algorithm, the Recursive Bisection, defines the optimal allocation by assigning the final weights to the securities in the portfolio[122].



[122] HRP Portfolio weights. Data Source: A personal elaboration on excel.

The HRP portfolio proposed by de Prado in his famous paper "*Building Diversified Portfolios that outperform out-of-sample"* in 2016, presents outstanding results in terms of robustness. Indeed, one of the main discoveries of de Prado, is that the machine learning based portfolio provides better risk performance indicators out-of-sample, thus outperforming the traditional portfolio allocations methodologies

Since an investor doesn't have the option of not risking, the only option he has is to choose the amount and type of risk he wants to be exposed to. In order to do this, it is necessary to manage the portfolio risks on a continuous basis. This is the main goal of risk management, which is the set of actions aimed at configuring the portfolio to maintain risk exposure within set limits and to optimize risks with respect to the conditions imposed. Only after this meticulous and comprehensive assessment it is possible to identify the most efficient portfolio that provides the best returns for the accepted level of risks. For this reason, I was intrigued by the possibilities of the de Prado's Hierarchical Risk Parity model as it provides an innovative and dynamic portfolio optimization framework which gives strong empirical results in terms of risk minimization and remarkably uses a machine learning algorithm, offering a high-level understanding through digital images. I also observed the empirical literature on the HRP model is quite scarce, something which may prevent other practitioners from further adopting the algorithm.

The above-mentioned considerations lead me to empirically test the validity of the HRP portfolio approach. For the empirical analysis I have decided to compare five different portfolio allocation approaches for two distinct portfolios: the Dow Jones Industrial Average portfolio and all ETFs portfolio. The choice of the Dow 30 index for the analysis is motivated both because it is computed in a different way and because it presents a diversified composition across several sectors which makes it more interesting for creating different scenarios[123]. On the other hand, I have concentrated my attention on an all ETFs portfolio due to the great importance these index-based funds cover in the financial markets nowadays[124].

The analysis is carried out over the period that goes from 01/01/2012 to 01/01/2019. I apply the optimization models on the performance of the two studied portfolios from 01/01/2012 to 01/01/2016 for the in-sample test and out-of-sample from the 01/01/2016 to 01/01/2019. The

---

[123] ETFs portfolio constituents. Data Source: A personal elaboration on excel from Yahoo-finance.
[124] Dow Jones 30 constituents. Data Source: A personal elaboration on excel from Yahoo-finance.

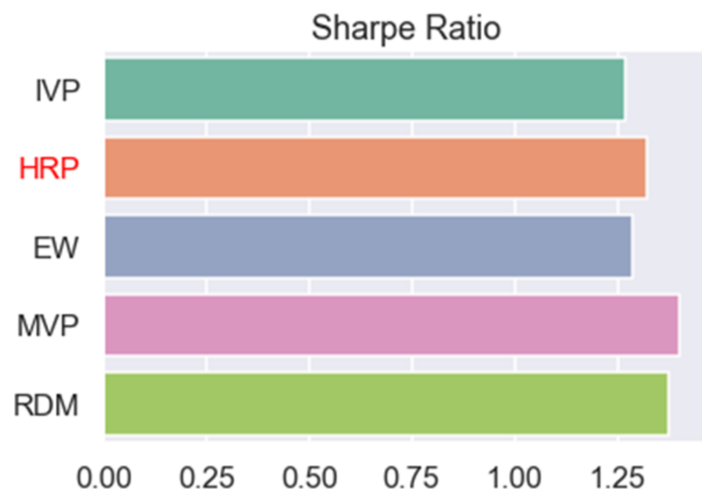
main purpose of this empirical research is to investigate whether HRP based portfolios outperform the traditional strategies out-of-sample, as noted by de Prado.

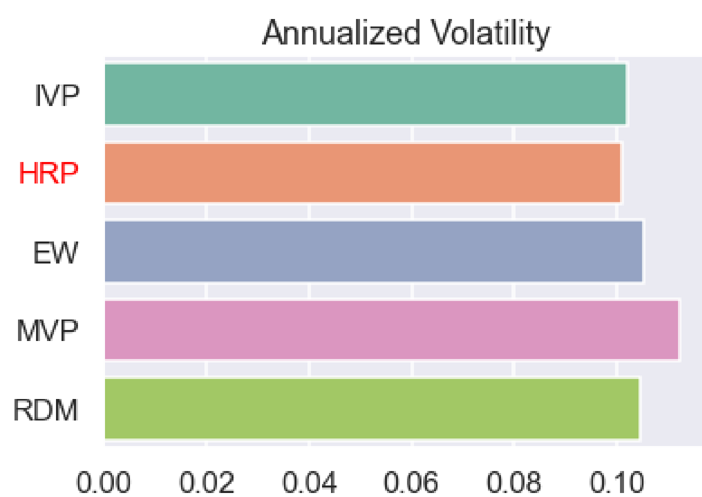
Ticker	Name	Asset Class	Index
QQQ	Invesco QQQ Trust, Series 1	Equity	NASDAQ-100 Index
IWF	iShares Russell 1000 Growth ETF	Equity	Russell 1000 Growth Index
SPY	SPDR S&P 500	Equity	S&P 500 Index
VTI	Vanguard Total Stock Market ETF	Equity	CRSP US Total Market Index
OEF	iShares S&P 100 ETF	Equity	S&P 100 Index
DIA	SPDR Dow Jones Industrial Avera	Equity	Dow Jones Index
IWO	iShares Russell 2000 Growth ETF	Equity	Russell 2000 Growth Index
IJH	iShares Core S&P Mid-Cap ETF	Equity	S&P MidCap 400 Index
MDY	SPDR MidCap Trust Series I	Equity	S&P MidCap 400 Index
IWM	iShares Russell 2000 ETF	Equity	Russell 2000 Index
IWD	iShares Russell 1000 Value ETF	Equity	Russell 1000 Value Index
IWN	iShares Russell 2000 Value ETF	Equity	Russell 2000 Value Index
FEZ	SPDR DJ Euro STOXX 50 Etf	Equity	EURO STOXX 50 Index
PFF	iShares US Preferred Stock ETF	Fixed Income	CE Exchange-Listed Preferred & Hybrid Securities Index
DBC	Invesco DB Commodity Index Trac	Commodity	DBIQ Optimum Yield Diversified Commodity Index

Ticker	**Company**	**Sector**	**Industry**
AAPL	Apple	Consumer Goods	Electronic Equipment
AXP	American Express	Financial	Consumer Financial Services
BA	Boeing	Capital Goods	Aerospace & Defense
CAT	Caterpillar	Capital Goods	Construction & Agriculture Machinery
CSCO	Cisco Systems	Technology	Networking & Communication Devices
CVX	Chevron	Energy	Oil & Gas - Integrated
DIS	Disney	Services	Broadcasting
DOW	Dow	Basic Materials	Chemicals
GS	Goldman Sachs	Financial	Investment Brokerage - National
HD	Home Depot	Services	Retail (Home Improvement)
IBM	International Business Machines	Technology	Computer Hardware
INTC	Intel	Technology	Semiconductors
JNJ	Johnson & Johnson	Healthcare	Major Drugs
JPM	JP Morgan Chase	Financial	Money Center Banks
KO	Coca-Cola	Consumer/Non-Cyclical	Beverages (Non-Alcoholic)
MCD	McDonald's	Services	Restaurants
MMM	3M	Conglomerates	Conglomerates
MRK	Merck	Healthcare	Major Drugs
MSFT	Microsoft	Technology	Software & Programming
NKE	Nike	Consumer Goods	Textile - Apparel Footwear & Accessories
PFE	Pfizer	Health Care	Major Drugs
PG	Procter & Gamble	Consumer/Non-Cyclical	Personal & Household Products
TRV	The Travelers Companies	Financial	Property & Casualty Insurance
UNH	Unitedhealth Group	Healthcare	Health Care Plans
UTX	United Technologies	Conglomerates	Conglomerates
V	Visa	Financial	Credit Services
VZ	Verizon	Services	Communications Services
WBA	Walgreens	Services	Pharmaceutical Retailers
WMT	Wal-Mart	Services	Retail (Department & Discount)
XOM	ExxonMobil	Energy	Oil & Gas - Integrated

I have constructed and analyzed 4 long-only asset allocation strategies that are used as a good benchmarking tool for the HRP approach: The Minimum Variance portfolio (MV) or CLA, the Inverse Variance portfolio (IV), the Equal Weighted portfolio (EW) and a Random Weighted portfolio (RDM), where the weights are randomly chosen according to the total number of simulations, which I set equal to 10000. Subsequently I have applied these strategies on the two constructed portfolios. The main goal of the final analysis is to compare the two Hierarchical Risk Parity portfolios (the Dow Jones Index one and the one built with the fifteen ETFs), with other three traditional portfolio allocation approaches in terms of profitability, diversification and risk minimization. I have personally encoded all the analysis on the Anaconda Jupyter notebook of Python (vv.3.7).

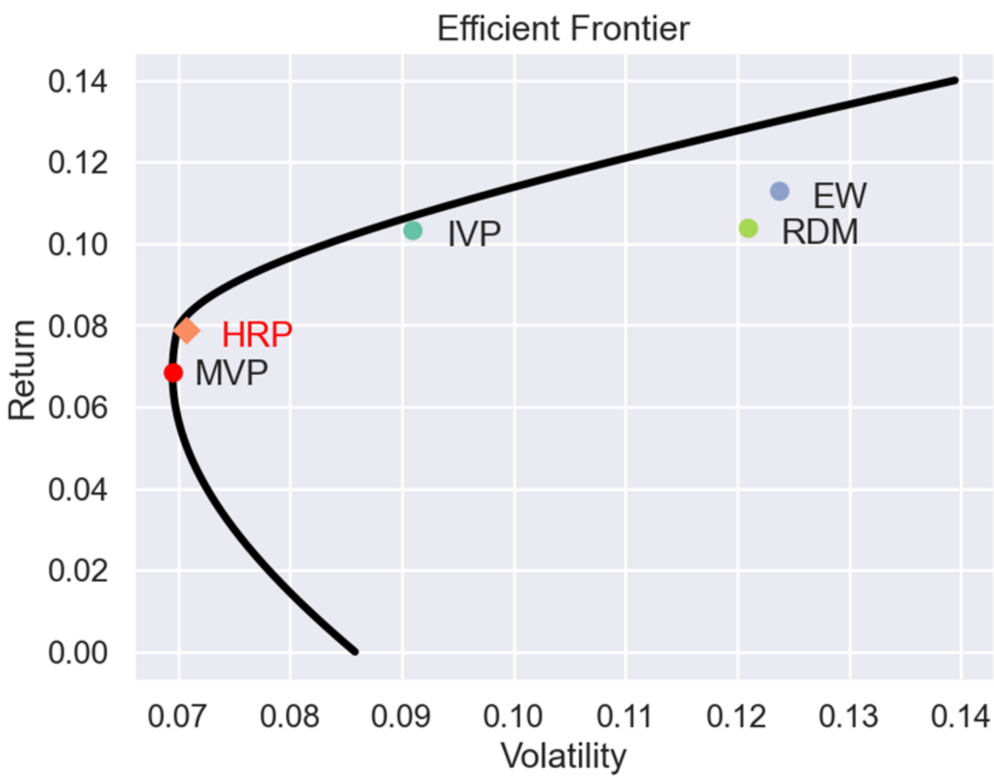
The Dow Jones Index Hierarchical Risk Parity (HRP) portfolio notably manages to strengthen its robustness in the out-of-sample test. Not only, in line with de Prado's findings, it achieves the lowest annualized volatility outperforming even the CLA solution, but it also achieves a very good Sharpe ratio result[125]. In the out-of-sample test, the MV portfolio seems to be subject to random shocks, and therefore it is much more volatile as demonstrated by its cumulative return and annualized volatility's results.





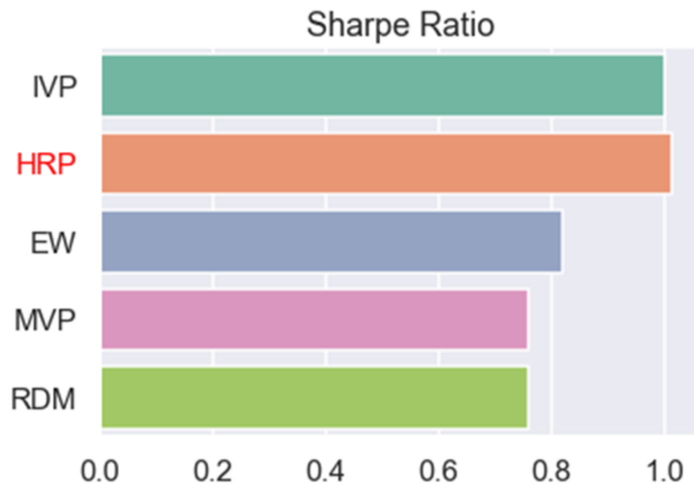
[125] Dow Jones Portfolio Sharpe ratio and annualized volatility. Data Source: A personal elaboration on Python.

For what concerns the all ETFs HRP portfolio, the in-sample efficient frontier, shows the HRP portfolio is the only one, together with the MV portfolio, to lie on the line[126].



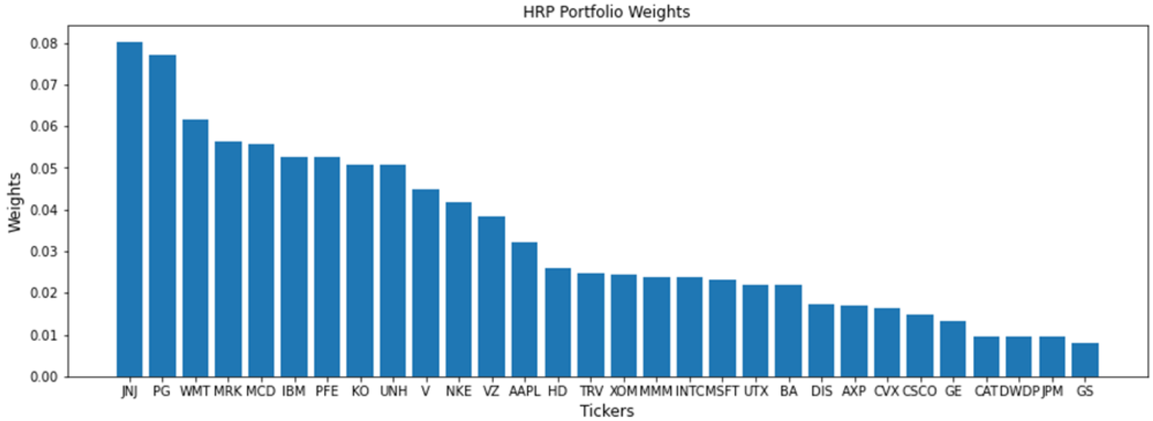
Though it does not manage to beat the MV portfolio in terms of risk minimization out-of-sample, it still achieves a very low level of volatility, and it presents the highest level of Sharpe ratio compared to the other allocation approaches[127].

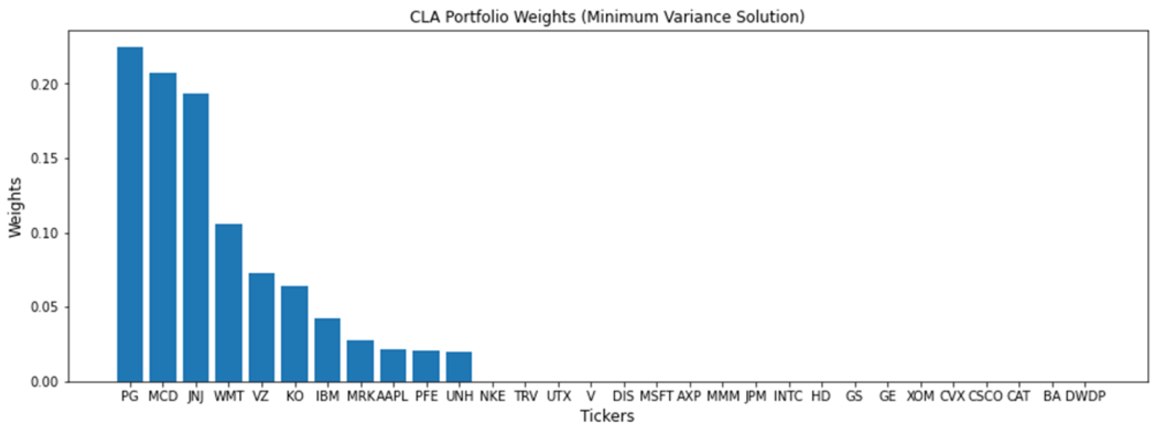
[126] In-sample ETF portfolio efficient frontier. Data Source: A personal elaboration on Python.
[127] ETF Portfolio Sharpe ratio and annualized volatility. Data Source: A personal elaboration on Python.



Both the Dow Jones Index and the all ETFs HRP portfolios exhibit the lowest level of maximum drawdown, a very low level of Value at Risk (even if higher than the one achieved with the minimum variance approach) and a very good return distribution as well. Furthermore, the HRP algorithm, allows for a much more uniform weight allocation especially if compared with the portfolios built with the CLA algorithm, which literally concentrate 75% of their holdings on the top 4-5 investments. This means the HRP methodology allows for a better risk diversification[128].

[128] Dow Jones HRP & CLA portfolio weights distribution. Data Source: A personal elaboration on Python.





From the empirical results, therefore, it is possible to conclude the HRP strategy seems a good approach to obtain well diversified portfolios. In addition to that, it leads to an optimization of the risk and performance metrics, especially in the out-of-sample test.

Nonetheless, there are few features that might be implemented to improve the model presented in this thesis I leave as a suggestion for further research. First, the HRP portfolio allocation approach allows only for long positions. Therefore, short sales and capital gains taxes can be added in the analysis. Secondly, new portfolio allocation strategies can be proposed for benchmarking the HRP portfolio. For instance, a Maximum Diversification portfolio (MD) and an Equal Risk Contribution portfolio (ERC) can be used as a good comparison tool. Then, the HRP method still does not produce fully diversified portfolios, with certain securities exhibiting the greatest weight for an extended period of time. Lastly, it would be interesting to evaluate the HRP performance out-of-sample via Monte Carlo as done by de Prado in his paper "*Building Diversified Portfolios that Outperform Out-of-Sample*". Indeed, a Monte Carlo experiment might provide even a better result in terms of variance minimization when the model is tested out-of-sample.

BIBLIOGRAPHY

Bai, Z. D., & Yin, Y. Q. (2008). Limit of the smallest eigenvalue of a large dimensional sample covariance matrix. In *Advances In Statistics* (pp. 108-127).

Bailey, D. H., & Lopez de Prado, M. (2012). Balanced baskets: a new approach to trading and hedging risks. *Journal of Investment Strategies (Risk Journals)*, *1*(4).

Bailey, D. H., & López de Prado, M. (2013). An open-source implementation of the critical-line algorithm for portfolio optimization. *Algorithms*, *6*(1), 169-196.

Bartholdy, J., & Peare, P. (2005). Estimation of expected return: CAPM vs. Fama and French. *International Review of Financial Analysis*, *14*(4), 407-427.

Benartzi, S., & Thaler, R. H. (2001). Naive diversification strategies in defined contribution saving plans. *American economic review*, *91*(1), 79-98.

Burggraf, T., & Vyas, A. (2020). Beyond Risk Parity-A Machine Learning-based Hierarchical Risk Parity Approach on Cryptocurrencies. Available at SSRN 3534773.

Burns, P. (2007). Random portfolios for performance measurement. In *Optimisation, Econometric and Financial Analysis* (pp. 227-249). Springer, Berlin, Heidelberg.

Chan, L. K., Karceski, J., & Lakonishok, J. (1999). On portfolio optimization: Forecasting covariances and choosing the risk model. *The review of Financial studies*, *12*(5), 937-974.

Choueifaty, Y., & Coignard, Y. (2008). Toward maximum diversification. *The Journal of Portfolio Management*, *35*(1), 40-51.

Clarke, R., De Silva, H., & Thorley, S. (2013). Risk parity, maximum diversification, and minimum variance: An analytic perspective. *The Journal of Portfolio Management*, *39*(3), 39-53.

Cottle, R. W., & Infanger, G. (2010). Harry Markowitz and the Early History of Quadratic Programming. In *Handbook of Portfolio Construction* (pp. 179-211). Springer, Boston, MA.

De Prado, M. L. (2016). Building diversified portfolios that outperform out of sample. *The Journal of Portfolio Management*, *42*(4), 59-69

De Prado, M. L. (2018). *Advances in financial machine learning*. John Wiley & Sons.

Demey, P., Maillard, S., & Roncalli, T. (2010). Risk-based indexation. *Available at SSRN 1582998*.

DeMiguel, V., Garlappi, L., & Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy?. *The review of Financial studies*, *22*(5), 1915-1953.

Elton, E. J., & Gruber, M. J. (1997). Modern portfolio theory, 1950 to date. *Journal of Banking & Finance*, *21*(11-12), 1743-1759.

Elton, E.J., Gruber, M.J. (2011). “Modern Portfolio Theory and Investment Analysis”, *Wiley*, 9.

Fama, E. F., & French, K. R. (1992). The cross-section of expected stock returns. *the Journal of Finance*, *47*(2), 427-465.

Fama, E. F., & French, K. R. (2004). The capital asset pricing model: Theory and evidence. *Journal of economic perspectives*, *18*(3), 25-46.

Friend, I., & Blume, M. (1970). Measurement of portfolio performance under uncertainty. *The American economic review*, *60*(4), 561-575.

Füss, R., Kaiser, D. G., & Schindler, F. (2007). Dynamic linkages between hedge funds and traditional financial assets: Evidence from emerging markets.

Goetzmann, W., Ingersoll, J., Spiegel, M. I., & Welch, I. (2002). Sharpening sharpe ratios. *National bureau of economic research.*

Harry Markowitz (1952). "Portfolio Selection", *The Journal of Finance*, 7(1), 77-91.

Hsu, J. C. (2004). Cap-weighted portfolios are sub-optimal portfolios. *Journal of investment Management*, *4*(3).

Huberman, G., & Jiang, W. (2006). Offering versus choice in 401 (k) plans: Equity exposure and number of funds. *The Journal of Finance*, *61*(2), 763-801.

Hurst, B., Johnson, B., & Ooi, Y. H. (2010). Understanding risk parity. *AQR Capital Management*.

Idzorek, T. (2007). A step-by-step guide to the Black-Litterman model: Incorporating user-specified confidence levels. In *Forecasting expected returns in the financial markets* (pp. 17-38). Academic Press.

Jaeger, M., Krügel, S., Marinelli, D., Papenbrock, J., & Schwendner, P. (2020). Understanding Machine Learning for Diversified Portfolio Construction by Explainable AI. *Available at SSRN*.

Jain, P., & Jain, S. (2019). Can Machine Learning-Based Portfolios Outperform Traditional Risk-Based Portfolios? The Need to Account for Covariance Misspecification. *Risks*, *7*(3), 74.

Kempf, A., & Memmel, C. (2006). Estimating the global minimum variance portfolio. *Schmalenbach Business Review*, *58*(4), 332-348.

Khokhlov, V. (2011). An Algorithm for Finding a Portfolio with the Highest Sharpe Ratio. *Available at SSRN 1767338*.

Lintner, J. (1965). Security prices, risk, and maximal gains from diversification. *The journal of finance*, *20*(4), 587-615;

Magdon-Ismail, M., & Atiya, A. F. (2004). Maximum drawdown. *Risk Magazine*, *17*(10), 99-102.

Maillard, S., Roncalli, T. and Teïletche, J. (2010). The Properties of Equally Weighted Risk Contribution Portfolios. The Journal of Portfolio Management, vol. 36(4), pp. 60-70.

Markowitz, H. (1955). The optimization of a quadratic function subject to linear constraints. *RAND CORP, (68). Santa Monica (CA).*

Merton, R. C. (1980). *On estimating the expected return on the market: An exploratory investigation* (No. w0444). National Bureau of Economic Research; Pratt, S. P., & Grabowski, R. J. (2008). *Cost of capital*. John Wiley & Sons..

Michaud, R. O., & Michaud, R. (2007). Estimation error and portfolio optimization: a resampling solution. *Available at SSRN 2658657*.

Mossin, J. (1966). Equilibrium in a capital asset market. *Econometrica: Journal of the econometric society*, 768-783.

Niedermayer, A., & Niedermayer, D. (2010). Applying Markowitz's critical line algorithm. In *Handbook of portfolio construction* (pp. 383-400). Springer, Boston, MA.

Pfitzinger, J., & Katzke, N. (2019). *A constrained hierarchical risk parity algorithm with cluster-based capital allocation* (No. 14/2019).

Qian, E. E. (2005). On the financial interpretation of risk contribution: Risk budgets do add up. *Available at SSRN 684221*.

Raffinot, Thomas, The Hierarchical Equal Risk Contribution Portfolio (August 23, 2018). Available at SSRN: https://ssrn.com/abstract=3237540 or http://dx.doi.org/10.2139/ssrn.3237540

Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of risk*, *2*, 21-42.

Roncalli, T., & Weisang, G. (2016). Risk parity portfolios with risk factors. *Quantitative Finance*, *16*(3), 377-388.

Scherer, B. (2007). Portfolio construction and risk budgeting. *Edhec Business School, 5.*

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *The journal of finance*, *19*(3), 425-442.

Sharpe, W. F. (1994). The sharpe ratio. *Journal of portfolio management*, *21*(1), 49-58.

Simon, H. A. (1991). Organizations and markets. *Journal of economic perspectives*, *5*(2), 25-44.

Simon, H. A. (1991). The architecture of complexity. In *Facets of systems science* (pp. 457-476). Springer, Boston, MA.

Sironi, A., & Resti, A. (2007). Risk management and shareholders' value in banking: from risk measurement models to capital allocation policie. *John Wiley & Sons,*(417).

Smyth, C. J. (1986). A coloring proof of a generalisation of Fermat's Little Theorem. *The American Mathematical Monthly*, *93*(6), 469-471.

Sortino, F., & Price, L.N. (1994). Performance Measurement in a Downside Risk Framework. *Journal of Investing*, 3(3), 50-58.

Williams, J. B. (1938). “The theory of investment value” (No. HG4521 W48).

Zakamulin, V. (2015). A test of covariance-matrix forecasting methods. *The Journal of Portfolio Management*, *41*(3), 97-108.

Zweig J. (1998). “Five investing lessons from America’s top pension fund”, *Money*, 115-118.

SITOGRAPHY

http://gautier.marti.ai/qfin/2018/10/02/hierarchical-risk-parity-part-1.html.
http://www.thierry-roncalli.com/download/erc-slides.pdf

https.://www.nber.org/cycles.html

https://amers2.apps.cp.thomsonreuters.com/web/cms/?navid=45050

https://analystprep.com/cfa-level-1-exam/portfolio-management/minimum-variance-portfolios/

https://breakingdownfinance.com/finance-topics/alternative-investments/maximum-diversification-portfolio/

https://breakingdownfinance.com/finance-topics/modern-portfolio-theory/hierarchical-risk-parity/

https://en.wikipedia.org/wiki/Dow_Jones_Industrial_Average

https://en.wikipedia.org/wiki/Lagrange_multiplier#Interpretation_of_the_Lagrange_multipliers

https://en.wikipedia.org/wiki/Monte_Carlo_method.
https://en.wikipedia.org/wiki/Sharpe_ratio

https://github.com/hudson-and-thames/mlfinlab/blob/master/mlfinlab/portfolio_optimization/hrp.py

https://github.com/hudson-and-thames/research/blob/master/Chapter16/Chapter16.ipynb
https://github.com/KennnnyZhou/Hierarchical_Risk_Parity/blob/master/Paper%20for%20803%20Project.pdf
https://github.com/KennnnyZhou/Hierarchical_Risk_Parity/blob/master/HRP.ipynb

https://github.com/robertmartin8/PyPortfolioOpt/blob/master/examples.py https://gmarti.gitlab.io/qfin/2018/10/15/hierarchical-risk-parity-part-3.html.

https://gmarti.gitlab.io/qfin/2018/10/15/hierarchical-risk-parity-part-2.html.

https://hudsonthames.org/an-introduction-to-the-hierarchical-risk-parity-algorithm/.

https://investresolve.com/blog/portfolio-optimization-simple-optimal-methods/.

https://kse.ua/wp-content/uploads/2019/03/Serhii-Liutov16.pdf.pdf

https://learn-eu-central-1-prod-fleet01-xythos.s3-eu-central-1.amazonaws.com/5b7140d096263/489538?response-content-disposition=inline%3B%20filename%2A%3DUTF-8%27%27ML%25202019_PM_lectures_2_3_From_MPT_to_AM%2520MSC%2520FIN.pdf&response-content-type=application%2Fpdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20200318T143902Z&X-Amz-SignedHeaders=host&X-Amz-Expires=21600&X-Amz-Credential=AKIAZH6WM4PLYI3L4QWN%2F20200318%2Feu-central-1%2Fs3%2Faws4_request&X-Amz-Signature=9787ef86167a4b6ece3e7f1a2c1669fc994270800238e3a6ec1a7bb11fc1efa4

https://medium.com/datadriveninvestor/hierarchical-risk-parity-in-portfolio-construction-fc368db18c78

https://qoppac.blogspot.com/2018/12/portfolio-construction-through_7.html

https://quantdare.com/hierarchical-risk-parity/

https://quantpedia.com/hierarchical-risk-parity/

https://systematicinvestor.wordpress.com/2013/03/22/maximum-sharpe-portfolio/

https://towardsdatascience.com/efficient-frontier-optimize-portfolio-with-scipy-57456428323e

https://towardsdatascience.com/efficient-frontier-portfolio-optimisation-in-python-e7844051e7f

https://www.interviewqs.com/blog/value_at_risk

https://www.investopedia.com/terms/s/sharperatio.asp

https://www.investopedia.com/terms/s/sortinoratio.asp

https://www.investopedia.com/terms/t/treynorratio.asp https://www.quantopian.com/posts/hierarchical-risk-parity-comparing-various-portfolio-diversification-techniques

https://www.quora.com/Why-does-value-at-risk-result-being-negative-value