
Erasmus Mundus QEM: Models and Methods of Quantitative Economics

Final Thesis

PORTFOLIO INSURANCE STRATEGIES:
An empirical analysis of a frontier market - the Vietnamese case

Supervisor
Ch. Prof. Antonella Basso

Graduand
Trieu Tien Dat
Matriculation Number 861160

Academic Year
2016 / 2017
TABLE OF CONTENTS

TABLE OF CONTENTS ................................................................. 2
LIST OF FIGURES ................................................................. 4
LIST OF TABLES ................................................................. 6
INTRODUCTION ........................................................................ 8

CHAPTER 1: PORTFOLIO INSURANCE ........................................... 10

1.1. Introduction of Portfolio Insurance ........................................... 10
1.2. The evolution of Portfolio Insurance ...................................... 12
1.3. Classification of Portfolio insurance strategies ......................... 22
    1.3.1. Buy and hold strategy (BH) ........................................... 22
    1.3.2. Constant Proportion portfolio insurance strategy (CPPI) .... 23
    1.3.3. Option-based portfolio insurance strategy (OBPI) ......... 26
    1.3.4. VaR-based portfolio insurance strategy (VaRPI) ........... 29
    1.3.5. Stop-loss portfolio insurance strategy (SLPI) ............... 32
    1.3.6. A comprehensive comparison of the popular portfolio insurance strategies 33

CHAPTER 2: METHODOLOGY OF PERFORMANCE COMPARISON
BETWEEN DIFFERENT PORTFOLIO INSURANCE STRATEGIES .......... 41

2.1. The design of simulation methods and data issues .................... 41
    2.1.1. Introduction to Monte Carlo simulation method and backtesting .... 41
    2.1.2. The simulation design .............................................. 44
2.2. Performance measures of Portfolio Insurance Strategies ........... 50
    2.2.1. First four moments of the returns distribution ................... 50
    2.2.2. Performance ratios .................................................. 53

CHAPTER 3: RESULTS AND DEFINITION OF THE DOMINANT STRATEGY 57

3.1. Results of Monte Carlo simulation method ............................ 57
    3.1.1. Bear market scenario .............................................. 58
3.1.2. Normal market scenario .......................................................... 62
3.1.3. Bull market scenario ............................................................. 65
3.2. Results of Backtesting ............................................................... 68
3.3. Comparison with different rebalancing frequencies ...................... 70

CONCLUSION AND FURTHER RESEARCH ........................................... 76
APPENDIX A ..................................................................................... 78
APPENDIX B ..................................................................................... 84
APPENDIX C ..................................................................................... 87
REFERENCES ................................................................................... 129
LIST OF FIGURES

Figure 1.1. Profit and loss of insured and uninsured portfolios: an example. (Source: Lee, Lee, & Lee, 2010) ................................................................. 12
Figure 1.2. Distribution of expected returns on insured and uninsured portfolios. (Source: Lee, Lee, & Lee, 2010) ................................................................. 12
Figure 1.3. Value at maturity of protective put .................................................. 27
Figure 1.4. Probability distribution for gain in portfolio value during time T, based on (Hull, 2012) ................................................................. 31
Figure 1.5. Alternative situation to Figure 1.4, VaR is the same, but the potential loss is larger, based on (Hull, 2012) .................................................................................. 31
Figure 2.1. A diagram of steps in a basic Monte Carlo simulation ........................................ 42
Figure 3.1. The performance of the portfolio insurance strategies for VN30EQ during the insurance period with one out of 10,000 random paths of Scenario 1...................... 59
Figure 3.2. The performance of the portfolio insurance strategies for VN30EQ during the insurance period with one out of 10,000 random paths of Scenario 2...................... 61
Figure 3.3. The performance of the portfolio insurance strategies for VN30EQ during the insurance period with one out of 10,000 random paths of Scenario 3 ...................... 63
Figure 3.4. The performance of the portfolio insurance strategies for VN30EQ during the insurance period with one out of 10,000 random paths of Scenario 4 ...................... 64
Figure 3.5. The performance of the portfolio insurance strategies for VN30EQ during the insurance period with one out of 10,000 random paths of Scenario 5 ...................... 66
Figure 3.6. The performance of the portfolio insurance strategies for VN30EQ during the insurance period with one out of 10,000 random paths of Scenario 6 ...................... 67
Figure 3.7. The performance of the portfolio insurance strategies for VN30EQ from the beginning of April 2012 to the end of April 2017 with Backtesting ...................... 69
Figure 3.8. The performance of the portfolio insurance strategies for VN30EQ from the beginning of April 2012 to the end of April 2017 with Backtesting (weekly rebalancing) .... 72
Figure 3.9. The performance of the portfolio insurance strategies for VN30EQ from the beginning of April 2012 to the end of April 2017 with Backtesting (daily rebalancing) ........ 74

Figure A. 1. Annualized mean distributions with market scenario 1 (μ = -15%, σ = 20%) .... 78
Figure A. 2. Annualized mean distributions with market scenario 2 (μ = -15%, σ = 40%) .... 79
Figure A. 3. Annualized mean distributions with market scenario 3 (μ = 8%, σ = 20%) ......80
Figure A. 4. Annualized mean distributions with market scenario 4 (μ = 8%, σ = 40%) ......81
Figure A. 5. Annualized mean distributions with market scenario 5 (μ = 15%, σ = 20%) .....82
Figure A. 6. Annualized mean distributions with market scenario 6 (μ = 15%, σ = 40%) .....83
LIST OF TABLES

Table 1.1. A review of the main literature on portfolio insurance ........................................ 21
Table 1.2. Summary of the considered aspects of the portfolio insurance strategies .......... 39
Table 1.3. Condensed summary of the portfolio insurance strategies ............................... 40
Table 2.1. Summary of the market scenarios ................................................................. 49
Table 3.1. Comparison of the performance of the strategies in Scenario 1 ......................... 58
Table 3.2. Comparison of the performance of the strategies in Scenario 2 ....................... 60
Table 3.3. Comparison of the performance of the strategies in Scenario 3 ....................... 63
Table 3.4. Comparison of the performance of the strategies in Scenario 4 ....................... 64
Table 3.5. Comparison of the performance of the strategies in Scenario 5 ....................... 65
Table 3.6. Comparison of the performance of the strategies in Scenario 6 ....................... 67
Table 3.7. Comparison of the performance of the strategies in the real situation of VN30EQ from April 2012 to April 2017 ................................................................. 68
Table 3.8. Comparison of the performance of the strategies in Scenario 2 with weekly rebalancing ................................................................. 70
Table 3.9. Comparison of the performance of the strategies in Scenario 4 with weekly rebalancing ................................................................. 71
Table 3.10. Comparison of the performance of the strategies in Scenario 2 with daily rebalancing ................................................................. 73
Table 3.11. Comparison of the performance of the strategies in Scenario 4 with daily rebalancing ................................................................. 73

Table A.1. The comparison of the AMean of the strategies across market scenarios ...... 84
Table A.2. The comparison of the AStd of the strategies across market scenarios ...... 84
Table A.3. The comparison of the AKurt of the strategies across market scenarios ...... 84
Table A.4. The comparison of the ASkew of the strategies across market scenarios ....... 85
Table A.5. The comparison of the Sharpe ratio of the strategies across market scenarios ... 85
Table A.6. The comparison of the Sortino ratio of the strategies across market scenarios ... 85
Table A.7. The comparison of the Omega ratio of the strategies across market scenarios ... 86
Table A.8. The comparison of the Floor Protection ratio of the strategies across market scenarios ...................................................................................... 86
Table A. 9. The comparison of the rank of the strategies across market scenarios and the Overall rank

<table>
<thead>
<tr>
<th>Market Scenario</th>
<th>Strategy 1</th>
<th>Strategy 2</th>
<th>Strategy 3</th>
<th>Strategy 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Medium</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Low</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Overall rank: 86
INTRODUCTION

Insurance was invented to adapt the inherent need of humans to get rid of losses. In the field of investments, full of risks, Portfolio Insurance is inevitably necessary for prudent investors. However, due to specific characteristics compared to other activities that insurance had already covered, insurance for portfolio did not exist until 1976, when Hayne Leland found out the idea and commercialize it in 1981 to investors as a suitable technique to avoid downside risk, while not to miss the gains from the favorable trend of the market. Since then, Portfolio Insurance became increasingly famous amongst investors, but along with program trading, it was accused of the stock market crash of October in 1987 and, apparently, lost its popularity right after that. Nevertheless, any portfolio should be still insured somehow, especially for risk-averse investors. Therefore, there was considerable interest in the improvement of Portfolio Insurance strategies so as not to repeat that sad history.

The main goal of this research is to compare the merits and choose the dominant strategy with regards to some widely-adopted statistical measures and performance ratios in the case of a frontier stock market like Vietnam, which attracts virtually no attention from the academics and practitioners in the field of portfolio insurance. Since most of the prior studies do not compare all the of strategies mentioned in this work, we intend to contribute to the literature of the field with a more extensive comparison among the most popular (from standard to more refined) portfolio insurance strategies, namely Buy and Hold, Constant Proportion Portfolio Insurance strategy with two different multipliers, Option-based Portfolio Insurance strategy, VaR-based Portfolio Insurance strategy, and Stop-loss Portfolio Insurance strategy.

We carry out a Monte Carlo simulation analysis in total six market scenarios composing of 3 types of market (bull, no-trend and bear markets) in combination with two different market volatilities (low and high). We also implement the backtesting of the VN30 Total Return Equal Weight Index (VN30EQ Index) in the insurance period of 5 years from 2012 to 2017 to verify how the mentioned portfolio insurance strategies would have behaved if we had invested in these strategies at the beginning of the insurance period. However, in general, the VN30EQ Index experienced an uptrend throughout the 5-year insurance period. Therefore, in each of the six market scenarios, we also choose one out of 10,000 paths of our Monte Carlo simulations and implement them like the backtesting method to give a broader view to see if the insurance strategies really insure the underlying portfolio.

The structure of the Thesis is organized as follows. In the first chapter, we discuss the history of Portfolio Insurance and the development of researches in the field, before going into
the technical details of the aforementioned portfolio insurance strategies. To conclude the chapter, we provide a comparison table on some main aspects to highlight the relationships between the strategies.

The second chapter explains our methodology used to implement the comparison among the strategies. In the first half of the chapter, we introduce the Monte Carlo simulation and backtesting methods, by which we perform our comparison in this study, prior to giving information about the data and our simulation setup. We also provide a brief history and description of the Vietnamese stock market. In the other half of the chapter, we discuss the performance measures that we use in our study, such as the first four moments, the Sharpe, Sortino, Omega and the protection ratios.

In the last chapter, the results of our study are presented as regards the case of 6 market scenarios and the real path of the VN30EQ Index. The result tables provide the values of the performance measures for our portfolio insurance strategies. Unlike most of the previous studies, based on the values, we give the explicit overall rank for all strategies in each case. The graphs supplement the results, to furnish the reader a demonstrative way to check if the insurance strategies requirements are achieved in all cases.

Furthermore, in APPENDIX A, we also provide tables of histograms to give a full picture of the shape of the expected returns distributions of the portfolio insurance strategies in the 6 Monte Carlo simulated scenarios. In APPENDIX B, there are various tables in order to help readers easily compare the performance measures and the overall ranking of all strategies across market scenarios. We also provide the Python codes that we have written to serve the needs of our research in monthly rebalancing case in APPENDIX C.
CHAPTER 1: PORTFOLIO INSURANCE

1.1. Introduction of Portfolio Insurance

The idea about portfolio insurance was originated by Leland in the second half of 1970's and appeared as a financial product in the early 1980's (Luskin, 1988). Regardless of being associated with the Black Monday in 1987, in recent years, after the huge stock market downturn in 2008, portfolio insurance has received a lot of attention, not only from the institutional and individual investors but also from academia. The history and development of the portfolio insurance strategies will be discussed in more detail in the next section. So far, we think it is important to grasp the basic concepts of portfolio insurance first.

Portfolio insurance is defined as a dynamic hedging process which guarantees the investor with an acceptable minimum level of wealth for his portfolio of risky assets while allowing him to participate in the potential gains to maximize his portfolio’s returns over a predetermined investment horizon.

Portfolio insurance is called a dynamic hedging process, because of its continuous adjustments to hedge the portfolio against frequent changes of the prices in the stock market. The dominance in the capital protection of a dynamic hedging process over the fixed-allocation strategy can be illustrated by a vivid situation mentioned in Grannis (1988). There are two drivers racing via Pacific Coast Highway in the two same cars, except that the brake on one car is not good. It is similar to the situation where there are two investors with two identical underlying portfolios of risky assets, except that one investor has a good portfolio insurance “brake” and the other does not. The investor with the “good brake” can speed up on the straightaways, meaning he can invest more money in the underlying portfolio to obtain more profit on the growing market. Furthermore, the investor – “racer with good brake” can slow down on the curves by shifting the funds from the risky assets to the risk-free assets, when the market falls. Apparently, both cars encounter the same risk of taking a turn too fast, but the driver with good brake or the investor armed with portfolio insurance can adjust his speed depending on which type of road he is driving on, rising or falling market. Conversely, the poor-brake car cannot do likewise, he has to keep his speed relatively slow and constant through the journey so as not to shoot off the edge of the mountain. Theoretically, the investor armed with portfolio insurance is most likely to win. Perhaps just in extreme cases, for example, of

---

1 The investor, who does not have the “good brake”, invests in only risky assets. This strategy is called No hedge strategy.
there being nearly all straightaways throughout the race, the poor-brake car can win. It is because the friction of the brake (as a metaphor for portfolio insurance cost) can cause the other car a bit slow down or making the investor with portfolio insurance obtain less profit in the almost-rising market over the given period of investment.

The risky assets in the definition refer to the traditional assets (namely stocks, high-yield bonds), and the alternative assets (such as commodities, real estate, foreign exchange). We can consider portfolio insurance as a composition of the underlying risky portfolio and an insurance policy to absorb a specified loss of that underlying portfolio. Should the price of the needed-to-be-insured underlying portfolio decrease, the insurance policy’s value will increase to help compensate the decline in the price of that underlying portfolio. In a better scenario when the value of the underlying portfolio increases, the value of the insured portfolio still increases; however, less than the increase in the risky (uninsured) underlying portfolio itself.

We use the simple example in Handbook of Quantitative Finance and Risk Management (Lee, Lee, & Lee, 2010) to make things clearer. An investor purchased an EUR 95 underlying portfolio and an EUR 5 insurance policy to guarantee that the underlying portfolio would never fall below EUR 95 in value, while the uninsured portfolio’s value could decrease unlimitedly to 0 in the worst case. If the value of the underlying portfolio increases, the value of the global insured portfolio will increase; however, as much as that of the uninsured one less the insurance premium. Figure 1.1 illustrates the profit and loss of the underlying portfolio with and without the insurance policy.

Furthermore, with the portfolio insurance, the investor can reshape the probability distribution of the returns according to his risk appetite. Figure 1.2 illustrates the expected return distribution of both portfolios with and without the insurance policy, assuming that expected returns follow a normal distribution. We can see that the uninsured portfolio has greater profit as well as greater downside risk, while the left half of the insured portfolio’s return distribution is truncated and the remaining right half moves leftwards with the distance equaling the cost of the insurance. It means that not only the downside loss of the insured portfolio is limited, but its upside potential is always lower than that of the uninsured portfolio. This is the cost of portfolio insurance that the investor has to pay when prices increase. Risk-averse investors would tend to choose the portfolio insurance to limit their risks, whereas risk-loving investors would prefer the uninsured portfolio to seek a higher profit.
The idea about Portfolio insurance was thought by Hayne E. Leland at September 11, 1976 (Leland & Rubinstein, 1988). As a consequence of the 1973–74 stock market crash (due to the collapse of the Bretton Woods system, 1973 oil crisis and “Nixon shock”), many pension funds ran away from the declining stock market. Leland thought if there had been an insurance for their portfolio, those pension funds would not have sold out their stocks at crisis. Subsequently, they felt regret and needed to buy back the stocks they sold one year earlier with much higher price in the bull market in 1975. The portfolio insurance seemed to be appealing to investors; however, the problem was how to create it.
In contrast with regular insurance, for say houses and cars, where only one or a few insured objects are destroyed by separate accidents, whereas when there is an “accident” in the stock market, it would be a nightmare for the underwriters of portfolio insurance. Since the chosen for insured portfolio of an investor is well-diversified, which eliminates all the specific risks, there are only systemic risk or market risk to be hedged. Hence, when the market goes down, all the portfolios will have similar “accidents” and the insurance cannot afford for all of the compensations. In short, the usual insurance method “risk pooling” cannot be applied, since the more participants in the pool, the greater the risk the underwriters have to bear.

In 1976, although Leland was still not an expert in options, he knew that a put option on an entire portfolio could be an equivalent to the insurance policy that he wanted to create. Thanks to the arbitrage argument used by Black and Scholes to price options in a famous paper (Black & Scholes, 1973), Leland thought a European put option could be purchased, or it could be replicated as a synthetic put of any particular strike price and maturities which might be not available in the option market by a simple dynamic strategy which involved the underlying portfolio and a risk-free asset. This is the original idea of Option-based portfolio insurance (OBPI), and not until February 1981, when the Los Angeles based Leland O’Brien Rubinstein Associates, Inc. (LOR) was incorporated, the first OBPI products were commercialized.

As an alternative portfolio insurance strategy to the more complex one of OBPI, Constant Proportion Portfolio Insurance (CPPI) was first publicly introduced in “Simplifying portfolio insurance” of Black and Jones (1987). In this article, the authors offer an easier approach for investors to understand and implement. Regardless of not involving any complex formulas, this method is still flexible enough to be designed for any investor’s objectives even in the most extreme cases. According to the two seminal articles by Black and Jones (1987) and Black and Perold (1992), the amount invested in the underlying portfolio is a linear function of a cushion - the difference between the value of the insured portfolio and the floor (the predetermined lowest acceptable value of the portfolio for the investor). Unlike the OBPI, the floor set by the investor may grow at the risk-free rate over time. However, both strategies guarantee downside protection by optimally shifting the investment from the underlying portfolio into the risk-free assets when the market decreases, and still participate in the market when it rises by moving the investment from risk-free assets into the underlying portfolio.

---

1 In fact, all the portfolio insurance programs and licensees, which covered USD 50 billion of stocks of Leland’s company providing portfolio insurance dropped drastically to USD 10 billion right after the “accident” of Black Monday in 1987.
Unfortunately, the credibility of portfolio insurance was significantly damaged by the 1987 stock market downturns when the Dow Jones Industrial Average dropped by more than 20% on the merely Black Monday, October 19 and the underwriters of portfolio insurance failed to provide the required protection. There was already a market decline during the period from Wednesday to Friday of the previous week (October 14 to October 16), and as a consequence, the sales of USD 12 billion of stocks or index futures were programmed before the market opening on the following Monday, October 19. Nevertheless, the portfolio insurance companies could sell only USD 4 billion due to the overloaded stock exchange system. It stemmed from the massive queue to sell off the risky assets of the investors, who anticipated the actions of the portfolio insurers. The portfolio insurance was therefore blamed for exacerbating the decline in stock prices. Needless to say, the popularity of the portfolio insurance products (or at least its name\(^1\)) has dropped significantly right after this turmoil.

So, “does portfolio insurance have a future?” as asked in the conclusion “After the fall” of Luskin’s collection of the best current thinking about portfolio insurance and dynamic hedging (Luskin, 1988). Luskin answered himself that “Though many of its weaknesses emerged in high relief as the result of the stress-testing of the crash, portfolio insurance has shown itself to be a robust strategy. Nonetheless, after the crash, an investor contemplating a program of portfolio insurance has many uncertainties to face. These uncertainties have always been a part of portfolio insurance, but they can now no longer be ignored.” Indeed, these uncertainties have been identified by other researchers, such as uncertain interest rates, uncertain volatility, security price jumps, security mispricing and transaction costs (Rubinstein, Alternative Paths to Portfolio Insurance, 1988) and volatility misestimation (Rendleman & Thomas J., 1990). These real-world complications are exactly what are assumed away in the Black-Scholes-Merton option pricing theory, on which the original portfolio insurance strategy is based. Luskin continued “[T]hese strategies (refer to portfolio insurance and dynamic hedging) are fundamental means by which investors seek to control their pattern of investment outcomes. Now that investors have been shown that they possess this control, they will never willingly give it up.” (Luskin, 1988).

Despite its vicissitude in the previous century, portfolio insurance has re-emerged since the 2000s thanks to small structuring and trading costs, as well as the broadening in the asset classes which needed principal protection (Pain & Rand, 2008). There has been more and more

---

\(^1\) Recently, portfolio insurance is often referred to as capital protection strategy or principal-protected products in practice. See (Allianz SE, 2013), (Wealth Focus Pty Ltd., 2010), and (Pain & Rand, 2008).
literature on portfolio insurance and typically large banks, especially in Europe, still offer what they call principal-protected products, which are based on portfolio insurance rules, to high-net-worth individuals, private banks and institutional investors (Pain & Rand, 2008). Moreover, according to Pain and Rand (2008), the traditional OBPI investments have not been particularly common due to its complexity to be explained to investors, whereas the CPPI investments have become much more popular and have been refined to incorporate with additional variations in, for example, floor, multiplier, and exposure to risky assets.

Apart from the two above-mentioned most popular portfolio insurance strategies, OBPI and CPPI, there are still other alternative strategies. However, all of these strategies have a common convex payoff function (Lee, Lee, & Lee, 2010), which stems from the “buy high, sell low” rule for the underlying portfolio of risky assets. This rule seems to be contrary to many practitioners’ intuition as they often “buy low, sell high”\(^1\), meaning that they buy more risky assets at a low price when the market is expected to rise, and they sell off a fraction of the risky assets being considered at a high price before the market falls. Actually, the “whipsaw” movement of market price can be another case when the portfolio insurance strategies perform badly. It is just as the car with good brake, mentioned in the previous section’s example speeds up and then has to brake hard after every consecutive bend in a sinuous road. The other popular portfolio insurance strategies are the static Stop-loss portfolio insurance strategy (SLPI)\(^2\) (see (Rubinstein, 1988), (Bird, Dennis, & Tippett, 1988)) and the comparatively newer Portfolio Insurance strategy with Downside Risk Control, which includes two approaches due to its two measures of downside risk: Value-at-Risk (VaR) and Expected Shortfall (ES)\(^3\). Last but not least, although the static Buy and hold strategy (BH) is not often categorized as a portfolio insurance strategy, it is usually used as a benchmark to evaluate and compare the effectiveness of other portfolio insurance strategies. (see (Annaert, Van Osselaer, & Verstraete, 2009), (Cesari & Cremonini, 2003), (Costa & Gaspar, 2014), (Ho, Cadle, & Theobald, 2011), and (Khuman, Maringer, & Constantinou, 2008)). This strategy keeps the allocation between the underlying portfolio of risky assets and the riskless assets fixed over the whole investment period. A related point to consider is that unlike other portfolio insurance strategies, its payoff function is linear. That is the reason why BH is occasionally sorted out

---

\(^1\) This intuition is similar to the Constant Mix dynamic allocation strategy, which has a concave payoff function.

\(^2\) If we consider static strategy as the simplest form of dynamic one.

\(^3\) We will talk more about these two terms in subsection 1.3.4.
from the group of portfolio insurance strategies which have convex payoff functions. We will delve into these strategies in the following section.

There is a growing number of portfolio insurance strategies, so it is natural to find out which one outperforms the others and in which market scenario. Since the late 1980s, after the birth of the second strategy - CPPI in 1987 by Black and Jones, till early 1990s, most of the studies of the field tend to focus on comparisons of the performances between the existing strategies, i.e. OPBI, CPPI, and SLPI.

Typically, Zhu and Kavee (1988) first analyze the performances of the CPPI and the older strategy - OBPI. Using the Monte Carlo simulation method to generate time series with normally distributed daily returns of 15% and five different volatility levels ranging from 10% to 30% (step: 5%), they find out that in case of low volatility, both strategies succeed to reshape the return distribution in order to reduce downward risk, but allow some participation in the upward trend. Unfortunately, in the case of high market volatility, when the investors need the portfolio insurance the most, neither strategies can guarantee the floor properly (Zhu & Kavee, 1988).

In an article by Benninga (1990), he adds SLPI to the comparison. He employs Monte Carlo simulation with one set of parameters (mean return = 10%, volatility = 20% and transaction cost = 0.5%). Finally, he finds that as for both expected terminal wealth and Sharpe ratio, the simple SLPI seems to dominate both OBPI and CPPI for all floor values, let alone taking into consideration its comparatively low level of transactions costs (Benninga, 1990).

Since the turn of the new century, there has been a re-emergence of studies in the field of portfolio insurance, which provide new strategies, or variations of the existing portfolio insurance ones, or develop new performance measures. As a result, there has been an increasing interest in comparing them with both the traditional and recent performance measures using Monte Carlo simulation, historical simulation methods and backtesting. To the best of our knowledge, due to the complication of reality in the financial world, so far there has not been yet a formal consensus about the effectiveness of portfolio insurance. Despite the controversial (and sometimes not clear and not explicit) research results, we would like to enumerate the most prominent recent research works to give a literature review of what has been done from 2000.

Cesari and Cremonini (2003) compare extensively nine different popular strategies of dynamic asset allocation, including BH, Constant mix, CPPI and CPPI with a lock-in of profit, three modifications of OBPI and two kinds of SLPI, using eight measures for risk, return and
risk-adjusted performance\(^1\). Furthermore, these strategies are compared in nine different market scenarios (no-trend (returns in range of \((-5\%,\ 5\%))\), bull (5\%,\ 30\%), bear (-30\%,\ -5\%) markets, each of which paired with low (in range of (10\%,\ 15\%)), medium (15\%,\ 25\%), and high (25\%,\ 30\%) volatilities), taking into account transaction costs and discrete rebalancing of portfolios. The Monte Carlo simulations show that there is a dominant role of the two CPPI strategies in bear and normal markets, and the Constant mix in a bull market is preferred, while two strategies of CPPI and one variation of OBPI - BCDT are the best strategies regardless of the market scenarios. These results seem to be independent of the volatility level and the risk-adjusted performance measures used. However, regarding backtesting of stock market indices across the world: MSCI World, North America, Europe, and Pacific, no strategy is found to be dominant in all market situations (Cesari & Cremonini, 2003).

Unlike Cesari and Cremonini, in a very famous paper in the field, Bertrand and Prigent (2005) focus on only the two standard strategies: OBPI and CPPI. They have shown the drawback of using criteria, namely first order stochastic dominance and the first four moments of the rates of return to compare the two strategies due to the non-linearity property of the payoff functions. However, alongside the four moments, they develop techniques to measure their performance, such as the comparison of the dynamics of both methods. They also compare the dynamics of both approaches and particularly prove that the OBPI strategy is equivalent to the CPPI one in the Black and Scholes framework, with the multiple, being the decreasing function of the risky asset value. They also conclude that the OBPI dominates the CPPI in a normal market, or when there exists a sudden drop in the risky asset value when it is high. In the case of a bull market and no drop during the investment period, the value of CPPI is greater than that of the OBPI (Bertrand & Prigent, 2005).

In their working paper, Khuman et al. (2008) focus on only CPPI, which recently has gained popularity in both industry and academic research. However, they provide more insights about the statistical properties and main problems in the implementation of CPPI, such as the multiplier value and the rebalancing frequency. They describe extensions to the standard model by identifying the impact of trading frequencies, of leverage effects, volatility effects, kurtosis effects, of jump size and jump frequency, management fees and transaction cost. They also propose an alternative price processes. They use the Student t-distribution, instead of the standard Normal distribution to model returns, since Normal distribution cannot “exhibit a high enough kurtosis to realistically represent stock returns”, which face the gap risk in a discrete

\(^{1}\) More details are shown in Table 1.1
rebalancing stages. In conclusion, they show that “in CPPI under discrete trading with typical market assumptions, a good multiplier value lies between 3 and 5 and a suitable rebalancing frequency is monthly”, but in case the volatility is greater than 10%, CPPI with a multiplier greater than 1 has worse performance than CPPI 1\(^1\) (Khuman, Maringer, & Constantinou, 2008).

Annaert et al. (2009) evaluate the performance of CPPI, OBPI, BH strategies and the lightly researched SLPI. Based on a more realistic alternative to the classical assumption that risky asset returns are normally distributed, they use empirical distributions to simulate returns, a method called block-bootstrap simulation. They introduce the stochastic dominance framework to evaluate portfolio insurance strategies. Their results suggest that BH has higher returns than other portfolio insurance strategies, albeit at the cost of losing better downside protection. Regarding stochastic dominance, no dominance is identified among all considered strategies. They also find that 100% floor value should deliver the best downside protection. The paper also reveals that daily rebalancing for OBPI and CPPI dominates the less regular ones, even though we take into consideration the higher transaction costs caused by more frequency of rebalancing (Annaert, Van Osselaer, & Verstraete, 2009).

Within a currency portfolio context, Ho et al. (2011) analyze a complete comparison between the two standard portfolio insurance strategies, OBPI and CPPI, and three modern risk-based portfolio insurance strategies, such as Normal VaR- based (NVaR), Historical VaR-based (HVaR) and Historical ES-based (HES) strategies. With the aim of preserving portfolio value, the hedging merits of the five strategies are assessed and ranked in terms of four perspectives, namely Sharpe Ratio, return distribution, cumulative portfolio returns across years, and average turnover. They also compare the average hedging performance of the five strategies from the same four perspectives in the bull and bear market years. Finally, they conclude explicitly that HES is ranked at the top of the overall ranking in the whole period from 2001 to 2008, and in bull years (2002-2004, 2006-2007), as well as bear years (2001, 2005, 2008) (Ho, Cadle, & Theobald, 2011).

(Costa & Gaspar, 2014) try to compare popular strategies, namely OBPI, CPPI with the so-called naïve strategies, i.e. SLPI and BH\(^2\), which could be more easily set up by investors. With the standard assumption for stock prices, they use Monte Carlo simulations to compare these strategies with the first four moments of the rates of return, performance rations, such as

---

\(^1\) In their paper, they refer CPPI 1 to gapless portfolio or BH in our work.

\(^2\) In their paper, BH has another name as CPPI1. Indeed, BH is a generalized form of the CPPI where the multiplier is 1. We will discuss more about this in the next section.
Sharpe Ratio, Sortino Ratio, Omega Ratio, Upside Potential Ratio, and the stochastic dominance mechanism. They define eight market scenarios, bear (returns = -5% and -15%), normal (5%) and bull (15%), each with two different volatilities (15% and 40%). They also consider two floor values, i.e. 80% and 100%. In conclusion, the prominent result that they find is that the naïve strategies seem to be the best in comparison with other more complicated strategies (Costa & Gaspar, 2014).

Our thesis intends to complement and expand these research works by conducting a more extensive comparison between the portfolio insurance strategies, which are used in these earlier comparative studies, i.e. BH, CPPI and its variation of the multiplier, OBPI, VaRPI, and SLPI. Another contribution is that we conduct the empirical analysis using data from a frontier market – the Vietnamese stock market. Furthermore, like most other studies, we also use Monte Carlo simulations to compare the merits of the above-mentioned strategies with the classical assumptions of geometric Brownian motion for stock prices in predetermined market scenarios: bear (returns = -15%), normal (8%) and bull (15%), each paired with two levels of volatility of rates of return: 20%, and 40%. In terms of performance measures, we use the first four moments, Sharpe, Sortino, Omega and Floor Protection ratios, to give an explicit and practical ranking in all the determined market scenarios and the real market situation of Vietnam. Indeed, these measures are consistent with every investor’s objective of preserving portfolio value when they use portfolio insurance. We only consider the floor values of 100%, meaning that the investor requires to obtain at least 100% his initial investment at the end of the insurance period. Like (Cesari & Cremonini, 2003), we choose five years for the insurance period.

In Table 1.1, we provide a comparative review of the selected papers, based on their used-for-comparison strategies, their comparative methods, their parameters for market scenarios, and their used performance measures. The highlighted columns indicate the strategies used in their comparative studies. Due to the space limit in the table, some elements are abbreviated, which can be decoded by the footnotes. 

19
<table>
<thead>
<tr>
<th>Paper + method of simulation + prominent results</th>
<th>Parameters for market scenarios</th>
<th>Performance measures</th>
<th>OBPI</th>
<th>CPPI</th>
<th>SLPI</th>
<th>BH</th>
<th>VaRPI</th>
<th>HESPI/ NESPI</th>
<th>CM/ RF/NH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Zhu &amp; Kavee, 1988)</td>
<td>$\mu = 15%$, $\sigma = 10%-30%$, $r = 7%$, $T = 1$</td>
<td>$\mu$, $\sigma$, Max, Min, 75% Qtl, 25% Qtl</td>
<td></td>
<td></td>
<td>m=1.5, 2.5, 3.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Method</strong>: MC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Result</strong>: when $\sigma$ is high, <strong>none</strong> can guarantee FL properly.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 (Benninga, 1990)</td>
<td>$\mu = 10%$, $\sigma = 20%$, $r = 7%$, $FL = 55%-105%$ (step = 5), $T = 250$, $\Delta t = 1$, $TC = 0.5%$</td>
<td>ATW, Sharpe, TC, skew</td>
<td></td>
<td></td>
<td>m=2, 3, 4, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Method</strong>: MC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Result</strong>: SLPI dominates OBPI and CPPI with and without TC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 (Cesari &amp; Cremonini, 2003)</td>
<td>$\mu = (-30%$, -5%), (-5%, 5%), (5%, 30%), $\sigma = (10%$, 15%), (15%, 25%), (25%, 30%), $T = 5$, $\Delta t = 1/52$</td>
<td>$\mu$, $\sigma$, Sharpe, Sortino, RaR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CM</td>
</tr>
<tr>
<td><strong>Method</strong>: MC and BT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Result</strong>: BT shows <strong>no</strong> dominant strategies over all markets. <strong>MC</strong> shows that CP, CPL dominant in bear and no-trend markets, CM in bull markets and CP, CPL, BCDT in all markets.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 (Bertrand &amp; Prigent, 2005)</td>
<td>$S_o=100$, $\mu = 10%$, $r = 5%$, $\sigma = 20%$, $T = 1$</td>
<td>$\mu$, $\sigma$, skew, kurt, SD, Qtl</td>
<td></td>
<td></td>
<td>m = 4, 6, 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Method</strong>: MC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Result</strong>: OBPI dominates CPPI in normal market, or when there exists a sudden drop in the risky asset value when it is high, CPPI dominates OBPI in bull market and no drop. OBPI is equivalent to the CPPI with a stochastic multiple.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 (Khuman, Maringer, &amp; Constantinou, 2008)</td>
<td>$\mu = 10%$, $\sigma = 20%$, $r = 5%$, $T = 1$, $\Delta t = 1/4$, 1/12, 1/52, 1/252</td>
<td>ATW, PrL, ExL</td>
<td></td>
<td></td>
<td>m &gt; 1</td>
<td></td>
<td></td>
<td></td>
<td>RF</td>
</tr>
<tr>
<td><strong>Method</strong>: MC and BT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Result</strong>: With typical market assumptions, CPPI is good with $m = [3, 5]$ and $\Delta t = 1$. When the volatility is greater than 10%, CPPI with $m &gt; 1$ is dominated by CPPI 1.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 (Annaert, Van Osselaer, &amp; Verstraete, 2009)</td>
<td>$\sigma = 9.19%$, 13.11%, 20.09%, $FL = 80%$, 100%$, $T = 30$, $\Delta t = 1/12$, 1/52, 1/252, $TC = 0.5%$</td>
<td>$\mu$, $\sigma$, skew, Sharpe, VaR 5%, ES 5%, Omega, SD 1</td>
<td></td>
<td></td>
<td>FL 100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Method</strong>: BS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Result</strong>: BH has higher $+\mu$ than other strategies, but losing better downside protection. On SD, none is dominant. FL= 100% delivers the best downside protection. <strong>OBPI and CPPI with $\Delta t = 1/252$ dominates other $\Delta t$, regardless of TC.</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Result</td>
<td>Parameters for market scenarios</td>
<td>Performance measures</td>
<td>Other strategies</td>
<td>Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>---------------------------------</td>
<td>----------------------</td>
<td>-----------------</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Ho, Cadle, &amp; Theobald, 2011)</td>
<td><strong>Method:</strong> MC and BT</td>
<td><strong>Result:</strong> HES ranks the best strategy in all markets.</td>
<td><strong>μ</strong> = -15%, -5%, 5%, 15%, <strong>σ</strong> = 15%, 40%, <strong>FL</strong> = 80%, 100%, <strong>T</strong> = 5, <strong>Δt</strong> = 1/252</td>
<td><strong>μ</strong>, <strong>σ</strong>, skew, kurt, Sharpe, Sortino, Omega, UP, SD (1, 2, 3, 4), PrF, PrR</td>
<td><strong>NH</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Costa &amp; Gaspar, 2014)</td>
<td><strong>Method:</strong> MC and BT</td>
<td><strong>Result:</strong> SLPI, BH outperform OBPI, CPPI3, CPPI5.</td>
<td><strong>μ</strong> = -15%, 8%, 15%, <strong>σ</strong> = 20%, 40%, <strong>FL</strong> = 100%, <strong>T</strong> = 5, <strong>Δt</strong> = 1/12</td>
<td><strong>μ</strong>, <strong>σ</strong>, skew, kurt, Sharpe, Sortino, Omega, Prot.</td>
<td>m = 3, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Our work</strong></td>
<td><strong>Method:</strong> MC and BT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Result:</strong> BH is the dominant strategy in bear market, while OBPI may be preferred in the normal market and bull market with high volatility. CPPI5 dominates other strategies in the bull market with low volatility. SLPI performs badly in most of the market scenarios. See Table A, 9 for detailed ranking.</td>
<td></td>
<td></td>
<td>m = 3, 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1.1. A review of the main literature on portfolio insurance

1 Parameters for market scenarios: **μ** stands for mean returns, with which (in 6th row) “+” represents average excess return, and “-” average negative excess return, **σ** for volatility (standard deviation), **TC** for transaction cost, **T** for insurance period (year), **Δt** for frequency of rebalancing, **FL** for floor value, **K** for strike price of option, **S_o** for initial value of risky asset, **r** for risk-free rate.

2 Performance measures: skew stands for skewness, kurt for kurtosis, **TC** for transaction cost, **T** for insurance period (year), **Max** for maximum value, **Min** for minimum value, **Qtl** for quantile, Sharpe for Sharpe ratio, Sortino for Sortino ratio, Omega for Omega ratio, RaR for return at risk, **SD** for stochastic dominance criterion (1, 2, 3, 4 represents the order), **UP** for upside potential ratio, **ATW** for average terminal wealth, **PrL** for Probability of losses, **ExL** for Expected value of losses, **ShRD** for the shift in the portfolio returns distribution, **CuR** for the cumulated portfolio return, **Turn** for the turnover within **T**, **PrF** for probability of a portfolio value close to the floor, **PrR** for probability of a portfolio value above the risk-free investment, **Prot.** for the Floor Protection ratio.

3 OBPI variations: In 3rd row, **BCDT** stands for the method proposed by (Bird, Cunningham, Dennis, & Tippett, 1990), **NL** for Non-linear, and **PS** for the method of (Perold & Sharpe, 1988).

4 CPPI variations: **m** stands for multiplier of CPPI, **CPL** for lock-in-profit CPPI.

5 SLPI variations: **MA** stands for 30-week moving average, **MA2** for 10-week moving average.

6 ESPI variations: **HESPI** stands for Historical ESPI (appears only in 7th row), **NESPI** stands for Normal ESPI (appears only in 9th row).

7 Other strategies: **CM** stands for Constant Mix (appears only in 3rd row), which is a type of dynamic asset allocation strategies, but not a portfolio insurance strategy. **RF** stands for Risk-free asset (appears only in 5th row). **NH** stands for No-hedge strategy (appears only in 7th row).

8 Methods: **MC** stands for Monte Carlo simulations, **BT** for backtesting, **BS** for block bootstrap simulations.
1.3. Classification of Portfolio insurance strategies

As can be seen in the previous section, there has been an increasing number of portfolio insurance strategies over time. However, it is worth pointing out again that the class of portfolio insurance strategies distinguishes from others classes of strategies by its convex payoff function¹ (Cesari & Cremonini, 2003). In this section, we try to provide the basic mechanism behind the “base case” Buy and hold (BH) and the most widely-adopted portfolio insurance strategies in literature, namely the Constant proportion portfolio insurance (CPPI), the Option-based portfolio insurance (OBPI), the Risk-based portfolio insurance (RBPI), which includes VaR and ES approaches (VaRPI and ESPI respectively), and the Stop-loss portfolio insurance (SLPI). Based on this, we write our Python codes to carry out the Monte Carlo simulation method and backtesting.

1.3.1. Buy and hold strategy (BH)

The Buy and hold strategy (BH) is the simplest one, by which what the investor can do is doing nothing after buying his optimal portfolio of risky and riskless assets. Usually, the risk-averse investor applies the Markowitz portfolio theory (MPT) to construct for himself a portfolio with optimal weights assigned to all the risky assets and the riskless asset (e.g. bonds, cash, T-bills), which can either maximize expected return for a given level of risk, or minimize the risk level for a given desired level of expected returns. Then he just holds this portfolio for the whole insurance period, regardless of the fluctuations on the market. This portfolio includes riskless assets and sufficiently many risky assets, based on the concept of diversification. In the classical MPT framework, the portfolio expected return is the weighted average of all the constituent assets' expected returns², while the portfolio returns’ volatility (standard deviation) is defined as the square root of a function of the correlation coefficients of all pairs of the component assets’ expected returns.³ Indeed, diversification can eliminate (or at least reduce) the specific volatility of each asset’s expected return thanks to the negative correlations. Roughly speaking, when a risky asset’s rate of return decreases, and another risky asset in the

¹ A convex function is defined as a function such that, the set of points on or above its graph is a convex set. And a convex set can be simply understood as a region, where you cannot draw any line segment connecting two points within or on the boundary of the region, which can breach the boundary.
² \( \mathbb{E}(R_p) = \sum_i w_i \mathbb{E}(R_i) \), where \( \mathbb{E} \) is the expected value operator; \( R_p, R_i \) are returns of the portfolio and of asset \( i \) respectively, \( w_i \) is the weight of the component asset \( i \).
³ \( \sigma_p = \sqrt{\sum_i \sum_j w_i w_j \sigma_i \sigma_j \rho_{ij}} \), where \( \sigma_p, \sigma_i, \sigma_j \) are the sample standard deviations of the expected returns of the portfolio and assets \( i, j \) respectively; \( \rho_{ij} \) is the correlation coefficient between the expected returns of assets \( i \) and \( j \) and \( \rho_{ij} = 1 \), for \( i = j \) and \(-1 \leq \rho_{ij} \leq 1 \).
same portfolio increases in its returns (it means they have a negative correlation in returns), hence the gain from one asset can compensate the loss from the other. In practice, it is the virtually impossible case when all the two different assets have perfect positive correlation ($\rho_{ij} = 1$), meaning that all the pairs of two assets fall down together at the same rate since the investor does not only diversify into the assets but also into the industries that these assets belong to.

The investor with BH holds a well-diversified portfolio, and does not suffer from the already-eliminated specific risks of the constituent assets but still has to encounter the market risk. However, instead of passively holding the portfolio until the end of the investment period like BH, he may need an active insurance mechanism against the downside market risk by adjusting in some way the allocation between the risky and the risk-free assets to adapt the perpetual change of the market. It is why other portfolio insurance strategies come into play, and BH is a good benchmark, to see if it is efficient to use one of many portfolio insurance strategies, and if the answer is yes, which strategy is the best one and how best it is for each market scenario.

1.3.2. Constant Proportion portfolio insurance strategy (CPPI)

In his famous autobiography, Emanuel Derman, a very seasoned quantitative analyst, writes a lot about Fischer Black, whom he considers as “the most remarkable person I met at Goldman (Sachs).” (Derman, 2004). He once writes “His (Fischer Black’s) most noticeable quality was his stubborn and meticulous devotion to clarity and simplicity. … He wanted accuracy and honesty without the technical details” (Derman, 2004). Indeed, apart from his classic Black–Scholes model, he is also one of the inventors of the “strategy that is easy to understand, straightforward to implement, and flexible under changing conditions.” - Constant Proportion portfolio insurance strategy (CPPI) (Black & Jones, 1987). This strategy is easily implemented for the individual investors, the brokers, the investment advisors and helps their coordination smoother. Furthermore, according to (Pain & Rand, 2008), this strategy has been much more prevalent than the elder Option-based portfolio insurance strategy (OBPI) over time and “has been an important element in the expansion of structured products (which is a more general group of portfolio insurance products)” (see also (Allianz SE, 2013), (Professional Wealth Management, 2015), and (Wealth Focus Pty Ltd., 2010)).

This type of portfolio insurance strategy contains two parts, called an active asset and a reserve asset. “The reserve asset has an acceptable minimum return, while the active asset has a higher expected return than the reserve asset.” (Black & Jones, 1987). Moreover, the investor
has to adjust the ratio between them so that the value of the insured portfolio does not decrease below the predetermined protection level at the protection horizon. In our work, the active asset is the underlying portfolio of risky assets, and the reserve asset is the risk-free asset (e.g. T-bills, bond, cash). The exposure $E_t$, the amount of money that the investor invests in the underlying portfolio of risky assets at date $t$, is computed as follows:

$$E_t = m^{CPPI} C_t = m^{CPPI} (V_t^{CPPI} - FL_t)$$

(1.1)

where $C_t$ is the cushion, which equals to the difference between $V_t^{CPPI}$, the value of the portfolio insured by CPPI, and $FL_t$, the protected floor value. The subscript $t$ indicates “at date $t$”, while $m^{CPPI}$ is a constant multiplier of the CPPI.

Depending on the multiplier $m^{CPPI}$, we have many ways to identify the exposure $E_t$ to risky assets. A simple case is when $m^{CPPI}$ equals 1, which is equivalent to the BH. In this case, the initial floor $FL_0$ equals to the initial value of the risk-free asset purchased $B_0$ and it grows over time at the risk-free rate $r$, the initial cushion $C_0$ - the remainder of the investor’s money after purchasing the risk-free asset, is invested in the underlying portfolio of risky assets and equals to $E_0$. Subsequently, when time goes by, the $FL_t$ grows at $r$, while $C_t$ (or $E_t$ since $m = 1$) and $V_t^{CPPI}$ evolve according to the market.

However, if the investor is more aggressive, he can increase his exposure $E_t$ by increasing the multiplier $m^{CPPI}$ ($(m > 1)$. The higher the $m^{CPPI}$, the higher the $E_t$, the more the investor obtains profits in a bull market, and of course, the more he suffers from the loss in the bear market. It is worth noticing that in the equation (1.1), the $m^{CPPI}$ is a constant and the $FL_t$ increases at the assumed-to-be-known risk-free rate $r$ throughout the protection horizon to reach $FL_T$ at maturity, whereas the unpredictable changing values due to the fluctuation of the markets are the $C_t$ and consequently the $E_t$. $C_t$ changes because of the change of $V_t^{CPPI}$ – value of the insured portfolio including both risky and riskless assets. In the market rise, $V_t^{CPPI}$ increases leading to the rise of $E_t$, which requires the investor to invest more money in the risky assets.

During a strong bull market period, $E_t$ may approach the predetermined maximum percentage of the insured portfolio to invest in risky assets (which is termed “limit” by Black and Jones). The limit is generally 100%, meaning that $E_t = V_t^{CPPI}$ and the value of the risk-free assets $B_t = 0$. During this time, the investor is not allowed to purchase any more risky
assets due to borrowing constraints\(^1\). Conversely, as the market falls, \(E_t\) decreases, the investor has to reduce his exposure to risky assets and buys more risk-free one. Furthermore, in a bad scenario, when \(V^\text{CPPPI}_t\) approaches \(FL_t\) or \(C_t\) approaches 0, the exposure to the risky assets \(E_t\) will be zero regardless of the value of \(m\). It prevents the \(V^\text{CPPPI}_t\) from “piercing through” the floor \(FL_t\). But sometimes, the floor is “pierced through” when there is a flash crash\(^2\), and the investor cannot trade in time to reduce the exposure. In that case, \(C_t \leq 0\), then \(E_t\) should be equal to 0, or the investor’s fund is fully invested in risk-free asset, he cannot go back to trade in the stock market. The investor is locked in the riskless asset till the end of the protection horizon, which prevents him from participation in the bull market (if any). The exposure to risky assets can be expressed in both extreme cases as follows (Do, 2002):

\[
E_t = \max\{\min(mC_t, V_t), 0\}
\]

Moreover, the cushion has a more general rule:

\[
C_t = \max(V_t - FL_t, 0)
\]

The two extreme cases show the drawback of path dependence of the CPPI. We can understand the term “path dependence” as follows: there are two paths of the price of the risky assets with the same ending, but the first path reached the floor during the investment period while the other has not, the final strategy value will be \(FL_T\) and \(V_T\), respectively, even sometimes \(V_T\) is much higher than \(FL_T\). It is also demonstrated by (Carvalho, Gaspar, & Sousa, 2016) that higher multiplier (\(m = 3, m = 5\) in their study) raise the degree of path dependence of the strategy.

The last but essential concept of CPPI that we cannot ignore is the tolerance \(\tau\), which helps reduce the trading costs thanks to fewer rebalancings. Tolerance is defined as the percentage move of the market that triggers a trade to rebalance. Tolerance is defined for both upward and downward moves. In practice, \(\tau\) can be different for up and down movements, and the smaller the tolerance, the more precisely the \(E_t\) is related to the \(C_t\), but the more rebalancings the investor should implement, implying higher transaction costs. To be more formal, a rebalancing will occur when:

\[
\frac{|E_{t+1} - E_t|}{E_t} \geq \tau
\]

\(^1\) This 100% limit relates to the strictly self-financing assumption for portfolio insurance, which is once proposed by (Bird, Cunningham, Dennis, & Tippett, 1990). In this case, when \(B_t = 0\), the investor does not have any risk-free assets to sell in exchange for buying more risky assets.

\(^2\) “A flash crash is a very rapid, deep, and volatile fall in security prices occurring within an extremely short time period.” (cited from Wikipedia)
Another advantage of the CPPI is its flexibility. Investors can have this strategy tailored to their risk appetite by modifying the multiplier $m_{CPPI}$, or the initial floor $FL_0$, or the tolerance of market moves, $\tau$ to proceed trades. The investor is even flexible to change them during the protection horizon to adapt with unpredictable circumstances. However, in our study, we apply the time rebalancing, the more popular discipline in practice. We rebalance our insured portfolio on a regular basis, for example, monthly as benchmark. We also use daily and weekly frequencies with the mere goal of comparison the performance of the strategies. Furthermore, the CPPI strategy defined in the beginning are fixed until the maturity.

1.3.3. Option-based portfolio insurance strategy (OBPI)

As mentioned before, according to Leland’s original idea, an investor can guarantee, at maturity, the value of a stock (the same logic for a well-diversified underlying portfolio of risky assets) that he holds not to be lower than the predetermined floor $FL_T$\(^1\) by purchasing a corresponding European put option on that stock (or market index) with the strike price $K$ equaling $FL_T$. This strategy is called protective put (Hull, 2015). For example, since the investor wants to guarantee his stock not to fall below his initial investment in that stock $S_0$, he purchases an at-the-money European put option (with strike price $K = S_0$) and pays $p$ (say, EUR) for that put. Figure 1.3 illustrates his protective put method for his stock and its at-the-money European put option. As can be seen from the figure, at maturity $T$, the value of that already-insured stock $V_T$ (the red line) is guaranteed not be less than the strike price $K$ net the price of the European put option $(K - p)$.

However, it is often not possible to buy the listed European options with every strike prices and maturities, which are suitable for specific needs of each investor. Furthermore, the large fund managers do not always find enough liquidity of the necessary put options to trade in the option market. Fortunately, in 1981, based on the option pricing formula discovered by (Black & Scholes, 1973) and (Merton, 1973), Rubinstein and Leland provide an alternative method to obtain any desired European put options by creating them synthetically (Rubinstein & Leland, 1981). This method is now known as Option-based portfolio insurance strategy (OBPI), which involves a dynamic process to readjust the mixture between the underlying portfolio and the risk-free asset continuously by switching some shares of the portfolio to risk-

\[^1\text{Unlike the CPPI, of which the floor grows at risk-free rate during the protection period, the floor of OBPI is constant } FL_T = K.\]
free assets, or vice versa. \(^1\) The Black-Scholes model gives a value of a European put option at date \(t\) as:\(^2\)

\[
p_t = Ke^{-r(T-t)}N(-d_2) - S_te^{-q(T-t)}N(-d_1)
\]

\[
d_1 = \frac{\ln\left(\frac{S_t}{K}\right) + (r - q + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}
\]

\[
d_2 = d_1 - \sigma\sqrt(T-t)
\]

where, \(p_t\) is the value of the European put at date \(t\) (expressed in years), \(K\) is the strike price (which is equivalent to the floor at maturity \(FL_T\) of CPPI), \(r\) is the risk-free rate, \(T-t\) is the time to maturity of the option (expressed in years), \(N(.)\) is the cumulative normal distribution function, \(S_t\) is the value of the underlying portfolio at date \(t\), \(q\) is the dividend yield on the underlying portfolio (continuously compounded), \(\sigma\) is the volatility of the underlying portfolio. The volatility of the well-diversified underlying portfolio \(\sigma\) is usually assumed to be the multiplication of its beta and the volatility of a market index.

\(^1\) Actually, there are other ways to apply OBPI (See (Perold & Sharpe, 1988), (Bird, Cunningham, Dennis, & Tippett, 1990), (Cesari & Cremonini, 2003)). However, in our study, we choose the original method by (Rubinstein & Leland, 1981), but also consider the strictly self-financing principle for portfolio insurance.

\(^2\) This formula is based on a more general formula with dividends \(q\) proposed by (Do, 2002) and the notations of (Hull, 2015).
From equation (1.5), we have:

\[- e^{-q(T-t)} N(-d_1) = e^{-q(T-t)} [N(d_1) - 1] = \Delta_t \tag{1.8}\]

where, \(\Delta_t\) is the delta of the desired European put option at date t that the investor wants to replicate \(^1\). As a result, we have:

\[p_t = Ke^{-r(T-t)} N(-d_2) + \Delta_t S_t \tag{1.9}\]

It means that to create the desired option synthetically, at date t, the investor has to sell a proportion of \(-\Delta_t\) the underlying portfolio (because \(\Delta_t\) in equation (1.9) has negative value) and the proceeds invested in the riskless assets with value of \(Ke^{-r(T-t)} N(-d_2)\).

Using the equation (1.9), to replicate dynamically the aforementioned static protective put strategy involving supplementing a European put option on a market index to the well-diversified underlying portfolio, or in other words, to proceed the OBPI, we have:

\[S_t + p_t = S_t + (Ke^{-r(T-t)} N(-d_2) + \Delta_t S_t) = Ke^{-r(T-t)} N(-d_2) + (1 + \Delta_t) S_t \tag{1.10}\]

From equation (1.10), we can see that the OBPI at any date t involves a long position in the risk-free assets with the value: \(B_t = Ke^{-r(T-t)} N(-d_2)\) and another long position in the underlying portfolio of risky assets: \(E_t = (1 + \Delta_t) S_t = N(d_1) S_t\). It is worth emphasizing again the \(\Delta_t\) of the European put option has negative value, the exposure to the risky assets \(E_t\) of the OBPI method is less than the yet-to-be-insured underlying portfolio a proportion equaling to \(-\Delta_t\). It is because to create the synthetic put, the investor has to sell a proportion of \(-\Delta_t\) the underlying portfolio.

However, as we know, the \(\Delta_t\) of the option changes over time, so to know the exact proportion of \(S_t\) after a change in the value of the underlying portfolio, say, from date t to date \(t+1\) (from \(S_t\) to \(S_{t+1}\)) to switch between the risk-free assets and underlying portfolio so that the investor remains insured by OBPI, he needs to compute \(\Delta_t\) and \(\Delta_{t+1}\) and then their difference.

According to the equations (1.6), and (1.8), when \(S_t\) increases, \(d_1\) increases too, leading to the increase of \(N(d_1)\) and \(\Delta_t\). Subsequently, \(-\Delta_t\) decreases, or the proportion of the stocks in the underlying portfolio sold decreases. It means that the investor has to repurchase some of the underlying portfolio. Conversely, when \(S_t\) goes down, \(-\Delta_t\) goes up, the investor has to sell more than the prior proportion of the underlying portfolio. This is exactly the “buy high, sell low” rule for the underlying portfolio that was mentioned in the previous section.

\(^1\Delta_t\) is defined as “as the rate of change of the option price with respect to the price of the underlying asset.” (Hull, 2015). \(\Delta_t\) take on values between -1 and 0.
We can describe the steps of OBPI in more detailed as follows. At date 0, to insure an underlying portfolio by OBPI, \((-\Delta_0)\) of that portfolio should be sold initially to invest in the risk-free assets. As a result, the portfolio insurance now includes two components: the needed-to-be-insured underlying portfolio and the risk-free asset with the weights: \(1 - (-\Delta_0)\) and \((-\Delta_0)\) respectively at the beginning. At time \(t+1\), if the value of the portfolio changes, according to the OBPI rule, the investor has to sell \((-\Delta_{t+1}) - (-\Delta_t)\) of the underlying portfolio. When the value of the underlying portfolio \(S_t\) increases \((S_{t+1} > S_t)\), \((-\Delta_t)\) decreases \((-\Delta_{t+1}) < (-\Delta_t))\), the investor has to sell a negative \((-\Delta_{t+1}) - (-\Delta_t)\) of the underlying portfolio or to buy that portion of portfolio with the money gained from selling riskless assets. Contrariwise, when the portfolio’s value \(S_t\) falls \((S_{t+1} < S_t)\), the investor has to sell a positive portion of the portfolio that equals to \((-\Delta_{t+1}) - (-\Delta_t)\) and the proceeds invested in the risk-free asset. The cost of the OBPI comes from the fact that the investor buys the stocks in his well-diversified portfolio after the market has just increased and sells them after the market decline. The option can be created synthetically by trading the index futures, which reduce dramatically the cost of OBPI. However, it is out of the scope of this study, which concentrates on Vietnamese stock market, where there is not yet a derivative market.

The last but really interesting thing we want to mention is that in their paper, Bertrand and Prigent prove that CPPI with a constant \(m_{CPPI}\) is just a particular case of OBPI with a random multiplier \(m_{OBPI}(S_t)^1\) and given as a function of \(S_t\):

\[
m_{OBPI}(S_t) = \frac{S_t N(d_1)}{C(S_t, K)}
\]

(1.11)

where \(C(S_t, K)\) is the cushion, which is a function of \(S_t, K\) and equals to the price of European call option with the same strike and maturity at date \(t\), \(c_t\) (Bertrand & Prigent, 2005).

1.3.4. VaR-based portfolio insurance strategy (VaRPI)

A financial institution has to deal with many types of risk measures of thousands of market variables that it exposes, to name but a few, an interest rate, an equity index, or a commodity price. Therefore, thousands of risk measures are produced every day, albeit being useful, making senior risk managers difficult to oversee. They need a total risk measure of all these specific risk measures. Obviously, this single number is also useful for financial regulators, who administer thousands of financial institutions, to require each of them to keep an exact reserve capital for the risks it is bearing.

---

1 We consider a constant is a special case of a random variable.
2 For proof, see (Bertrand & Prigent, 2005).
Value at Risk (VaR) is an attempt to provide a total risk measure, which is invented by the J. P. Morgan bank in 1994. Based on (Hull, 2012), we can define this term by making a statement similar to the following: “We are $\alpha$ probability certain that we will lose at most VaR in time $T$”, where VaR by convention always has positive value, can be expressed in the unit of any currency or percentage. It can be paraphrased like “We have $(1 - \alpha)$ probability that the loss will exceed VaR in time $T$.”. From now on, we define such $(1 - \alpha)$ as the shortfall probability and the Value at Risk can be defined by (Olivieri & Pitacco, 2011) as the amount $VaR_\alpha$ such that:

$$\mathbb{P}[Z \leq VaR_\alpha] = 1 - \alpha$$  \hspace{1cm} (1.12)

Such VaR of the portfolio is a function of $T$ – the time horizon, and $\alpha$ - the confidence level. As a side note, since there is often not enough data for the historical simulation to estimate the VaR of more than one day, the following assumption is often used in practice (Hull, 2015):

$$N\text{ day VaR} = 1\text{ day VaR} \times \sqrt{N}$$  \hspace{1cm} (1.13)

However, the VaR metric has its drawbacks. Indeed, VaR answers the question: “How likely is it that I lose more than EUR 1 billion in 1 week?”, the answer can be, say, 1%. However, it cannot respond to the following question: “Ok. There is 1% chance that I can lose more than EUR 1 billion in 1 week. But how much more than EUR 1 billion can I lose? 5 billion, 10 billion?”. We can continue “I need a number, at least an average”. Expected shortfall (ES), CVaR (Conditional value at risk), Average Value-at-Risk (AVaR) or other names can provide us this number. Figure 1.4 and Figure 1.5 give a demonstrative comparison between VaR and ES. These two portfolios have the same VaR, but the portfolio in Figure 1.4 is less risky since the potential losses are much less.

ES can be considered as a weighted average of the mass (losses), which are left to the VaR and is defined more formally as follows:

$$ES_\alpha = \mathbb{E}[Z | Z < VaR_\alpha]$$  \hspace{1cm} (1.14)
VaR and ES have been recently applied for portfolio insurance, e.g. (Ho, Cadle, & Theobald, 2011), (Jiang, Ma, & An, 2009), and they are called VaR-based portfolio insurance (VaRPI) and ES-based portfolio insurance (ESPI) respectively. In general, a portfolio insurance approach always has a constraint, beyond which the rebalancing is instigated. This constraint can be either the floor value $FL_t$ in CPPI, the strike price $K$ in OBPI (or $FL_T$) or the downside risk measure (VaR or ES) in VaRPI or ESPI respectively. These risk-based portfolio insurance strategies (RBPI) involve controlling dynamically the constraints of predetermined $VaR_{\alpha,t}(R_p)$ or $ES_{\alpha,t}(R_p)$, which reflects the investor’s risk tolerance during the protection horizon, by adjusting the investment weights between the risky and riskless assets in the insured portfolio.

In their paper in 2009, Jiang et al. prove that the weight of the risk-free assets $W_t$ in the insured portfolio is a decreasing function of the market value of the underlying portfolio in a relatively volatile market (Jiang, Ma, & An, 2009). It is also intuitively clear that in the case of

---

1 Control can be to keep the downside risk measure be greater or equal to the target one (see (Jiang, Ma, & An, 2009), (Ho, Cadle, & Theobald, 2011)) or to let it varies in a tolerance range (see (Gupta & Kartinen, 2000)). In our analysis in the following chapter, we apply the latter. (Or maybe we choose the former).
a bear market, the shortfall probability is higher than required due to the exposure to risky assets, the investor has to reduce his exposure by shifting his funds from risky to risk-free assets and vice versa. Again, we can see that this action complies with the “buy high, sell low” rule of portfolio insurance. The calculation of the weight of the risk-free assets $W_t$ is given as follows (Jiang, Ma, & An, 2009):

$$W_t = \frac{\text{floor} - V_t e^{(\mu - 0.5 \times \sigma^2) \times (T - t)} - z_\alpha \times \sigma \times \sqrt{T - t}}{V_t e^{r \times (T - t)} - V_t e^{(\mu - 0.5 \times \sigma^2) \times (T - t)} - z_\alpha \times \sigma \times \sqrt{T - t}}$$

(1.15)

where, $z_\alpha$ is the $\alpha$ – percentile of the standard normal distribution, while other elements are defined similarly as before.

More interestingly, like the case of OBPI, CPPI with a constant $m^\text{CPPI}$ is proved to be just a special case of VaRPI with a dynamic and time-varying implied multiplier $m_t^\text{VaR}$. In their paper, Herold et al. provide the formula to compute the $m_t^\text{VaR}$ at date t, with $1 - \alpha$ shortfall probability for VaRPI:

$$m_t^\text{VaR} = \frac{1}{VaR_{\alpha,t}(R_{p,t})}$$

(1.16)

where, $VaR_{\alpha,t}$ is the Value at risk with $1 - \alpha$ shortfall probability, at date t, of the insured portfolio, and $R_p$ is the returns of the insured portfolio.

In 2011, Ho et al. extend the formula to compute the multiplier $m_t^\text{ES}$ for ESPI.

$$m_t^\text{ES} = \frac{1}{ES_{\alpha,t}(R_{p,t})} = \frac{1}{\mathbb{E}[R_{p,t} | R_{p,t} \leq VaR_{\alpha,t}(R_{p,t})]}$$

(1.17)

where, $ES_{\alpha,t}(R_p)$ is the Expected shortfall with $1 - \alpha$ shortfall probability, at date t, of the insured portfolio, and $\mathbb{E}[(.))|(.)]$ is the conditional expectation operator.

Since the ESPI is relatively new and is still not thoroughly studied by the academics and practitioners of the field, we leave this interesting and useful portfolio insurance strategy for further research.

### 1.3.5. Stop-loss portfolio insurance strategy (SLPI)

The stop-loss portfolio insurance strategy was first introduced in 1988 by Rubinstein, and Bird, Dennis, and Tippett. The investor using this strategy initially invests his entire funds into the risky assets. Like other aforementioned portfolio insurance strategies, the constraint

---

1 We consider a constant is a special case of a variable.

2 For proof, see (Herold, Maurer, & Purschaker, 2005). For Monte Carlo simulation method and backtesting, we will consider two different methods to compute $VaR(R_p)$, proposed by (Ho, Cadle, & Theobald, 2011).

3 For proof, see (Ho, Cadle, & Theobald, 2011)
triggering rebalancing at date $t$ is the floor value $FL_t$. It is determined as the present value of the minimum portfolio at a risk-free rate $r$ during the protection horizon. The insured portfolio remains entirely invested in risky assets until its value falls below the current $FL_t$. Then the insured portfolio is fully transferred to risk-free assets and evolves at $r$ to reach exactly the $FL_T$ prespecified by the investor at the end of the insurance period. To be more formal, the Value of the insured portfolio by SLPI strategy can be computed as follows (Carvalho, Gaspar, & Sousa, 2016):

$$V_t^{SLPI} = \frac{V_0^{SLPI}}{E_0^{SLPI}} E_t^{SLPI} 1_{\{t < t'\}} + FL_t 1_{\{t > t'\}}$$  \hspace{1cm} (1.18)

where, $V_0^{SLPI}$ and $E_0^{SLPI}$ are the initial values of the insured portfolio and the risky assets in that portfolio corespondingly, $t'$ is the first instant that the $V_t^{SLPI}$ “touches” $FL_t$, if it does exist ($t' > 0$). The indicator function $1_{\{t < t'\}}$ equals to 1 if $t < t'$ is true, or date $t$ is before that “if-exist-touching moment”, otherwise it equals to 0; the same logic applies to $1_{\{t > t'\}}$.

As can be seen from equation (1.18), like CPPI with high $m$, the SLPI also shows a high degree of path dependence. Therefore, the value of the insured portfolio by SLPI does not depend solely on the final value of the underlying portfolio, but also on the path taken by the value of the underlying portfolio over the insurance period. This path dependence helps SLPI perform well in case that the value of the underlying portfolio never “pierces through” the $FL_t$ or it breaches the current $FL_t$ at date $t$ but never recovers and closes below the prespecified at the end of the insurance period. However, it is much less desirable in case that after the equity portfolio’s value falls below the current $FL_t$, it bounces back and rises much above the target minimum $FL_T$. The investor already converts his entire money to riskless assets and have no chance to go back in the subsequent rising market.

One more time, like for all of the abovementioned strategies, there exists a relationship between SLPI and CPPI. In 1992, Black and Perold showed that SLPI is a special case of CPPI when $m$ goes to infinity (Black & Perold, 1992). Indeed, from equation (1.2), it is easy to check this relationship. When $m \rightarrow +\infty$ and $C_t > 0$, leading to $mC_t \rightarrow +\infty$, $\min(mC_t, V_t)$, therefore, always equals $V_t$ and as a result, $E_t = \max[\min(mC_t, V_t), 0] = V_t$, meaning that when $C_t > 0$, the whole portfolio is the risky assets. Conversely, when $C_t < 0$, $E_t = 0$ and the entire portfolio does not expose to the risky assets whatsoever.

### 1.3.6. A comprehensive comparison of the popular portfolio insurance strategies

In section 1.3, we have already introduced the basic mechanism of all the popular portfolio insurance strategies. Since some aspects are prominent in one strategy, but are not in
others, in the previous subsections, we just mention the important ones. However, for the sake
of a comprehensive and illustrative comparison of all the aforementioned portfolio insurance
strategies, we will describe all the mentioned strategies briefly in terms of all the aspects in
Table 1.2.

As you can see in for the first aspect, all the strategies differ from each other by the value
of multiplier m. For example, BH with a constant $m^{BH} = 1$ is the special case of $m^{CPP}$ which
is itself a constant and has value greater than 1\(^1\), whereas $m^{SLPI} = +\infty$. In contrast, $m^{OBPI}(S_t)$
is a function of the random variable $S_t$, hence it is also a random variable, while $m^{VaR}$ and $m^{ES}$
are time varying.

Prior to implementing any portfolio insurance strategy, the buyer of the insurance must
decide what he wants to protect against for his underlying portfolio. It can be the traditional
concept of floor $FL_t$, which the investor does not want his portfolio to fall below at date t.
However, due to the increasing regularity of gap risk because of the discrete portfolio
rebalancing, liquidity of the assets, the insurance issuers do not 100% guarantee the required
$FL_t$. To overcome this drawback, the academics come up with a newer concept of shortfall
probability 1- $\alpha$. In plain English, they accept the fact of “not 100% guarantee the required
$FL_t” , but they need the insurance issuers to be, say, 95% sure to guarantee the required $FL_t$.
We name the second aspect as constraint, since like any maximization problem, while even
implementing a portfolio insurance strategy, the investor still aims to maximize his returns
given that his constraint of guarantee, e.g. the $FL_t$ or the $\alpha$, is controlled.

Whenever this constraint is violated, the investor is forced to rebalance his insured
portfolio according to the rule of portfolio insurance strategy he chose at the beginning.
However, in practice, due to the transaction cost, may be caused by continuous rebalancing, he
can set his tolerance range that allows his constraint to be varied in his tolerable range to avoid
unnecessary trades. For the third aspect, we mention about this. The investor can set his
tolerance for all of the strategies, but for the BH, simply because the investor choosing this
strategy does nothing, so he does not define this aspect.

For the fourth aspect, we mention about the cushion, which is originally used to describe
the CPPI mechanism. Since the BH has $m = 1$, its cushion is exactly the exposure to risky
assets. Interestingly, in line with the idea to find the relationship of multiplier between CPPI

\(^1\) When $0 < m < 1$, (and $FL_0 = 0$), it is the multiplier of Constant mix strategy.
and OBPI, (Bertrand & Prigent, 2005) find the formula for the cushion of OBPI, although it is quite complex.

As mentioned above, a portfolio insurance or an insured portfolio includes the needed-to-be-insured underlying portfolio of risky assets and the risk-free assets. The three following aspects (5th, 6th, 7th) give the formula to compute their respective values at date t. They are the Value of the risky assets, the Value of the risk-free assets, and the total Value of the insured portfolio.

For the eighth aspect, we mention the path dependence aspect of a portfolio insurance strategy. The more the strategy is path dependent, the more chance it has to be locked in the risk-free assets and miss the opportunity to participate again in the succeeding bull market. The BH, OBPI are path independent while SLPI is highly path dependent and CPPI becomes more path dependent with a higher multiplier.

For the ninth aspects, we discuss the algorithm used for Python codes to calculate the value of the insured portfolio after each period. For the full codes of the strategies and others, see APPENDIX C.

With an endeavor to show an illustrative and condensed comparison, we will put all the strategies and all the most important aspects together in Table 1.3 and name the aspects as the titles of the rows. Before going into details, we caution that due to the space limit in the table, the aspects are abbreviated, and can be decoded by the footnotes. Furthermore, in case that the context is clear, we avoid the superscript that indicates the type of strategy. The reader can still follow easily since they are placed within the columns of the corresponding strategy. For instance, in column CPPI, we use $V_t$ instead of $V_t^{CPPI}$. 
### Buy and Hold

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1/ Multiplier: ( m = 1 )</td>
<td>9/ Algorithm: (Box A)</td>
</tr>
<tr>
<td>2/ Constraint: ( FL_t = B_t )</td>
<td>Similar to Box B with the Input ( m = 1 ).</td>
</tr>
<tr>
<td>3/ Tolerance range: Not available</td>
<td></td>
</tr>
<tr>
<td>4/ Cushion: ( C_t = V_t - B_t )</td>
<td></td>
</tr>
<tr>
<td>5/ Exposure to risky assets: ( E_t = C_t )</td>
<td></td>
</tr>
<tr>
<td>6/ Risk-free asset: ( B_t = B_T \times e^{-r(T-t)} )</td>
<td></td>
</tr>
<tr>
<td>7/ Value of the insured portfolio: ( V_t = E_t + B_t )</td>
<td></td>
</tr>
<tr>
<td>8/ Path dependence: None</td>
<td></td>
</tr>
</tbody>
</table>

### Constant Proportion portfolio insurance strategy

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1/ Multiplier: ( m &gt; 1 ), constant</td>
<td>9/ Algorithm: (Box B)</td>
</tr>
<tr>
<td>2/ Constraint: ( FL_t = FL_T \times e^{-r(T-t)} )</td>
<td></td>
</tr>
<tr>
<td>3/ Tolerance range: Available</td>
<td></td>
</tr>
<tr>
<td>4/ Cushion: ( C_t = \max (V_t - FL_t, 0) )</td>
<td></td>
</tr>
<tr>
<td>5/ Exposure to risky assets: ( E_t = \max (mC_t, V_t) )</td>
<td></td>
</tr>
<tr>
<td>6/ Risk-free asset: ( B_t )</td>
<td></td>
</tr>
<tr>
<td>7/ Value of the insured portfolio: ( V_t = E_t + B_t )</td>
<td></td>
</tr>
<tr>
<td>8/ Path dependence: higher ( m ), more path dependent</td>
<td></td>
</tr>
</tbody>
</table>

**Input:** \( FL_T, m, r, T, \) path of returns \([returns_0, ..., returns_{n-1}]\) (n elements); Create path of zeros for values of the insured portfolio \([V_0 = 0, ..., V_n = 0]\) (n + 1 0s); Set \( dt = \frac{T}{n} \); Set first value \( V_0 \); Set the first present value of floor: \( FL_0 = FL_T \times e^{-rT} \); Index \( t \) repeats \( n \) times from 0 to \((n-1)\):

- If \( V_t > FL_t \) then: \# when the insured portfolio does not pierce through the floor (see equation (1.2))
  - \( C_t = V_t - FL_t \);
  - \( E_t = C_t \times m \);
  - \( B_t = V_t - E_t \);
- If \( E_t > V_t \) then: \# in strong bull market, exceed the limit (see equation (1.2))
  - \( V_{t+1} = V_t \times e^{returns_t} \);
- Else: \# when vice versa, exposure to risky assets is less than the money we have (see equation (1.2))
  - \( V_{t+1} = E_t \times e^{returns_t} + B_t \times e^r \times dt \);
Else: # when the insured portfolio pierces through the floor (see equation (1.2))

\[ V_{t+1} = V_t \times e^{rt} \times dt, \]
\[ FL_t = FL_T \times e^{-r(T-t) \times dt}; \]

Output: new path of values of the insured portfolio \([V_0, ..., V_n]\) (n + 1 elements).

---

**Option-based portfolio insurance strategy**

<table>
<thead>
<tr>
<th>1/ Multiplier: ( m = \frac{S_t N(d_1)}{C(S_t, K)} )</th>
<th>9/ Algorithm: (Box C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{(Bertrand &amp; Prigent, 2005)} )</td>
<td><strong>Input:</strong> ( K, S, T, \sigma, r ), path of returns ([\text{returns}<em>0, ..., \text{returns}</em>{n-1}]) (n elements);</td>
</tr>
<tr>
<td>2/ Constraint: ( FL_t = FL_T = K )</td>
<td>Create path of zeros for values of the insured portfolio ([V_0 = 0, ..., V_n = 0]) (n + 1 0s);</td>
</tr>
<tr>
<td>3/ Tolerance range: Available</td>
<td>Set ( dt = \frac{T}{n} );</td>
</tr>
<tr>
<td>4/ Cushion: ( C_t = \text{value of European call option} ) ( (c_t) ) ( \text{(Bertrand &amp; Prigent, 2005)} )</td>
<td>Set first value ( V_0 );</td>
</tr>
<tr>
<td>5/ Exposure to risky assets:</td>
<td>Set the first present value of floor: ( FL_0 = FL_T \times e^{-rT} );</td>
</tr>
<tr>
<td>( E_t = S \times N(d_1) )</td>
<td>Index ( t ) repeats n times from 0 to (n-1):</td>
</tr>
<tr>
<td>See equation (1.10)</td>
<td>( d_1 = \frac{\ln(S_t/K) + (r + \frac{\sigma^2}{2})(T-t)}{\sigma \sqrt{T-t}} ); # ( q = 0 )</td>
</tr>
<tr>
<td>6/ Risk-free asset:</td>
<td>( d_2 = d_1 - \sigma \sqrt{T-t}; )</td>
</tr>
<tr>
<td>( B_t = Ke^{-r(T-t)} N(-d_2) )</td>
<td>( w\text{Risky}_t = \frac{S(t) \times N(d_1)}{Ke^{-r(T-t)} N(-d_2) + S \times N(d_1)} ); # Weight of risky assets = ( \frac{E_t}{B_t + E_t} )</td>
</tr>
<tr>
<td>See equation (1.10)</td>
<td>( w\text{Riskfree}_t = 1 - w\text{Risky}_t ); # Weight of riskfree assets</td>
</tr>
<tr>
<td>7/ Value of the insured portfolio:</td>
<td>( V_{t+1} = E_t \times e^{rt} \times \text{returns} + B_t \times e^{r \times dt}; )</td>
</tr>
<tr>
<td>( V_t = E_t + B_t )</td>
<td>( S_{t+1} = S_t \times e^{rt} \times \text{returns} );</td>
</tr>
<tr>
<td>8/ Path dependence: None</td>
<td><strong>Output:</strong> new path of values of the insured portfolio ([V_0, ..., V_n]) (n + 1 elements).</td>
</tr>
</tbody>
</table>
### VaR-based portfolio insurance strategy

1/ **Multiplier:** \( m = \frac{1}{\text{VaR}_{\alpha,t}(R_p)} \)  
   (Herold et al., 2005)

2/ **Constraint:** \( \text{VaR}_{\alpha,t}(R_p) \)

3/ **Tolerance range:** Available

4/ **Cushion:**  
   \( C_t = w_{\text{Risky}} t \times \text{VaR}_{\alpha,t}(R_p) \times V_t \)

5/ **Exposure to risky assets:**  
   \( E_t = w_{\text{Risky}} \times V_t \)

6/ **Risk-free asset:**  
   \( B_t = w_{\text{Riskfree}} \times V_t \)

7/ **Value of the insured portfolio:**  
   \( V_t = E_t + B_t \)

8/ **Path dependence:** To be determined.

9/ **Algorithm:** (Box D)

**Input:** \( \alpha \) (confidence level), floor, \( r \), \( T \), \( \mu \), \( \sigma \) path of returns \([\text{returns}_0, ..., \text{returns}_{n-1}]\) (n elements);  
Create path of zeros for values of the insured portfolio \([V_0 = 0, ..., V_n = 0]\) (n + 1 0s);

- Set \( dt = \frac{T}{n} \);
- Set first value \( V_0 \);
- Index \( t \) repeats \( n \) times from 0 to (n-1):
  
  \[
  w_{\text{Riskfree}} = \frac{\text{floor} - V_t \times e^{(\mu - 0.5 \times \sigma^2) \times (T-t) - z_\alpha \times \sigma \times \sqrt{T-t}}}{V_t \times e^{r \times (T-t)} - V_t \times e^{(\mu - 0.5 \times \sigma^2) \times (T-t) - z_\alpha \times \sigma \times \sqrt{T-t}}};
  \]

  # Weight of riskfree assets. See (Jiang et al., 2009) for the formula.

  \[
  w_{\text{Risky}} = 1 - w_{\text{Riskfree}}; \]

  # Weight of risky assets

  \[
  V_{t+1} = w_{\text{Risky}} \times V_t \times e^{\text{returns}_t} + w_{\text{Riskfree}} \times V_t \times e^{r \times dt};
  \]

  \[
  S_{t+1} = S_t \times e^{\text{returns}_t};
  \]

**Output:** new path of values of the insured portfolio \([V_0, ..., V_n]\) (n + 1 elements).

### ES-based portfolio insurance strategy

1/ **Multiplier:** \( m = \frac{1}{\text{ES}_{\alpha,t}(R_p)} \) (Herold et al., 2005)

2/ **Constraint:** \( \text{ES}_{\alpha,t}(R_p) \) (Ho et al., 2008)

3/ **Tolerance range:** Available

4/ **Cushion:**  
   \( C_t = w_{\text{Risky}} t \times \text{ES}_{\alpha,t}(R_p) \times V_t \)

5/ **Exposure to risky assets:**  
   \( E_t = w_{\text{Risky}} \times V_t \)

6/ **Risk-free asset:**  
   \( B_t = w_{\text{Riskfree}} \times V_t \)

7/ **Value of the insured portfolio:**  
   \( V_t = E_t + B_t \)

8/ **Path dependence:** To be determined.

9/ **Algorithm:** (Box E)

To be determined.
Stop loss portfolio insurance strategy

<table>
<thead>
<tr>
<th>1/ Multiplier: $m = +\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/ Constraint: $\text{FL}_t = \text{FL}_T \times e^{-r(T-t)}$</td>
</tr>
<tr>
<td>3/ Tolerance range: Available</td>
</tr>
<tr>
<td>4/ Cushion: $C_t = V_t - \text{FL}_t$</td>
</tr>
<tr>
<td>5/ Exposure to risky assets: $E_t = V_t 1_{(t &lt; t')}$</td>
</tr>
<tr>
<td>6/ Risk-free asset: $B_t = V_t 1_{(t &gt; t')}$</td>
</tr>
<tr>
<td>7/ Value of the insured portfolio: $V_t = \frac{V_0}{E_0} E_t 1_{(t &lt; t')} + \text{FL}<em>t 1</em>{(t &gt; t')}$</td>
</tr>
<tr>
<td>8/ Path dependence: high</td>
</tr>
</tbody>
</table>

9/ Algorithm: (Box F)

**Input:** $\text{FL}_T$, $r$, $T$, path of returns $[\text{returns}_0, ..., \text{returns}_{n-1}]$ (n elements);

Create path of zeros for values of the insured portfolio $[V_0 = 0, ..., V_n = 0]$ (n + 1 0s);

Set $dt = \frac{T}{n}$;

Set first value $V_0$;

Set the first present value of floor: $\text{FL}_0 = \text{FL}_T \times e^{-rT}$;

Index $t$ repeats $n$ times from 0 to (n-1):

If $V_t > \text{FL}_t$ then: #when the insured portfolio does not pierce through the floor

$V_{t+1} = V_t \times e^{\text{returns}_t}$;

Else: #when the insured portfolio pierces through the floor

$V_{t+1} = V_t \times e^{r \times dt}$;

$\text{FL}_t = \text{FL}_T \times e^{-r(T-t) \times dt}$;

**Output:** new path of values of the insured portfolio $[V_0, ..., V_n]$ (n + 1 elements).

Table 1.2. Summary of the considered aspects of the portfolio insurance strategies
<table>
<thead>
<tr>
<th>Aspects</th>
<th>BH</th>
<th>CPPI</th>
<th>OBPI</th>
<th>VaRPI</th>
<th>ESPI</th>
<th>SLPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( m = 1 )</td>
<td>( m &gt; 1, \text{constant} )</td>
<td>( S_t N(d_1) )</td>
<td>( \frac{1}{VaR_{\alpha,t}(R_p)} )</td>
<td>( \frac{1}{ES_{\alpha,t}(R_p)} )</td>
<td>(+\infty)</td>
</tr>
<tr>
<td>2 constr:</td>
<td>( FL_t = B_t )</td>
<td>( FL_t = FL_T e^{-r(T-t)} )</td>
<td>( FL_t = FL_T = K )</td>
<td>( VaR_{\alpha,t}(R_p) )</td>
<td>( ES_{\alpha,t}(R_p) )</td>
<td>( FL_t = FL_T e^{-r(T-t)} )</td>
</tr>
<tr>
<td>3 ( \tau ):</td>
<td>-</td>
<td>( \checkmark )</td>
<td>( \checkmark )</td>
<td>( \checkmark )</td>
<td>( \checkmark )</td>
<td></td>
</tr>
<tr>
<td>4 ( C_t )</td>
<td>( V_t - B_t )</td>
<td>( \max(V_t - FL_t, 0) )</td>
<td>( c_t )</td>
<td>( w_{\text{Risky}<em>t} \times \frac{VaR</em>{\alpha,t}(R_p) \times V_t}{V_t} )</td>
<td>( w_{\text{Risky}<em>t} \times ES</em>{\alpha,t}(R_p) \times V_t )</td>
<td>( V_t - FL_t )</td>
</tr>
<tr>
<td>5 ( E_t )</td>
<td>( C_t )</td>
<td>( \max[\min(mC_t, V_t), 0] )</td>
<td>( S_t \times N(d_1) )</td>
<td>( mC_t )</td>
<td>( mC_t )</td>
<td>( V_t1_{(t &lt; t')} )</td>
</tr>
<tr>
<td>6 ( B_t )</td>
<td>( B_T e^{-r(T-t)} )</td>
<td>( V_t - E_t )</td>
<td>( K e^{-r(T-t)} N(-d_2) )</td>
<td>( (1 - w_{\text{Risky}_t}) \times V_t )</td>
<td>( (1 - w_{\text{Risky}_t}) \times V_t )</td>
<td>( V_t1_{(t &gt; t')} )</td>
</tr>
<tr>
<td>7 ( V_t )</td>
<td>( E_t + B_t )</td>
<td>( \max \left[ (E_t + B_t)1_{(V_t &gt; E_t)} + V_t1_{(V_t &lt; E_t)} \right] )</td>
<td>( E_t + B_t )</td>
<td>( E_t + B_t )</td>
<td>( E_t + B_t )</td>
<td>( \frac{V_0}{E_0} E_t1_{(t &lt; t')} + FL_t1_{(t &gt; t')} )</td>
</tr>
<tr>
<td>8 PD(^2):</td>
<td>0</td>
<td>higher ( m ), more PD</td>
<td>0</td>
<td>TBD(^3)</td>
<td>TBD</td>
<td>high</td>
</tr>
<tr>
<td>9 Algo</td>
<td>Box A</td>
<td>Box B</td>
<td>Box C</td>
<td>Box D</td>
<td>Box E</td>
<td>Box F</td>
</tr>
</tbody>
</table>

Table 1.3. Condensed summary of the portfolio insurance strategies

\(^1\) \( m \) stands for multiplier, constr for constraint, \( \tau \) for tolerance, \( C_t \) for cushion, \( E_t \) for Value of the risky assets, \( B_t \) for Value of the risk-free assets, \( V_t \) for Value of the insured portfolio, PD for Path dependence, Algo for Algorithm. The subscript \( t \) stands for “at date \( t \)”, \( T \) for the maturity, \( c_t \) is the value of European Call option, \( w_{\text{Risky}_t} \) for the weight of risky assets in the insured portfolio.

\(^2\) 0 means the strategy is path independent (Carvalho, Gaspar, & Sousa, 2016).

\(^3\) TBD stands for to be determined, due to the fact that they are still lightly researched, - for not available, \( \checkmark \) for available.
CHAPTER 2: METHODOLOGY OF PERFORMANCE COMPARISON BETWEEN DIFFERENT PORTFOLIO INSURANCE STRATEGIES

2.1. The design of simulation methods and data issues

In this study, we use the standard Monte Carlo simulation method and backtesting to carry out an extensive comparison of merits between the portfolio insurance strategies which were discussed in the previous section. We will first describe these two popular simulation methods briefly in the following two subsections. Subsequently, the simulation set up and issues related to real data are provided in the two remaining subsections.

2.1.1. Introduction to Monte Carlo simulation method and backtesting

A Monte Carlo simulation method is a procedure to sample repeatedly tens of thousands or hundreds of thousands of times random outcomes of a stochastic process. It is named after a town of the Principality of Monaco, where the Casino de Monte-Carlo is situated since casino games are random and the essence of this simulation is randomness. The modern version of this simulation was first developed by Stanislaw Ulam when he worked on the Manhattan project about nuclear weapons, and then John von Neumann programmed the computer to implement its calculations in the late 1940s. Based on the law of large numbers, the method produces the sample average, which converges to the actual value with a sufficiently large sample; moreover, the central limit theorem provides some information about the accuracy of the results (Glasserman, 2003).

Monte Carlo simulation has been widely applied in many industries such as finance, business, meteorology, manufacturing, pharmaceuticals, astronomy, particle physics. It is not until 1964 that David B. Hertz first applied this technique to the finance field (Hertz, 1964). In finance, the Monte Carlo simulation method creates simulated data following certain probability distributions of various uncertain factors (inputs) that affect the value of any financial instrument, e.g. stocks, derivatives, bonds, or the portfolio composing any of these assets. Eventually, with the help of a quantitative model (transfer equation), it produces the probability distribution of the value of the financial instrument (output). The probability distribution of the output tells us not only which value the output may be reached, but also how probable it is to occur. In this way, the researcher conducting this Monte Carlo simulation has a comprehensive view of all the possible outcomes at his disposal to obtain an estimate of the value of the instrument, to analyze the risk impact or to do whatever available analysis he wants.
for better decision making. The whole procedure of a basic Monte Carlo simulation is shown in Figure 2.1.

![Diagram of steps in a basic Monte Carlo simulation](image)

**Figure 2.1.** A diagram of steps in a basic Monte Carlo simulation.

Depending on the transfer equation, we provide different required inputs to obtain the different output. For instance, if the transfer equation is the function to create a matrix of random returns following Geometric Brownian motion, the inputs are the expected rate of return of the risky asset per annum, the volatility of the risky asset price per annum, the investment horizon, the number of time steps, and the number of paths that we want to simulate. The output will be a matrix of returns with the number of rows is the number of time steps, and the number of columns is the number of paths.

Another example of the transfer equation (that we use in our study) can be the function to create the Monte Carlo simulated matrix of the values of the insured portfolio with SLPI through 60 insurance sub-periods with 10,000 paths. The inputs are the floor at maturity, the risk-free rate, and the matrix of Monte Carlo simulated returns that we mentioned in the previous example. The output we obtain is 10,000 simulated paths of the value of the SLPI.
portfolio from month 0 to month 60. For more details, see the codes with the comment “# Function for the Monte Carlo simulation of SLPI”, provided in APPENDIX C.

As a side note, we also simulate the matrices of the insured portfolios with other strategies. However, because the data are random, it is inefficient to compare the strategies with different underlying paths of returns. The Monte Carlo simulation provides a powerful tool, which is termed “seed” to ensure the reproducibility. As a result, with the same “seed”, or the same path of Monte Carlo simulated returns, we can compare how our various portfolio insurance strategies behave, which is illustrated in Figure 3.1 to Figure 3.6 in Chapter 3.

The abovementioned portfolio insurance strategies include two parts: the risk-free assets and the risky assets. The former is not random, so there is nothing to do with the Monte Carlo simulation, but we can use the simulation to generate a sufficiently large sample of the evolution paths for the risky assets price.

The price of a risky asset $S$ is widely assumed to follow the geometric Brownian motion, which can be approximated by the discrete-time version as follows:

$$\frac{\Delta S}{S} = \mu \Delta t + \sigma \epsilon \sqrt{\Delta t}$$

(2.1)

where $\frac{\Delta S}{S}$ is the percentage change of the price of the risky asset in the short period of time $[t, t + \Delta t]$, $\mu$ is the expected rate of return of the risky asset per unit of time, $\sigma$ is the volatility of risky asset price per unit of time, and $\epsilon$ is a random variable with standard normal distribution, which describes the stochastic property of the risky assets (Hull, 2015). Both $\mu$ and $\sigma$ are calculated from historical data of market price.

The left-hand side of equation (2.1) includes two terms, in which the first term $\mu \Delta t$ indicates the expected value of the return in short period of time $[t, t + \Delta t]$, while the second one $\sigma \epsilon \sqrt{\Delta t}$ represents the stochastic component of the return. Since $\epsilon \sim N(0, 1)$, equation (2.1) implies that $\frac{\Delta S}{S}$ is approximately normally distributed with mean $\mu \Delta t$ and variance $\sigma^2 \Delta t$ (Hull, 2015):

$$\frac{\Delta S}{S} \sim N(\mu \Delta t, \sigma^2 \Delta t)$$

(2.2)

However, in practice, $\ln S$ (reads natural logarithm of $S$ (to base e)) is preferred to $S$ thanks to its theoretical and algorithmic benefits. For more details, see (Quantivity, 2011). More interestingly, there is an approximation between these two values, when we calculate the
change in the value of \( \ln S \) during a short period of time \( \Delta t \), \( \ln \frac{S_{t+\Delta t}}{S_t} \). \(^1\) It is because when \( \Delta t \) is small, \( \frac{\Delta S}{S} \) is consequently relatively small.

\[
\ln \frac{S_{t+\Delta t}}{S_t} \equiv \frac{\Delta S}{S}, \text{ when } \frac{\Delta S}{S} \ll 1
\]  

(2.3)

Therefore, in this study, we will conduct Monte Carlo simulations with equation (2.4) to write the codes to produce the sample of the path of the log-returns of the risky assets, which is derived from equations (2.1) and (2.3):

\[
\ln \frac{S_{t+\Delta t}}{S_t} = \mu \Delta t + \sigma \varepsilon \sqrt{\Delta t}
\]

(2.4)

Very similar to the Monte Carlo simulation method, the other method that we apply to compare the performance of alternative portfolio insurance strategies, the backtesting is more or less implemented through the steps which are already sketched in Figure 2.1. However, the process is simpler, because it uses only one real path of the inputs, instead of the thousands of simulated paths in the Monte Carlo simulation. The backtesting method has another advantage, as it does not require any assumption regarding the return distribution, and depicts the exact historical performance of portfolio insurance strategies as if we had implemented those strategies in the past.

However, this method gives us only a single scenario. Therefore, it does not provide us with the probability distribution of the possible performance outcomes of the considered strategies to imply the accuracy of our results. Furthermore, the results we obtain from backtesting depend highly on the chosen period. Hence, we use both simulation methods to supplement one another.

### 2.1.2. The simulation design

a. Introduction to the Vietnamese Stock Market

There have been many researches on the comparison between alternative portfolio strategies in the case of developed markets, including (Annaert, Van Osselaer, & Verstraete, 2009), (Ho, Cadle, & Theobald, 2011), (Costa & Gaspar, 2014), to name but a few, and in the case of developing markets, such as (Er & Erdogan Aktan, 2009) for the Turkish market, (Jiang, Ma, & An, 2009) for the Chinese market\(^2\). To the best of our knowledge, there has not existed any works on the frontier markets about the portfolio insurance up to now, let alone the

---

\(^{1}\) As we will use later on, the formula \( \ln \frac{S_{t+\Delta t}}{S_t} \) is the natural logarithm daily return with \( \Delta t \) can be usually a day, a week or a month.

\(^{2}\) For more details about the MSCI market classification, see (MSCI Inc., 2017).
Vietnamese market. Therefore, apart from providing a comprehensive comparison of portfolio insurance strategies, we contribute to the literature of the field with an empirical analysis of Vietnamese stock market - a fast-growing frontier market which attracts more and more domestic and foreign investors, including large hedge funds and investment banks.

The Vietnamese stock market is still very young in comparison with other developed markets, but since its establishment, the stock market has played a major role in the development of the Vietnamese economy. In recent years, it has constantly been ranked in the top 5 of the fastest growing stock markets in Southeast Asia region (State Securities Commission of Vietnam, 2016) thanks to the optimistic macroeconomic prospects and political stability.

There are two separate listing exchanges in the Vietnamese stock market, namely Ho Chi Minh City Stock Exchange (HOSE), which was first put into operation in 2000; and the Hanoi Stock Exchange (HNX), which was established 5 years later in 2005. Up to June 2017, in the bigger exchange HOSE, there are 372 securities, composing of 329 listed stocks of companies which have minimum charter capital of VND¹ 120 billion, 40 bonds, 3 fund certificates (ETF and mutual funds) (Ho Chi Minh Stock Exchange, 2017). In HNX, there are 378 securities of companies which have minimum charter capital of VND 30 billion, 571 bonds and 1 fund certificate (Ha Noi Stock Exchange, 2017). Furthermore, at HNX, the Unlisted Public Company Market (UPCoM) was first launched in 2009 with only 10 not-yet-listed state-owned and public companies, however, by the end of June 2017, this number has increased to 546 companies. The total capitalization of the entire capital market (including stocks and bonds) is VND 1.77 trillion (USD 80.45 billion), constituting about 74% of GDP of Vietnam in 2016. In terms of liquidity, the average trading value per session increased to VND 6,800 billion (USD 310 million) (State Securities Commission of Vietnam, 2016) ².

Thanks to these fast but stable achievements in an extended period, Vietnam outperforms its fellow frontier markets and is expected to be upgraded to emerging market status by MSCI³ shortly. In general, to be reclassified from a frontier to an emerging market, the Vietnamese stock market does not have so many struggles to meet the quantitative requirements, such as the market capitalization, liquidity. However, to adapt the qualitative criteria, the Ministry of

---

¹ VND (aka Vietnam Dong) is the Vietnamese local currency, the exchange rate is around USD/VND 22,000.
² For more details about the history and current situation of Vietnamese stock market, see (Trieu & Shalamov, 2015)
³ MSCI (Morgan Stanley Capital International) is a US-based leading provider of global fixed-income, equity indexes.
Finance and the State Securities Commission of Vietnam are striving to improve the legal framework to expand the foreign ownership limits, the mobilization of foreign currency flows for foreign investors, information disclosure in English, etc. The shift to the Emerging market status does not only help build the image of the country financial market but also gives additional boost to the constant rise of the Vietnamese stock market by directly luring the foreign capitals, like the last three promoted-to-emerging markets, viz. the United Arab Emirates, Qatar, and Pakistan, have witnessed the jump in their benchmark indices just after MSCI status shift.

Furthermore, to improve the competitiveness and transparency of the Vietnamese stock markets, the Vietnamese Government plans to merge the two stock exchanges HOSE and HNX by the end of 2017, which will have their own purposes. The HOSE is to handle stocks of companies, which were listed on both exchanges, while the HNX manages the government bonds and the new products, derivatives, which are introduced at most by the end of 2017 (State Securities Commission of Vietnam, 2016).

The derivative market, with its two first products, namely stock index and government bond futures, will furnish the investors with more tools to invest and will make the stock market of this new Asian Tiger even more attractive. It is worth mentioning again shortly about the main topic of our study that put options are used to insure the portfolios traditionally, or the futures are sometime preferred to direct trading stocks in the underlying portfolio to create a put option synthetically since it generally reduces the transaction costs. However, it takes time for the market makers in Vietnam to introduce other derivative products tailored for various needs of investors, such as options with different maturities or futures on different stock indices. Therefore, in our study, we will focus on using the risky assets themselves for the portfolio insurance strategies and the application of derivative products is left for further research when the derivative market in Vietnam will be more mature.

b. Data

Like other academics, in our research, we use the market index as a proxy for the well-diversified underlying portfolio. The dataset consists of the daily closing prices of the VN30 Total Return Equal Weight Index (VN30EQ), which comprises 30 stocks with the highest market capitalization and liquidity listed on the Ho Chi Minh City Stock Exchange (HOSE).

1 The index is originally calculated in USD by Phoenix Global Wealth Management and S&P Global. Bloomberg platform also provides the index in VND (Vietnamese local currency) by multiplying the original index with the daily exchange rates. We use the latter in this study.
the Vietnam largest and more established stock market, such as VN Dairy Products JSC, Masan Group Corp, VinGroup JSC, FPT Corp, to name but a few. The term “Total Return” means that the dividend payments from companies included in the Index are reinvested in the Index to make our research simpler\(^1\). The term “Equal Weight” indicates that the Index is the equal weighted average of 30 stocks. This “equal weight” aspect also simplifies a lot the procedure to build the underlying portfolio for the investors, particularly the individual investors with a lack of expertise, since they only need to select their optimal assets according to their stock-picking criteria and then assign all of the companies to the same weights in the portfolio.

In fact, there is an older index which represents the 30 blue chips of the stock market, VN30 Index, a capitalization-weighted average of 30 stocks with the highest market capitalization in HOSE. However, as proved in a paper (Phoenix Global Wealth Management, 2016), the VN30EQ that we choose for our research always outperforms the VN30 Index in the period from 2012 to 2016. It may be thought that the short period of 5 years of the above result does not give a reliable picture of the relative efficiency of the “equal weight” approach and the more traditional “market-cap weight” one. Backtesting of other markets around the world shows that in most of these countries “equal weight” index have dominated their counterparts, so in the long run, this trend should be with high probability the same in Vietnam (Phoenix Global Wealth Management, 2016). Additionally, we decide to use VN30EQ, instead of VN Index, which is a capitalization-weighted average index including all stocks listed in HOSE, since buying the 30 biggest stocks are more practical for both individual and institutional investors.

We consider a sample period from 03\(^{rd}\) April 2012 to 28\(^{th}\) April 2017, which provides us the real path of the VN30EQ daily price movement in a 5-year investment period for our backtesting of the considered portfolio insurance strategies. Our historical data is divided into 60 non-overlapping one-month insurance sub-periods; the first insurance sub-period is from 03\(^{rd}\) April 2012 to 30\(^{th}\) April 2012, and the last insurance sub-period is from 3\(^{rd}\) April 2017 to 28\(^{th}\) April 2017. In our study, from the path of the prices of first trading days of the 61 months of VN30EQ, we calculate the time series of 60 natural logarithmic monthly returns in place of the usual percentage change formula of the monthly stock prices. We choose the 5 recent years not the whole period of the HOSE since the market is more mature and reflects more correctly the actual situation of the recent Vietnamese stock market. Furthermore, the maturity of the investment period of 5 years is commonly selected in the literature (see for example (Cesari &

\(^1\) It means that in the equation (1.5), \(q = 0\).
Cremonini, 2003) and (Costa & Gaspar, 2014)). All the data of prices are obtained from Bloomberg platform with code name VNITR Index and are calculated in VND. However, in our study, for a comparison of the performance of the portfolio insurance strategies, we also calculate the daily and weekly log returns of VN30EQ.

In terms of a proxy for the risk-free rate, we choose the Vietnamese Interbank Offered Rate (VNIBOR), which is the average interest rates of the 8 prime contributed banks operating in Viet Nam, at which they agree to lend to other banks in VND. In fact, this rate is analogous to the LIBOR (London Interbank Offered Rate), a commonly-used proxy for the risk-free rate. The VNIBOR for 9 periods are calculated every working day, by taking the average of all the participating prime banks but the 2 top and 2 bottom contributor rates at that day\(^1\). The 9 periods are 1 day, 1 week, 2 weeks, 1 month, 2 months, 3 months, 6 months, 9 months and 1 year.

To be more precise, in this study, we choose the 1-month VNIBOR as the primary risk-free rate since we decide to rebalance our portfolio every month (We will discuss more about this discipline in the Rebalancing discipline part). We also obtain the data from Bloomberg platform, with the code name VNCD1MO Index. Like other researchers of this field, for the simplicity sake, the risk-free rate applied is the average of the 1-month VNIBOR rates in the whole investment period from 03rd April 2012 to 28th April 2017, which equals 4.14% per annum. However, to serve the purpose of a comparison between different rebalancing frequencies, the averages of 1-day and 1-week VNIBOR rates throughout the investment period are also calculated, which equal 2.92% and 3.2% per annum respectively.

c. Market scenarios

The weakness of using only one path of the past index movements in the backtesting method can be compensated by the Monte Carlo simulation method, which provides us more market scenarios that can be happened in the future for better comparison and analysis of alternative portfolio insurance strategies. In our study, we define 6 different market scenarios, which are defined by a combination of 3 different expected annual rates of return of the risky assets, viz. \( \mu = -15\% \), \( \mu = 8\% \) and \( \mu = 15\% \), and two different annual volatilities of the risky assets, i.e. \( \sigma = 20\% \) and \( \sigma = 40\% \). All of these parameters are selected based on the actual conditions of Vietnamese stock market, which are somewhat different from those indicated in previous studies for developed and emerging markets. For each of the 6 market scenarios, with the corresponding chosen parameters, 10,000 paths of the value of the risky assets, each with

---

\(^1\) The list of 12 prime participating banks is ACB, BIDV, Eximbank, Maritime, Techcombank, Sacombank, Saigonbank, VCB, Vietinbank, MB Bank, VP Bank, NCB Bank.
1241 daily log-returns, are simulated. The 6 market scenarios are summarized in Table 2.1, and they represent bear, normal, and bull markets with high and low volatilities.

<table>
<thead>
<tr>
<th>Types of market</th>
<th>Annual volatility</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low volatility ($σ = 20%$)</td>
</tr>
<tr>
<td></td>
<td>High volatility ($σ = 40%$)</td>
</tr>
<tr>
<td>Bear market ($μ = -15%$)</td>
<td>Scenario 1: $μ = -15%$, $σ = 20%$</td>
</tr>
<tr>
<td></td>
<td>Scenario 2: $μ = -15%$, $σ = 40%$</td>
</tr>
<tr>
<td>Normal market ($μ = 8%$)</td>
<td>Scenario 3: $μ = 8%$, $σ = 20%$</td>
</tr>
<tr>
<td></td>
<td>Scenario 4: $μ = 8%$, $σ = 40%$</td>
</tr>
<tr>
<td>Bull market ($μ = 15%$)</td>
<td>Scenario 5: $μ = 15%$, $σ = 20%$</td>
</tr>
<tr>
<td></td>
<td>Scenario 6: $μ = 15%$, $σ = 40%$</td>
</tr>
</tbody>
</table>

Table 2.1. Summary of the market scenarios

d. Floor value

One of the most important parameters for portfolio insurance strategies is the floor value. In this research, we set the floor value at maturity $F_{LT}$ as 100\% the value of the initial insured portfolio $V_0$, which is common in the literature (see (Bertrand & Prigent, 2005), (Annaert, Van Osselaer, & Verstraete, 2009). It means that at the end of the insurance period, the investor requires taking back at least 100\% of the money that he invested at the beginning of the period. In our study, to make all the mentioned portfolio insurance strategies comparable, the floor value at maturity for all strategies is VND 100,000,000 for the initial value of the insured portfolio set to VND 100,000,000.

e. Trading costs

Regarding the expense of the portfolio insurance strategies, it can be divided into two types: explicit cost (aka trading costs) and implicit cost (aka opportunity costs). Although the trading costs affect the performance of the portfolio insurance strategies, they are left for further research due to two reasons. The first reason is its complexity, the overall trading costs consist of some components such as the transaction costs, bid/ask spread, etc. Therefore, to investigate properly the impact of the trading cost on our mentioned portfolio insurance strategies, we need a more focused study. In fact, although the researchers in this field do not often take into consideration the trading costs in their comparison, there have been some papers addressing the impact of only transaction costs but only on one or two portfolio insurance strategies, for example CPPI ((Khuman, Maringer, & Constantinou, 2008), (Mkaouar & Prigent, 2010)), OBPI ((Clarke & Arnott, 1987), (Leland H. E., 1985)), CPPI and OBPI

---

1 For a brief description about this topic, readers may refer to (Northfield, 2015).
(Benninga, 1990), CPPI and VaRPI (Jiang, Ma, & An, 2009). The second reason is that our main aim is to compare among the strategies the implicit costs arising from the difference of the return distributions of the strategies.

f. Rebalancing discipline

In the presence of transaction costs, the portfolio insurance strategies become expensive in case of very frequent rebalancing. As a result, we encounter the trade-off between the accuracy of the guarantee of the strategy that the investors require to protect their portfolio, and the transaction costs that they have to pay. In other words, theoretically, if the investors pay less for the portfolio insurance strategy, they have to accept a less accurate strategy compared with the “theoretical” one. There are various rebalancing disciplines, namely calendar time discipline, market move discipline, lag discipline, etc. (Etzioni, 1986), (Arnott & Lovell, 1992). Actually, the topic of rebalancing disciplines is also really well-documented among academics and practitioners. Nevertheless, like the theme of portfolio insurance strategies, there is an ongoing debate on the dominance of any of the abovementioned disciplines. Because it is beyond the scope of this study, we focus on one of them, which is the calendar time discipline. The benchmark time frequency of rebalancing in our research is one month. It means that at the first working day of every month from 03rd April 2012 to 28th April 2017, the insured portfolio will be rebalanced to meet the target allocation. However, we also take into consideration the daily and weekly rebalancing frequencies.

2.2. Performance measures of Portfolio Insurance Strategies

In the previous subsection, we have already introduced all the necessary parameters to run our Monte Carlo simulation method and backtesting on the alternative portfolio insurance strategies that are discussed in the first chapter. However, to achieve our goal of comparing the performance of these strategies, the last but not least thing we want to mention is the more popular performance measures of portfolio insurance strategies that we choose from the literature of the field.

2.2.1. First four moments of the returns distribution

Although the focus of Portfolio insurance strategies is on the potential gains and the value protection, the first four moments of the returns distribution of Portfolio insurance strategies are still often used as basic but not sufficient measures for ranking the strategies (Bertrand & Prigent, 2005), (Annaert, Van Osselaer, & Verstraete, 2009), (Costa & Gaspar, 2014). They are also used for the calculation of some of the performance ratios that we will discuss later on.
The first moment (i.e. the first raw moment) of the returns distribution we want to apply in our study is the annualized mean return. Obviously, ceteris paribus, the higher annualized mean return, the better the portfolio insurance strategy becomes. This annualized mean return of the log returns is calculated as follows:\(^1\):

\[ A\text{Mean} = x\text{Mean} \times x = \frac{1}{n} \times \sum_{t=1}^{n} R_{p,t} \times x \]  

(2.5)

where \(x\text{Mean}\) can be daily, weekly, monthly, quarterly or annually returns, and \(x\) is an annualization factor, which equals 252, 52, 12, 4, or 1 respectively\(^2\); \(n\) is the number of time steps and \(R_{p,t}\) is the log return of the insured portfolio at step \(t\).

To be more explicit, in our study, the length of the investment horizon \(T\) is 5 years; the rebalancing frequency is every month, so \(n = 5 \times 12 = 60\), \(R_{p,t}\) is the monthly returns from month 1 to month 60 and \(x = 12\).

It is worth mentioning that equation (2.5) for calculating the first moment, and the equations below for other performance measures of the portfolio insurance strategies are just applied for the backtesting. To calculate any performance measure in the case of Monte Carlo simulation method, firstly, it is necessary to implement the respective equation for each path. Subsequently, we calculate the sample average of 10,000 paths for that performance measure. The result tables of the considered performance measures in Monte Carlo simulation and backtesting methods are shown in Chapter 3.

The second moment (i.e. the second central moment) of the returns distribution is the variance. However, its positive square root, aka standard deviation, is more commonly used in measuring performance. Ceteris paribus, the lower the standard deviation, the more preferred the strategy is by investors. In our research, we calculate the annualized standard deviation by the following mathematical formula:

\[ A\text{Std} = x\text{Std} \times \sqrt{x} = \sqrt{\frac{1}{n} \times \sum_{t=1}^{n} (R_{p,t} - x\text{Mean})^2} \times \sqrt{x} \]  

(2.6)

where \(n\), \(R_{p,t}\), and \(x\text{Mean}\) are similarly defined as in equation (2.5); however, unlike the previous equation, we apply the well-known square-root projection rule, i.e. \(\sqrt{x}\), which equals

\(^1\) Equation (2.5) shows the simple but exact formula to calculate the annualized mean return of the log returns, whereas it is just an approximation formula for the raw returns. This is one of the benefits of using log returns. For more details, see (Quantivity, 2011) and (Meucci, 2010).

\(^2\) \(x = 252\) for daily returns, which is conventionally accepted as the number of trading days in a year (Hull, 2015)
\[\sqrt{252}, \sqrt{52}, \sqrt{12}, \sqrt{4}, 1\] according to \(xStd\), which can be daily, weekly, monthly, quarterly or annually standard deviation (van Nierop & Bazant, 2016), (Meucci, 2010).

The less-frequent-used moments in practice are the remaining two. The third moment (i.e. the normalized third central moment) is the skewness. It determines the degree of asymmetry of the returns distribution. The distribution is called positively skewed (aka right skewed), if it has a long right tail, meaning that it has frequent small losses and a few extreme gains, which are located on the right side of the distribution. Vice versa, it is said to be negatively skewed (aka left-skewed), if it has a long left tail, and as a consequence, the distribution has many small gains but a few extreme losses. The normal distribution is symmetrical and has a skewness of zero (Wikipedia, 2017). Moreover, obviously, the more the returns distribution of the portfolio is positively skewed, the more attractive the insurance strategy becomes. The third moment we want to calculate in this study is the annualized skewness of the log returns distribution with the following formula:

\[
\text{ASKew} = xSkew \times \frac{1}{\sqrt{x}} = \frac{1}{n} \times \frac{\sum_{t=1}^{n} (R_{p,t} - xMean)^3}{xStd^3} \times \frac{1}{\sqrt{x}}
\]

where \(n, R_{p,t}, xMean, xStd\), are similarly defined as in equations (2.5) and (2.6); however, unlike the previous equations, we apply the scale \(1/\sqrt{x}\) equals \(1/\sqrt{252}, 1/\sqrt{52}, 1/\sqrt{12}, 1/\sqrt{4}, 1\) according to \(xSkew\), which can be daily, weekly, monthly, quarterly or annually skewness. For the detailed procedure to get the result, see (Meucci, 2010).

The fourth moment (i.e. the normalized fourth central moment\(^1\)) is the kurtosis. It depicts the flatness of the returns distribution. If the kurtosis coefficient equals to 3, the distribution is called mesokurtic, of which the most prominent example is the normal (aka Gaussian) distribution. If the returns distribution has kurtosis greater than 3, it is called leptokurtic (aka super-Gaussian), which has a thinner peak and allocates its “plentiful” remaining distribution to the fat tails (Kofidis, 2016). It can be interpreted that when there is less probability of outcomes around the mean, to assure that the total probability of all the outcomes is one, there must be a higher likelihood of extreme losses and gains as compensation. If the returns distribution has kurtosis less than 3, it is called platykurtic (aka sub-Gaussian), which has a broader peak and allocates its “inconsiderable” remaining distribution to the thinner tails (Kofidis, 2016). Contrary to the leptokurtic distribution, in the case of platykurtic distribution,

\(^1\) Some explanation for the technical terms of moment: the raw moments are the moments about zero, the central moments are the moments about the mean, and the n-th normalized (or standardized) moments are n-th central moment divided by \(\sigma^n\) with \(\sigma\) is the positive square root of the second central moment (Wikipedia, 2017).
there is a higher probability of outcomes around the mean, but the total probability sums up to one, there is less probability for the other extreme outcomes. Therefore, for the risk-averse investors, they prefer the returns distribution with smaller kurtosis coefficient, since they want to avoid the severe losses, even when they know that they may sacrifice some possible extreme gains. Vice versa, the risk-seeking investors love fat tails, which gives them the opportunity to obtain extreme profits even when they bet on extreme losses, therefore, they prefer bigger kurtosis coefficient. The fourth moment we want to calculate in this study is the annualized kurtosis of the log returns distribution, which is given by:

\[
AKurt = xKurt \times \frac{1}{x} = \frac{1}{n} \times \frac{\sum_{t=1}^{n}(R_{p,t} - xMean)^4}{xStd^4} \times \frac{1}{x}
\]  \hspace{1cm} (2.8)

where \( n, R_{p,t}, xMean, xStd \) are similarly defined as in equations (2.5) and (2.6); however, unlike the previous equations, we apply the scale \( 1/x \) equals \( 1/252, 1/52, 1/12, 1/4, 1 \) according to \( AKurt \), which can be daily, weekly, monthly, quarterly or annually skewness. For the detailed procedure to get the result, see (Meucci, 2010).

2.2.2. Performance ratios

a. The Sharpe ratio

One of the most widely-known performance measures to analyze the portfolio insurance strategies is the Sharpe ratio. This ratio determines the expected risk premium of the underlying portfolios (excess expected return over the risk-free rate) adjusted by its standard deviation (Sharpe, 1994). The higher the ratio, the better the performance of the portfolio insurance strategy becomes regarding the combination of risk and return. The Sharpe ratio is given by equation (2.9):

\[
Sharpe(r) = \frac{\mathbb{E}(R_p) - r}{\sigma_p}
\]  \hspace{1cm} (2.9)

where, \( \mathbb{E}(R_p) \) is the expected return of the portfolio; \( r \) is the risk-free interest rate and \( \sigma_p \) is the standard deviation of returns of the portfolio.

Although the Sharpe ratio is popular in use, it is not the universal performance metric and has its own limitations. Two of them are that the Sharpe ratio’s assumption about the normal distribution of returns of the portfolio is not always valid, and \( \sigma_p \) includes both upside and downside risks. They give rise to the other performance measures accounting for non-normality of return distributions or downside risk, including for example the Sortino ratio, the Omega ratio.
b. The Sortino ratio

The Sortino ratio is another return-to-risk ratio, and it is often called as a “Sharper” ratio (Rollinger & Hoffman, 2014), since it is a modification of the Sharpe ratio, considering downside volatility rather than the standard deviation (both downside and upside volatilities) as the measure of risk. Therefore, it outperforms the Sharpe ratio in many cases, particularly when valuing and comparing the performance of the portfolios which exhibit skewness in the return distributions. The Sortino ratio determines expected risk premium of the underlying portfolios adjusted by downside risk (Sortino & Price, 1994). The higher the ratio, the better the performance of the strategy is.

Before providing the formula to determine the Sortino ratio, we will introduce an approach to calculate the downside volatility, which is called Lower partial moment (LPM). LPM takes into consideration only the negative deviations between the realized returns of the portfolio and the minimal acceptable return (MAR) (in our study, it is the risk-free rate, r). In general, the LPM of order n, \( LPM_n(r) \), is calculated as in equation (2.10) (Eling & Schuhmacher, 2007). We note that, as implied by equation (2.10), \( LPM_0(r) \) is equivalent to the shortfall probability, \( LPM_1(r) \) can be interpreted as the expected shortfall that we mentioned in subsection 1.3.4, and \( LPM_2(r) \) as the semi-variance (Eling & Schuhmacher, 2007), which provides the calculation of downside volatility in equation (2.11).

\[
LPM_n(r) = \frac{1}{T} \sum_{t=1}^{T} \max[(r - R_{p,t}), 0]^n
\]  

where, \( R_{p,t} \) is the realized return of portfolio at time t.

Combined with the equation (2.10), the Sortino ratio is given by:

\[
Sortino (r) = \frac{\mathbb{E}(R_p) - r}{\sqrt{LPM_2(r)}}
\]  

where, \( \mathbb{E}(R_p) \) is the expected return of the portfolio.

c. Omega ratio

Regardless of their popularity and usefulness, the Sharpe and Sortino ratios fail to compare the performances between portfolios correctly, when two portfolios, say HR and LR, have identical negative risk premiums (\( \mathbb{E}(R_p) - r < 0 \)). According to equations (2.9), (2.10), both the denominators, \( \sigma_p \) and \( \sqrt{LPM_2(r)} \), are always positive, portfolio HR has higher risk than portfolio LR (meaning bigger denominator), the identical negative numerator leads to the fact that the Sharpe and Sortino ratios of HR becomes less negative (or higher) than those of LR. In standard parlance, portfolio HR with higher ratios are preferred to LR, however HR has higher risk for the same expected returns. This shows the inaccuracy in ranking the portfolios’
performances of the ratios (McLeod & van Vurren, 2004). Obviously, the Sharpe and Sortino ratios do not find out the better performing portfolio. Therefore, we need another alternative ratio to solve this weakness, which is called by its inventor as a universal performance measure, Omega ratio (Keating & Shadwick, 2002).

Furthermore, this recently-introduced ratio does not require the traditional normal distribution assumption for assets returns. It even deals with an inherent drawback of traditional risk-adjusted performance ratios, such as Sharpe and Sortino ratios, which is focusing on only the first two moments of the returns distribution, viz. mean and variance, yet ignoring the precise shape of the entire return distribution (described by higher moments of the returns distribution, including skewness and kurtosis).

In this study, the Omega ratio can be defined as a ratio of gains to losses of the portfolio with respect to the risk-free rate, r. The higher the Omega ratio, which indicates the higher cumulative probability of gains above r than that of losses below r, and as a result, the better the portfolio insurance strategy becomes. It can be determined by the mathematical definition in the footnote 1 below (Keating & Shadwick, 2002). However, to be consistent with equation (2.11) of Sortino ratio using $LPM_n(r)$, and for the simplicity sake of coding, we use the following formula for the Omega ratio, proved by (Kaplan, 2004) and used by (Eling & Schuhmacher, 2007):

$$Omega (r) = \frac{\mathbb{E}(R_p) - r}{LPM_1(r)} + 1$$

(2.12)

d. Floor protection ratio

We also use the Floor protection ratio as a performance measure, which is applied by (Do, 2002), and (Jiang, Ma, & An, 2009). It is defined as “the percentage of the realized portfolios that meet the insured value under a particular strategy” (Jiang, Ma, & An, 2009). This ratio measures the ability of each portfolio insurance strategy to guarantee the predetermined floor level at maturity. Investors prefer the strategy with a higher Floor protection ratio, which is consistent with their main goal to implement portfolio insurance strategies.

In conclusion, in this complicated finance world, to the extent of our knowledge, there is not (or at least has not been) a one-size-fits-all performance measure without any limitation,

---

\[ \Omega (r) = \frac{\int_{R_p_{min}}^{R_p_{max}} (1 - F(x)) dx}{\int_{R_p_{min}}^{R_p_{max}} F(x) dx}, \text{ where } F \text{ is the cumulative distribution function of returns defined on } (R_p_{min}, R_p_{max}) \text{ - the possible range of the portfolio returns} \text{ (Keating & Shadwick, 2002).} \]
assumption, and that can be applied in any case (even the so-called universal performance measure - Omega ratio\(^1\)). It is therefore important to combine these measures and compare the results to (somehow) depict a complete picture of the portfolio’s merits.

\(^1\) For more details, see (Caporin, Costola, Jannin, & Maillet, 2015) and (Vilkancas, 2014).
CHAPTER 3: RESULTS AND DEFINITION OF THE DOMINANT STRATEGY

So far, we have discussed all the related theoretical fundamentals and tools to carry out an extensive comparison of portfolio insurance strategies using the Monte Carlo simulation method and backtesting for the Vietnamese stock market. All of them can be presented briefly in the following lines so that readers can take a quick review before we introduce the results of our study.

- **Why to compare?** Find out the dominant strategy(ies) to guarantee a minimum loss, but maximum gain.

- **What to compare?** Portfolio insurance strategies: Buy and hold, Constant proportion. \((m = 3, 5)\), Option-based, VaR-based and Stop-loss with Floor = 100%.

- **How to compare?** Monte Carlo simulation method and backtesting.

- **By which measures to compare?** First four moments, Sharpe, Sortino, Omega and Floor protection ratios.

- **Where to compare?** A frontier market - the Vietnamese case.¹

We carry out all the whole procedure of this study with the help of Python programming language. The codes we have written are also provided in APPENDIX C.

3.1. Results of Monte Carlo simulation method

After running the Monte Carlo simulation method for the considered portfolio insurance strategies with the parameter setup discussed in section 2.1., we divide the results into six market scenarios, as presented before in Table 2.1, for a better comparison of the performance of the strategies. With respect to five out of the eight performance measures introduced in section 2.2. above (Annualized Mean, Annualized Standard deviation, Sharpe ratio, Sortino ratio, and Omega ratio), we rank all the strategies (in columns with the title “R”)². Subsequently, based on that, we also provide the overall rank to find the dominant strategy in each market scenario (in column with the title “O.R.”)³. For the other three measures (Annualized Skewness, Annualized Kurtosis, and Floor Protection ratio), we will only

---

¹ For more details, see Table 1.1.
² As mentioned in subsection 2.2.2., the Sharpe and Sortino ratios fail to compare the performances between the portfolios correctly, when the portfolios have negative risk premiums \(\mathbb{E}(R_p) - r < 0\). Therefore, in this case, we do not compare the portfolio with respect to these ratios.
³ The overall rank is the ranking of the arithmetic average of 5 ranks of the 8 performance measures, with the lowest value ranking first.
comment their results without ranking. Furthermore, to provide the whole distributions of the annualized mean returns of all portfolio insurance strategies under six market scenarios, we plot the respective histograms in APPENDIX A from Figure A. 1 to Figure A. 6.

3.1.1. Bear market scenario

a. Scenario 1 (μ = -15%, σ = 20%)

The series from Table 3.1 to Table 3.11 below rank the performance of various strategies (sorted by rows) at the end of their 5-year insurance period with respect to the measures (sorted by columns).

As can be seen in Table 3.1, the BH is the dominant strategy regarding all the ranked performance measures in the bear market scenario with low volatility. It is followed by the OBPI and VaRPI in turn, whereas the SLPI lies at the bottom with all of its measures ranking last.

The BH has the highest annualized mean of returns (nearly 2%) and the smallest standard deviation. In fact, most of the portfolio insurance strategies (except for the SLPI) have positive annual mean returns at maturity even in the bear market with big negative μ (μ = -15%) of the condition of the Monte Carlo simulation method for the underlying portfolio (VN30EQ Index). This fact shows the ability of the portfolio insurance strategies to limit the downside risk. Moreover, the annualized standard deviations of all the insured portfolio are low (less than 10%) in comparison with the volatility of 20% of the pure underlying portfolio.

<table>
<thead>
<tr>
<th>Strat.</th>
<th>AMean</th>
<th>R</th>
<th>AStd</th>
<th>ASkew</th>
<th>AKurt</th>
<th>Sharpe</th>
<th>R</th>
<th>Sortino</th>
<th>R</th>
<th>Omega</th>
<th>R</th>
<th>Prot.</th>
<th>O.R.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>0.0181</td>
<td>1</td>
<td>0.0274</td>
<td>-0.0405</td>
<td>0.0626</td>
<td>-0.8504</td>
<td>0</td>
<td>-0.5710</td>
<td>0</td>
<td>0.4188</td>
<td>1</td>
<td>1.0000</td>
<td>1</td>
</tr>
<tr>
<td>CPP13</td>
<td>0.0027</td>
<td>4</td>
<td>0.0566</td>
<td>-0.3778</td>
<td>0.5292</td>
<td>-0.6837</td>
<td>0</td>
<td>-0.8740</td>
<td>0</td>
<td>0.0672</td>
<td>4</td>
<td>0.6684</td>
<td>4</td>
</tr>
<tr>
<td>CPP15</td>
<td>0.0004</td>
<td>5</td>
<td>0.0738</td>
<td>-0.6567</td>
<td>1.0734</td>
<td>-0.5556</td>
<td>0</td>
<td>-0.8853</td>
<td>0</td>
<td>0.0251</td>
<td>5</td>
<td>0.1153</td>
<td>5</td>
</tr>
<tr>
<td>OBPI</td>
<td>0.0088</td>
<td>2</td>
<td>0.0521</td>
<td>-0.4185</td>
<td>0.5643</td>
<td>-0.6257</td>
<td>0</td>
<td>-0.7497</td>
<td>0</td>
<td>0.2038</td>
<td>2</td>
<td>0.9970</td>
<td>2</td>
</tr>
<tr>
<td>VaRPI</td>
<td>0.0074</td>
<td>3</td>
<td>0.0412</td>
<td>-0.0902</td>
<td>0.1546</td>
<td>-0.8252</td>
<td>0</td>
<td>-0.7969</td>
<td>0</td>
<td>0.1721</td>
<td>3</td>
<td>0.9510</td>
<td>3</td>
</tr>
<tr>
<td>SLPI</td>
<td>-0.0069</td>
<td>6</td>
<td>0.0972</td>
<td>-0.6670</td>
<td>1.0427</td>
<td>-0.4969</td>
<td>0</td>
<td>-0.9676</td>
<td>0</td>
<td>-0.1145</td>
<td>6</td>
<td>0.0193</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 3.1. Comparison of the performance of the strategies in Scenario 1

The bad performance of the SLPI can be interpreted more clearly by Figure 3.1, which shows one out of 10,000 paths simulated by the Monte Carlo method. From the beginning of the investment period to February 2016, when the value of the SLPI insured portfolio (blue dash-dot line) is greater than the present value of the floor (grey dot line), SLPI is exactly the

---

1 From the Table 3.1 to Table 3.11, all the values of the performance measures are rounded to 4 decimal places.
2 From now on, we call the present value of the floor simply the floor unless otherwise stated.
Index (red solid line), since before touching the floor, 100% of the SLPI insured portfolio is the risky asset. However, in March 2016, the Index pierces through the floor, the SLPI portfolio transfers from 100% risky assets to 100% risk-free assets growing at risk-free rate till the maturity. Obviously, the SLPI cannot meet the requirement of 100% Floor value at maturity. This makes the strategy perform worse in comparison with its counterparts. This gap risk is caused by the less-frequent rebalancing. We will verify in subsection 3.3. if the daily or weekly frequency may improve the situation.

Figure 3.1. The performance of the portfolio insurance strategies for VN30EQ during the insurance period with one out of 10,000 random paths of Scenario 1

Regarding the annualized skewness, the return distributions of all of the 6 strategies have negative skewness, meaning that there is more mass of returns which are smaller than the mean returns in comparison with those which are greater than the mean. The negative sign of this measure can be explained by the predetermined condition $\mu = -15\%$ of the market scenario 1 to create the 10,000 random paths of VN30EQ Index in 5 years through for the Monte Carlo simulation method.

In terms of the annualized kurtosis, all the strategies not only in the scenario 1, but also in all the other market scenarios have the platykurtic returns distribution (with AKurt is much
less than 3), meaning that they have a flat distribution with thin tails. These results are in line with the expectation of the risk-averse investors, seeking for the portfolio insurance strategy to avoid the extreme losses in the thinner tails.

The Floor Protection ratio of BH in Table 3.1 is the highest with the value of 1, meaning that 10,000 times out of 10,000, the final value of the BH portfolio exceeds the required floor value at maturity. Meanwhile, the SLPI strategy has the lowest ratio. There are only 193 times in which the investor with the SLPI strategy manages to keep his initial investment of 100,000,000 VND intact. The reason that lies behind may be again the frequency of rebalancing, especially in the case of the extremely bad market scenario. As we will see in other tables, the SLPI also have the lowest Floor Protection ratio in other market scenarios. However, the ratio becomes much better in the normal and bull market scenarios.

b. Scenario 2 ($\mu = -15\%$, $\sigma = 40\%$)

As shown in Table 3.2, the BH strategy still takes the lead, and the SLPI is at the bottom of the ranking. However, when the volatility ($\sigma$) increases from 20% to 40% (while $\mu$ remains the same), the VaRPI strategy takes the second place of OBPI, and the OBPI drops down to the 6th place. This may be explained by the underlying Black-Scholes formula of the OBPI strategy when it does not take into account the negative $\mu = -15\%$, but the volatility is, on the contrary, an important element in this famous formula. As can be easily seen in Box C of Table 1.2, higher volatility implies higher weight of the risky assets in the insured portfolio. In the bad market scenarios 1 and 2, the more exposure to the risky assets in scenario 2 leads to a worse performance with respects to all the ranked performance measures. To be more precise, the OBPI in scenario 2 has lower mean of returns, higher volatility, and lower Omega ratio.

<table>
<thead>
<tr>
<th>Strat.</th>
<th>AMean</th>
<th>AStd</th>
<th>ASkew</th>
<th>AKurt</th>
<th>Sharpe</th>
<th>Sortino</th>
<th>R</th>
<th>Omega</th>
<th>Prot.</th>
<th>O.R.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>0.0232</td>
<td>1</td>
<td>-0.0435</td>
<td>0.1070</td>
<td>-0.3018</td>
<td>-0.4290</td>
<td>0</td>
<td>0.5593</td>
<td>1.0000</td>
<td>1</td>
</tr>
<tr>
<td>CPP13</td>
<td>0.0076</td>
<td>3</td>
<td>-0.5461</td>
<td>0.9643</td>
<td>-0.2773</td>
<td>-0.7048</td>
<td>0</td>
<td>0.2015</td>
<td>0.3007</td>
<td>3</td>
</tr>
<tr>
<td>CPP15</td>
<td>0.0053</td>
<td>4</td>
<td>-0.8665</td>
<td>1.7261</td>
<td>-0.2523</td>
<td>-0.7131</td>
<td>0</td>
<td>0.1569</td>
<td>0.0678</td>
<td>4</td>
</tr>
<tr>
<td>OBPI</td>
<td>-0.0049</td>
<td>6</td>
<td>-0.2493</td>
<td>0.3720</td>
<td>-0.3224</td>
<td>-0.8770</td>
<td>0</td>
<td>-0.0240</td>
<td>0.1264</td>
<td>6</td>
</tr>
<tr>
<td>VARPI</td>
<td>0.0164</td>
<td>2</td>
<td>-0.0861</td>
<td>0.2218</td>
<td>-0.3067</td>
<td>-0.5592</td>
<td>0</td>
<td>0.4039</td>
<td>0.9523</td>
<td>2</td>
</tr>
<tr>
<td>SLPI</td>
<td>-0.0045</td>
<td>5</td>
<td>-0.8391</td>
<td>1.6510</td>
<td>-0.2638</td>
<td>-0.8133</td>
<td>0</td>
<td>-0.0183</td>
<td>0.0631</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 3.2. Comparison of the performance of the strategies in Scenario 2

On the contrary, as we calculate with the formula in Box D of Table 1.2, the VaRPI strategy with a higher volatility requires less exposure to the risky assets, which is necessary when the market witnesses a downturn. In fact, compared with scenario 1, the VaRPI strategy
of scenario 2 has higher mean of returns and higher Omega ratio, but still higher volatility due to the high volatility of the returns of the underlying portfolio.

Like in the previous scenario, the annualized volatilities of all strategies are decreased significantly (from 40% of the underlying portfolio to maximum 17% for SLPI, while BH has the lowest volatility of only 6%). Nevertheless, the portfolio insurance strategies still give higher mean of returns, which implies their enormous benefit for the investors to decrease their risk, but still get higher rate of returns in such bear market conditions.

![Performance of strategies with MC2](image)

Figure 3.2. The performance of the portfolio insurance strategies for VN30EQ during the insurance period with one out of 10,000 random paths of Scenario 2

Regarding the floor protection ratio, as expected, the more aggressive the strategy is, the more probable it breaches the floor, i.e. its protection ratio is lower. As mentioned in the property of the strategies in section 1.3, the value of the multiplier may imply the degree of the aggressiveness of the strategy. The BH strategy has the lowest m and the highest floor protection ratio, whereas the CPPI 3 and CPPI 5 have decreasing ratios in both 2 bear market scenarios. It is interesting to mention that the SLPI with its infinity multiplier has a much lower protection ratio. Furthermore, the low floor protection ratios of the strategies can be partly seen from one path of the returns of VN30EQ during 5 years in Figure 3.2, where the SLPI (dark
blue dash-dot line) and the OBPI (blue dot line) breach the floor and end up with the lower value of the insured portfolio than required. In this path, the CPPI 5 strategy (green dash line) exceeds the floor at maturity, in spite of its low floor protection ratio. However, this case of CPPI5 only happens 678 out of 10,000 times.

3.1.2. Normal market scenario

a. Scenario 3 (μ = 8%, σ = 20%)

In a better market scenario, the normal market with low volatility, the BH strategy is no longer the dominant, since it is considered the “do-nothing strategy” and the investor according to its rule does not expose the insured portfolio to more risky assets when the market witnesses a small uptrend and low volatility.

In this scenario, the more aggressive strategies, viz. CPPI5 and SLPI, rank first and second respectively regarding the mean returns measure. As a compensation, the more “risk-averse” strategies, namely BH, OBPI take the first and second places regarding the volatility.

The annualized skewness measures of all strategies (except for VaRPI) in this scenario are greater than those in the previous two bear market scenarios. It means that there is more mass of returns shifted from the left to the right of the mean returns. The annualized kurtosis of all strategies (again except for VaRPI) is also improved in favor of the risk-averse investor when this performance measure decreases in comparison with the bear market scenarios. The decline in this measure implies fewer extreme losses, which meets our expectation in the better market scenario. The explanation that we can give at the moment for the opposite behavior of the VaRPI is maybe because of its different mechanic for rebalancing. While other strategies have the Strong floor compliance, the VaRPI (and also ESPI) has Probabilistic floor compliance (Dichtl, Drobetz, & Wambach, 2015). We can also notice that the floor protection ratios of the VaRPI in all 6 scenarios are around 95%, which coincide with our predetermined confidence level α, while those ratios of other strategies, namely CPPI3, CPPI5, OBPI, and SLPI, vary a lot throughout the market scenarios (see Table A. 8 in APPENDIX B for more details).

The Sharpe and Sortino ratios of the strategies become positive in this normal scenario, because of the positive excess expected return over the risk-free rate. The Omega ratios of all strategies also improve significantly compared with those in the bear market scenarios.

In general, the portfolio insurance strategies tend to guarantee better the initial value of the investment after 5 years of a normal trend. Particularly, the floor protection ratios of CPPI3, CPPI5, and SLPI improve greatly in comparison with the bear markets (see Table A. 8).
Table 3.3. Comparison of the performance of the strategies in Scenario 3

Regarding Figure 3.3, in this path, the VaRPI fails to meet the floor value at maturity (which is rare with roughly 5%). However, as we can see, the SLPI and CPPI5 lines almost coincide with the Index line, which explains why their AMean roughly equal the AMean of the Index ($\mu = 8\%$). The graph also shows the bad performance of BH in comparison with other strategies.

![Performance of strategies with MC3](image)

Figure 3.3. The performance of the portfolio insurance strategies for VN30EQ during the insurance period with one out of 10,000 random paths of Scenario 3

b. Scenario 4 ($\mu = 8\%, \sigma = 40\%$)

Contrary to the previous scenario with the same mean return but lower volatility, the BH in scenario 4 takes the lead again in the overall rank (see Table A. 9). The OBPI loses the first
place in scenario 3 to the BH and ranks second in this scenario. This demotion can be explained in the same way as in the comments about the influence of high volatility to the performance of OBPI. The SLPI again ranks last, as the high volatility makes this strategy breach the floor easier with monthly rebalancing. It is interesting to see if the more frequent rebalancing can help this strategy promote in the ranking across the market scenarios. We will discuss this issue in more details in subsection 3.3.

<table>
<thead>
<tr>
<th>Strat.</th>
<th>AMean</th>
<th>R</th>
<th>AStd</th>
<th>R</th>
<th>ASkew</th>
<th>AKurt</th>
<th>Sharpe</th>
<th>R</th>
<th>Sortino</th>
<th>R</th>
<th>Omega</th>
<th>R</th>
<th>Prot.</th>
<th>O.R.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>0.0628</td>
<td>4</td>
<td>0.0909</td>
<td>1</td>
<td>0.0026</td>
<td>0.0562</td>
<td>0.2354</td>
<td>1</td>
<td>0.4441</td>
<td>1</td>
<td>1.6278</td>
<td>2</td>
<td>1.0000</td>
<td>1</td>
</tr>
<tr>
<td>CPPI3</td>
<td>0.0667</td>
<td>2</td>
<td>0.2028</td>
<td>3</td>
<td>-0.2792</td>
<td>0.4910</td>
<td>0.1248</td>
<td>3</td>
<td>0.1682</td>
<td>4</td>
<td>1.4721</td>
<td>4</td>
<td>0.7602</td>
<td>4</td>
</tr>
<tr>
<td>CPPI5</td>
<td>0.0603</td>
<td>5</td>
<td>0.2169</td>
<td>4</td>
<td>-0.5669</td>
<td>1.1345</td>
<td>0.0896</td>
<td>5</td>
<td>0.0269</td>
<td>5</td>
<td>1.2829</td>
<td>5</td>
<td>0.3240</td>
<td>5</td>
</tr>
<tr>
<td>OBPI</td>
<td>0.0743</td>
<td>1</td>
<td>0.2302</td>
<td>5</td>
<td>-0.0899</td>
<td>0.1532</td>
<td>0.1429</td>
<td>2</td>
<td>0.3357</td>
<td>2</td>
<td>1.7082</td>
<td>1</td>
<td>0.5539</td>
<td>2</td>
</tr>
<tr>
<td>VaRPI</td>
<td>0.0632</td>
<td>3</td>
<td>0.1849</td>
<td>2</td>
<td>-0.1023</td>
<td>0.3874</td>
<td>0.1179</td>
<td>4</td>
<td>0.2562</td>
<td>3</td>
<td>1.5172</td>
<td>3</td>
<td>0.9491</td>
<td>3</td>
</tr>
<tr>
<td>SLPI</td>
<td>0.0593</td>
<td>6</td>
<td>0.2351</td>
<td>6</td>
<td>-0.5746</td>
<td>1.1511</td>
<td>0.0761</td>
<td>6</td>
<td>0.0041</td>
<td>6</td>
<td>1.2512</td>
<td>6</td>
<td>0.2995</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 3.4. Comparison of the performance of the strategies in Scenario 4

![Figure 3.4](image)

Figure 3.4. The performance of the portfolio insurance strategies for VN30EQ during the insurance period with one out of 10,000 random paths of Scenario 4

The return to the first place of BH can be explained by the high $\sigma$ that we set for the simulated returns of the Index. It is worth mentioning that the higher volatility means both up
and down fluctuation of the returns of the Index, causing the increased AStd of all the strategies in this scenario. To have a more general view about the impact of the increased volatility of the Index’s returns on that of the strategies, the readers may refer to Table A. 2 and compare MC1 with MC2, MC3 with MC4, and MC5 with MC6.

Like the scenario 2 compared with scenario 1, the higher volatility in scenario 4 also harms the performance measures such as Sharpe, Sortino, Omega, and the floor protection ratios, making them reduce in values with respect to those in scenario 3.

In Figure 3.4, we can see that the strategies are more volatile (they range from 100M to roughly 600M, whereas those of scenario 3 range from 100M to 270M). The most fluctuated strategies with the Index are SLPI, CPPI5, CPPI3, and OBPI, while the VaRPI and BH are quite flat across 60 months.

3.1.3. Bull market scenario

a. Scenario 5 (μ = 15%, σ = 20%)

In the bull market with low volatility, the CPPI5 ranks first, and is followed by the other CPPI strategy with m = 3. The OBPI also performs well in this low volatility condition.

Again, the BH strategy is not the good choice, when it has really high implicit cost and halves the AMean of the underlying portfolio and ranks last. However, as can be seen somehow in Figure 3.5, a path of BH is flat during 5 years, thanks to its lowest volatility. In the meantime, the VaRPI has the highest AStd (approximately doubles the second highest one of SLPI and is much higher than the predetermined σ of the scenario). This behavior is clearly indicated in Figure 3.5. The high volatility of VaRPI also impacts its Sharpe, Sortino, and Omega ratios, leading to its last position in the Overall ranking.

<table>
<thead>
<tr>
<th>Strat.</th>
<th>AMean</th>
<th>R</th>
<th>AStd</th>
<th>R</th>
<th>ASkew</th>
<th>AKurt</th>
<th>Sharpe</th>
<th>R</th>
<th>Sortino</th>
<th>R</th>
<th>Omega</th>
<th>R</th>
<th>Prot.</th>
<th>O.R.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>0.0710</td>
<td>6</td>
<td>0.0484</td>
<td>1</td>
<td>0.0113</td>
<td>0.0244</td>
<td>0.6116</td>
<td>1</td>
<td>0.7950</td>
<td>6</td>
<td>1.8816</td>
<td>6</td>
<td>1.0000</td>
<td>5</td>
</tr>
<tr>
<td>CPPI3</td>
<td>0.1233</td>
<td>4</td>
<td>0.1491</td>
<td>3</td>
<td>-0.0362</td>
<td>0.0616</td>
<td>0.5493</td>
<td>2</td>
<td>1.6119</td>
<td>4</td>
<td>3.3885</td>
<td>3</td>
<td>1.0000</td>
<td>2</td>
</tr>
<tr>
<td>CPPI5</td>
<td>0.1321</td>
<td>2</td>
<td>0.1689</td>
<td>4</td>
<td>-0.0688</td>
<td>0.1066</td>
<td>0.5370</td>
<td>4</td>
<td>1.7022</td>
<td>1</td>
<td>3.5956</td>
<td>1</td>
<td>0.9815</td>
<td>1</td>
</tr>
<tr>
<td>OBPI</td>
<td>0.1216</td>
<td>5</td>
<td>0.1469</td>
<td>2</td>
<td>-0.0208</td>
<td>0.0708</td>
<td>0.5459</td>
<td>3</td>
<td>1.6183</td>
<td>3</td>
<td>3.3637</td>
<td>4</td>
<td>0.9972</td>
<td>3</td>
</tr>
<tr>
<td>VaRPI</td>
<td>0.1487</td>
<td>1</td>
<td>0.3431</td>
<td>6</td>
<td>-0.1947</td>
<td>0.4651</td>
<td>0.3127</td>
<td>6</td>
<td>1.0673</td>
<td>5</td>
<td>3.0607</td>
<td>5</td>
<td>0.9360</td>
<td>6</td>
</tr>
<tr>
<td>SLPI</td>
<td>0.1320</td>
<td>3</td>
<td>0.1749</td>
<td>5</td>
<td>-0.1262</td>
<td>0.2037</td>
<td>0.5180</td>
<td>5</td>
<td>1.6693</td>
<td>2</td>
<td>3.5954</td>
<td>2</td>
<td>0.7604</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 3.5. Comparison of the performance of the strategies in Scenario 5

The bad performance of VaRPI can be explained again by the formula in Box D of Table 1.2. The weight of risky assets in the portfolio increases with higher μ and lower σ. Therefore, the combination of a high enough μ and a low enough σ makes the VaRPI exposed more to the risky assets, causing an, even more, volatility than the Index itself. As a result, the VaRPI has
the highest AMean in the uptrend scenario. However, it cannot compensate for the unusually high volatility, leading to smaller ratios that have volatility (or downside risk) as the denominator (the Sharpe ratio ranks last, whereas the downside risk adjusted Sortino and Omega ratios rank above only those of BH).

![Performance of strategies with MC5](image)

**Figure 3.5.** The performance of the portfolio insurance strategies for VN30EQ during the insurance period with one out of 10,000 random paths of Scenario 5

b. Scenario 6 ($\mu = 15\%, \sigma = 40\%$)

In the bull market scenario with high volatility condition, the OBPI wins the first place that was of CPPI5 in the previous scenario. Moreover, as expected, the SLPI in the scenario with high volatility ranks last with the monthly rebalancing.

Although the AMean and AStd of OBPI rank first and last respectively like those of VaRPI in scenario 5, the high mean returns of the strategy still can make up for the high volatility (and downside risk). Therefore, the performance ratios of OBPI do not suffer as much as other strategies from their large denominators (volatility or downside risk) and rank the top places thanks to the high AMean that OBPI generates.

The AMean and AStd of BH in this scenario obtain the same rank as in the other bull market scenario with the same explanation about this “do-nothing” strategy. It is worth
emphasizing that the BH strategy reduces about 4 times the AStd of the underlying portfolio, while the second least volatile strategy reduces it roughly 1.7 times. In fact, the BH is the strategy which reduces most the volatility of the need-to-be-insured portfolio in all market scenarios.

As predicted, the higher volatility condition even in a bull market of scenario 6 still generally harms the efficiency of the risk-adjusted performance measures, namely Sharpe, Sortino, Omega ratios, of all strategies.

<table>
<thead>
<tr>
<th>Strat.</th>
<th>AMean</th>
<th>R</th>
<th>AStd</th>
<th>R</th>
<th>ASkew</th>
<th>AKurt</th>
<th>Sharpe</th>
<th>R</th>
<th>Sortino</th>
<th>R</th>
<th>Omega</th>
<th>Prot.</th>
<th>O.R.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>0.0826</td>
<td>6</td>
<td>0.1034</td>
<td>1</td>
<td>0.0121</td>
<td>0.0560</td>
<td>0.3985</td>
<td>1</td>
<td>0.8640</td>
<td>2</td>
<td>2.1949</td>
<td>3</td>
<td>1.000</td>
</tr>
<tr>
<td>CPPI3</td>
<td>0.1076</td>
<td>2</td>
<td>0.2345</td>
<td>2</td>
<td>-0.2051</td>
<td>0.3673</td>
<td>0.2823</td>
<td>3</td>
<td>0.7326</td>
<td>3</td>
<td>2.3673</td>
<td>2</td>
<td>0.8581</td>
</tr>
<tr>
<td>CPPI5</td>
<td>0.0983</td>
<td>4</td>
<td>0.2371</td>
<td>3</td>
<td>-0.4745</td>
<td>0.9516</td>
<td>0.2400</td>
<td>4</td>
<td>0.5208</td>
<td>5</td>
<td>2.0841</td>
<td>5</td>
<td>0.4406</td>
</tr>
<tr>
<td>OBPI</td>
<td>0.1200</td>
<td>1</td>
<td>0.2583</td>
<td>6</td>
<td>-0.0548</td>
<td>0.1128</td>
<td>0.3043</td>
<td>2</td>
<td>0.9849</td>
<td>1</td>
<td>2.7260</td>
<td>1</td>
<td>0.7004</td>
</tr>
<tr>
<td>VaRPI</td>
<td>0.0955</td>
<td>5</td>
<td>0.2466</td>
<td>4</td>
<td>-0.1156</td>
<td>0.4370</td>
<td>0.2194</td>
<td>6</td>
<td>0.6543</td>
<td>4</td>
<td>2.1926</td>
<td>4</td>
<td>0.9463</td>
</tr>
<tr>
<td>SLPI</td>
<td>0.0993</td>
<td>3</td>
<td>0.2563</td>
<td>5</td>
<td>-0.5043</td>
<td>1.0064</td>
<td>0.2259</td>
<td>5</td>
<td>0.5059</td>
<td>6</td>
<td>2.0755</td>
<td>6</td>
<td>0.4031</td>
</tr>
</tbody>
</table>

Table 3.6. Comparison of the performance of the strategies in Scenario 6

Figure 3.6. The performance of the portfolio insurance strategies for VN30EQ during the insurance period with one out of 10,000 random paths of Scenario 6
In terms of floor protection ratio, in this scenario and other ones, we can notice that the BH always has the lowest standard deviation thanks to its fixed allocation of risk-free and risky assets (see Table A. 2), therefore it also sustains the floor protection ratio (equals 1). Furthermore, unsurprisingly, the floor protection ratios of the other strategies endure the decrease due to higher standard deviation of the Index compared with scenario 5. However, they are still greater than the protection ratios of MC2 and MC4 (see Table A. 8).

Referring to Figure 3.6, the behaviors of the Index, CPPI5, CPPI3 and OBPI are similar to those of the Index and the strategies in the other scenarios with high volatility (scenarios 2 and 4), they fluctuate considerably during the insurance period, and in fact, the range is even broader due to the higher \( \mu \).

### 3.2. Results of Backtesting

After analyzing the performance results of the strategies using the Monte Carlo simulation method, we now apply the real index VN30EQ from April 2012 to April 2017 to these same strategies and compare their merits. The similar ranking for the strategies in this backtesting method is also provided in Table 3.7. However, it is worth pointing out that the values of the measures in Table 3.7 are real values, not the estimate (sample average of 10,000 simulated paths), and the lines in Figure 3.7 are their actual paths.

<table>
<thead>
<tr>
<th>Strat.</th>
<th>AMean</th>
<th>R</th>
<th>AStd</th>
<th>R</th>
<th>ASkew</th>
<th>R</th>
<th>AKurt</th>
<th>R</th>
<th>Sharpe</th>
<th>R</th>
<th>Sortino</th>
<th>R</th>
<th>Omega</th>
<th>R</th>
<th>O.R.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>0.0577</td>
<td>5</td>
<td>0.0446</td>
<td>1</td>
<td>0.0845</td>
<td>2</td>
<td>0.0047</td>
<td>1</td>
<td>0.3656</td>
<td>5</td>
<td>0.4208</td>
<td>5</td>
<td>1.4461</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>CPPI3</td>
<td>0.0634</td>
<td>4</td>
<td>0.1274</td>
<td>2</td>
<td>-0.0293</td>
<td>0</td>
<td>0.0123</td>
<td>5</td>
<td>0.1729</td>
<td>4</td>
<td>0.4366</td>
<td>4</td>
<td>1.5558</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>CPPI5</td>
<td>0.0725</td>
<td>4</td>
<td>0.1797</td>
<td>4</td>
<td>-0.0064</td>
<td>0</td>
<td>0.0047</td>
<td>3</td>
<td>0.1731</td>
<td>4</td>
<td>0.5245</td>
<td>3</td>
<td>1.7289</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>OBPI</td>
<td>0.0820</td>
<td>2</td>
<td>0.1388</td>
<td>3</td>
<td>0.0068</td>
<td>1</td>
<td>-0.0130</td>
<td>3</td>
<td>0.2923</td>
<td>3</td>
<td>0.7919</td>
<td>2</td>
<td>2.0337</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>SLPI</td>
<td>0.1178</td>
<td>1</td>
<td>0.2125</td>
<td>5</td>
<td>0.1094</td>
<td>2</td>
<td>0.0229</td>
<td>2</td>
<td>0.3595</td>
<td>2</td>
<td>1.2463</td>
<td>1</td>
<td>2.7644</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.7. Comparison of the performance of the strategies in the real situation of VN30EQ from April 2012 to April 2017.

The real \( \mu \) is 12%, which lies in the middle of the \( \mu \) of the normal and bull markets, whereas the \( \sigma \) is approximately 21%, which is quite similar the low volatility condition. It can be easily seen in Figure 3.7 that the VN30EQ generally experiences an uptrend. However, the Index witnessed a downturn from May 2012, and continued to perform poorly until the end of the year, when the Vietnamese banking tycoon Nguyen Duc Kien was announced to be arrested for economic violation in August 2012. The other decline in the Vietnamese stock market generally and the VN30EQ particularly was from September 2009, due to the territorial disputes in the South China Sea, and the drop of the world’s oil price (Trieu & Shalamov, 2015).
In this real path, since the Index never breaches the floor and increases, the SLPI is the dominant strategy when it imitates perfectly the Index, while other portfolio insurance strategies, including the risk-free assets in the portfolio, lost their chance to participate in the whole upturn of the Index.

However, we can see that the AStd of SLPI ranks last, equaling the volatility of the Index. In fact, the high volatility does not provide enough information about the behavior of the strategy during the insurance period, since the volatility takes into account not only the downside risk but also the often-preferred upside risk. This betrays the weakness of this standard measure for real risk. However, we can notice that the Sortino and Omega ratios which take into consideration only the downside risk are much higher in the case of SLPI compared with other strategies. The reason lying behind may be that the upside volatility constitutes the majority of the volatility, meaning that the downside risk, in this case, is small even when the total volatility is high.

![Performance of strategies with Backtesting](image)

**Figure 3.7.** The performance of the portfolio insurance strategies for VN30EQ from the beginning of April 2012 to the end of April 2017 with Backtesting
3.3. Comparison with different rebalancing frequencies

As mentioned in the comments in the previous section, in this part we try to compare the performance of the strategies with respect to different rebalancing frequencies, i.e. our benchmark monthly rebalancing with weekly and daily ones. Furthermore, we choose only market scenarios 2 and 4 for our comparison purpose, since these are the cases when the portfolio insurance strategies are more useful (bear and normal markets with high volatility).

a. Weekly rebalancing

There are some modifications of the data and parameters in the Monte Carlo simulation and backtesting methods. From the same dataset of daily price, we select the time series of the prices of last trading days of the 262 weeks of VN30EQ and calculate the time series of 261 natural logarithmic month returns. The risk-free rate is the mean of the 1-week VNIBOR rates throughout the investment period and equals 3.2% per annum. The insurance period is still 5 years but is divided into 261 sub-periods or 261 weeks.¹

<table>
<thead>
<tr>
<th>Strat.</th>
<th>AMean</th>
<th>R</th>
<th>AStd</th>
<th>R</th>
<th>ASkew</th>
<th>R</th>
<th>AKurt</th>
<th>R</th>
<th>Sharpe</th>
<th>R</th>
<th>Sortino</th>
<th>R</th>
<th>Omega</th>
<th>R</th>
<th>Prot.</th>
<th>O.R.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>0.0191</td>
<td>1</td>
<td>0.0487</td>
<td>1</td>
<td>-0.0110</td>
<td>0.0306</td>
<td>-0.2074</td>
<td>0</td>
<td>-0.4017</td>
<td>0</td>
<td>0.5982</td>
<td>1</td>
<td>1.0000</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPP13</td>
<td>0.0066</td>
<td>3</td>
<td>0.0994</td>
<td>3</td>
<td>-0.1786</td>
<td>0.2851</td>
<td>-0.3501</td>
<td>0</td>
<td>-0.7746</td>
<td>0</td>
<td>0.2022</td>
<td>3</td>
<td>0.5663</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPP15</td>
<td>0.0053</td>
<td>4</td>
<td>0.1134</td>
<td>4</td>
<td>-0.3831</td>
<td>0.6754</td>
<td>-0.3183</td>
<td>0</td>
<td>-0.8028</td>
<td>0</td>
<td>0.1664</td>
<td>4</td>
<td>0.0965</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OBPI</td>
<td>-0.0121</td>
<td>6</td>
<td>0.1477</td>
<td>6</td>
<td>-0.0694</td>
<td>0.1011</td>
<td>-0.3622</td>
<td>0</td>
<td>-1.2385</td>
<td>0</td>
<td>-0.3381</td>
<td>6</td>
<td>0.1274</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VaRPI</td>
<td>0.0130</td>
<td>2</td>
<td>0.0724</td>
<td>2</td>
<td>-0.0428</td>
<td>0.1507</td>
<td>-0.3923</td>
<td>0</td>
<td>-0.5844</td>
<td>0</td>
<td>0.4089</td>
<td>2</td>
<td>0.9496</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLPI</td>
<td>0.0005</td>
<td>5</td>
<td>0.1390</td>
<td>5</td>
<td>-0.5132</td>
<td>0.9644</td>
<td>-0.2942</td>
<td>0</td>
<td>-0.9184</td>
<td>0</td>
<td>0.0296</td>
<td>5</td>
<td>0.0461</td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.8. Comparison of the performance of the strategies in Scenario 2 with weekly rebalancing

As can be seen in Table 3.8, in the bear market scenario with high volatility, the change from monthly to weekly rebalancing does not alter the ranks with respect to the performance measures and the overall rank of the portfolio insurance strategies.

However, in general, the increase in rebalancing frequency makes most strategies perform worse in terms of AMean (except for SLPI). On the contrary, the AStd, ASkew, AKurt, and Omega of the strategies tend to be improved for the risk-averse investors (except for the Omega ratio of OBPI). Contrary to expectations, the floor protection ratios of the strategies are not enhanced notably. In fact, there is a blended result, indicating that the ratio of BH remains the

---

¹ In the real path, there is 261 weeks over 5 years, instead of 260 weeks in theory. There is 52 weeks and 1 day for the common year and 52 weeks and 2 days for the leap year, instead of a rounded value 52 weeks per year theoretically. We use 261 weeks to make the research nearer to the reality, however, there is not so much difference between the two choices.
same, those of CPPI3, CPPI5, OBPI improve (especially CPPI3), whereas those of VaRPI and SLPI do not.

<table>
<thead>
<tr>
<th>Strat.</th>
<th>AMean</th>
<th>R</th>
<th>AStd</th>
<th>R</th>
<th>ASkew</th>
<th>AKurt</th>
<th>Sharpe</th>
<th>R</th>
<th>Sortino</th>
<th>R</th>
<th>Omega</th>
<th>R</th>
<th>Prot.</th>
<th>O.R.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>0.0524</td>
<td>5</td>
<td>0.0757</td>
<td>1</td>
<td>0.0015</td>
<td>0.0176</td>
<td>0.3065</td>
<td>1</td>
<td>0.5936</td>
<td>2</td>
<td>1.6797</td>
<td>3</td>
<td>1.0000</td>
<td>2</td>
</tr>
<tr>
<td>CPPI3</td>
<td>0.0614</td>
<td>2</td>
<td>0.1808</td>
<td>2</td>
<td>-0.0827</td>
<td>0.1356</td>
<td>0.1781</td>
<td>3</td>
<td>0.4431</td>
<td>3</td>
<td>1.7946</td>
<td>2</td>
<td>0.9270</td>
<td>2</td>
</tr>
<tr>
<td>CPPI5</td>
<td>0.0549</td>
<td>3</td>
<td>0.1855</td>
<td>4</td>
<td>-0.2267</td>
<td>0.3977</td>
<td>0.1385</td>
<td>4</td>
<td>0.2231</td>
<td>5</td>
<td>1.5586</td>
<td>5</td>
<td>0.4520</td>
<td>5</td>
</tr>
<tr>
<td>OBPI</td>
<td>0.0727</td>
<td>1</td>
<td>0.2378</td>
<td>6</td>
<td>-0.0229</td>
<td>0.0409</td>
<td>0.1829</td>
<td>2</td>
<td>0.6982</td>
<td>1</td>
<td>2.1528</td>
<td>1</td>
<td>0.5502</td>
<td>1</td>
</tr>
<tr>
<td>VaRPI</td>
<td>0.0529</td>
<td>4</td>
<td>0.1819</td>
<td>3</td>
<td>-0.0794</td>
<td>0.2742</td>
<td>0.1303</td>
<td>5</td>
<td>0.3511</td>
<td>4</td>
<td>1.5931</td>
<td>4</td>
<td>0.9468</td>
<td>4</td>
</tr>
<tr>
<td>SLPI</td>
<td>0.0490</td>
<td>6</td>
<td>0.1930</td>
<td>5</td>
<td>-0.3921</td>
<td>0.7391</td>
<td>0.1026</td>
<td>6</td>
<td>0.0402</td>
<td>6</td>
<td>1.3620</td>
<td>6</td>
<td>0.2226</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 3.9. Comparison of the performance of the strategies in Scenario 4 with weekly rebalancing

In scenario 4 (normal market with high volatility) with weekly rebalancing, there are some changes in the Overall ranks of the strategies. The OBPI takes the first place of BH in the monthly rebalancing; however, the SLPI still remains at the bottom of the ranking.

Similar to the scenario 2, there is an unfavorable trend for AMean in all strategies when we change from monthly to weekly in all strategies. However, the other performance measures, namely Askew, AKurt, Sharpe, Sortino, and Omega ratios, show better results for risk-averse investors, when they rebalance more often. Regarding the floor protection ratio, there is a mixed result. The floor protection ratios with weekly rebalancing in Scenario 2 are higher than the ones with monthly rebalancing in case CPPI3, CPPI 5 and remain the same in case of BH with perfect floor protection. In the meantime, higher rebalancing frequency causes lower floor protection ratios for OBPI, VaRPI, and SLPI.

Comparing the backtesting of weekly rebalancing (Figure 3.8) and monthly rebalancing (Figure 3.7), we can notice the highly path dependence of the SLPI, which we mention in the first chapter. As we can see in Figure 3.8 for weekly rebalancing, the SLPI portfolio breaches the floor in the third week of September 2012. According to its rule, all the risky assets in the SLPI are transferred into risk-free assets, grows at risk-free rate till the end of the insurance period and loses its chance to participate in the strong uptrend right afterward. However, following the discipline of monthly rebalancing, the investor rebalances his portfolio every first day of the month, so he does nothing even when the value of the SLPI reduces under the floor in the interim of the month, therefore, he avoids a disastrous opportunity cost of VND 70 million when transferring the whole insured portfolio into risk-free assets and is stuck there with them till April 2017, seeing the risky assets, with great regret, rises to VND 170 million at maturity.
Furthermore, there is another minor change in the ranking of the final values of the strategies, besides the dropdown of SLPI from the first place to the last one. CPPI3 tends to perform better than CPPI5 when we change the frequencies from monthly to weekly, and they exchange their place. This can be explained by the same reasoning that we use for SLPI. As we mention in subsection 1.3.5., the SLPI is the CPPI with an infinity multiplier, meaning that it is an extremely aggressive strategy. CPPI5 with its higher multiplier is a more aggressive strategy vis-à-vis CPPI3 and performs worse when we increase the frequency of rebalancing.

Figure 3.8. The performance of the portfolio insurance strategies for VN30EQ from the beginning of April 2012 to the end of April 2017 with Backtesting (weekly rebalancing)

b. Daily rebalancing

Similar to weekly rebalancing, we also modify the inputs. The time series for the price of the Index includes 1261 days, from which we calculate 1260 natural logarithmic daily returns, for the backtesting method. The risk-free rate is changed to the mean of the 1-day VNIBOR rates, which equals 2.92% per annum.
Once again, in the bear market with high volatility, the Ranks of strategies according to the measures and the Overall rank of all strategies do not change when we shift from monthly to daily rebalancing.

This change in the rebalancing frequency also makes 4 out of 6 strategies perform worse with respect to AMean (the two exceptions are CPPI5 and SLPI). Quite similar to the change from monthly to weekly frequencies, in scenario 2, the strategies become less volatile (except for OBPI), the ASkew, AKurt of all strategies improve for the risk-averse investors, while the Omega ratio witnesses a mixed result, where the ratio is generally improved in case of CPPI with $m = 1$ (BH), $m = 3$, $m = 5$, and $m = +\infty$ (SLPI), while deteriorates in the other 2 strategies.

In a similar mixed vein, the floor protection ratio of the BH remains unchanged, those of CPPI3, CPPI5 improve, while those of OBPI, VarPI, and SLPI are worsened.

Table 3.11. Comparison of the performance of the strategies in Scenario 4 with daily rebalancing

In this normal market scenario with high volatility, in the case of daily rebalancing, there are again some changes in the Overall ranks of the strategies. The OBPI with CPPI3 take the first place of BH in the monthly rebalancing; however, the VarPI drops down to the bottom of the ranking.

Like in scenario 2, AMean of all strategies decrease, when we change from monthly to daily in all strategies, while we do not notice complete improvements in AStd, ASkew, and AKurt of all strategies as compensation. Regarding the performance ratios, they generally
perform better with a more frequent rebalancing; however, it is not the case for the floor protection ratio. There is a similar mixed result to that of scenario 2 with daily rebalancing, in which OBPI, VaRPI, and SLPI fail to protect better the floor value at maturity even when we rebalance more often the portfolio according to the movement of the Index.

Figure 3.9. The performance of the portfolio insurance strategies for VN30EQ from the beginning of April 2012 to the end of April 2017 with Backtesting (daily rebalancing)

Comparing the backtesting result of daily rebalancing (Figure 3.9) and monthly rebalancing (Figure 3.7), we can notice again the weakness about path dependence of SLPI. The Index makes a deep downturn in September 2012, which makes the SLPI, following its own rule, transform all the portfolio to risk-free assets and can never participate in the strong upturn of the market from December 2012. The rank of the final values of the strategies remains the same like in the weekly rebalancing.

In this subsection, our main aim is to verify if the results for all portfolio insurance strategies improve significantly if we rebalance more often. In our study, we do not consider the transaction cost for simplicity sake. However, even with that simplification, the results using weekly and daily rebalancing frequencies, do not show a clear improvement in the two considered scenarios 2 and 4. To be more precise, the annualized mean returns of all strategies

74
are decreased; however, we do not obtain a generally\(^1\) better results in other performance measures. Therefore, it seems optimal to rebalance every month rather than every week or every day.

---

\(^1\) In fact, there are still some individual cases when opposite results occur.
CONCLUSION AND FURTHER RESEARCH

This study contributes to the literature a broader comparison of the performance of the most widely-adopted portfolio insurance strategies. Specifically, we apply our comparison of the strategies to the real data of the Vietnamese market, which is a promising frontier market, but still is lightly researched by both the academics and practitioners.

We find in the Bear market with $\sigma = 20\%$ and $40\%$, and the Normal market with $\sigma = 40\%$, that BH dominates other strategies, while SLPI performs very badly. This result is in conformity with (Khuman, Maringer, & Constantinou, 2008), who finds that BH with $\sigma > 10\%$ dominates CPPI3 and CPPI5, and partly with (Costa & Gaspar, 2014), who conclude that both BH and SLPI outperforms the other strategies.

In the Normal market and Bull market with high volatility, the OBPI are preferred, whereas the CPPI5, CPPI3 are the two best strategies in the bull market scenario with low volatility. This finding concurs with the conclusion of (Bertrand & Prigent, 2005), who find that OBPI dominates CPPI in the normal market and bull market with high volatility; and CPPI dominates OBPI in the bull market and no drop.

In general, according to Table A. 9, the OBPI seems to be superior in all market conditions, followed by the BH. The VaRPI and SLPI take the two last positions.

According to the real data of the Vietnamese stock market from April 2012 to April 2017, the Backtesting method shows that the SLPI is the best strategy that the investor should have chosen at the beginning of the insurance period. However, the OBPI and BH are still good choices for the investors if they want to insure their portfolio. The dominance of SLPI in this real data analysis can be explained by Figure 3.7, when the SLPI $\xi$ coincides exactly the Index when it never touches the floor and the Index witnesses a really strong uptrend during the insurance period.

In this study, we also compare our benchmark monthly rebalancing with the daily and weekly frequencies to verify if it is worth rebalancing more often. We conclude that it is optimal to rebalance every month, rather than on daily or weekly basis.

However, this research still has room for development in further studies. Firstly, we assume there are no transactions costs, and the stocks are perfectly divisible. Secondly, we applied the assumption of Geometric Brownian Stochastic Process for stock prices, while it is not always the case in reality. Later on, we can apply more sophisticated models to address some limitations of the Geometric Brownian motion, including ignoring jump in price or constant volatility, to describe better the behaviour of the stock price, such as the Merton Jump.
Diffusion Model (Merton, 1976), and the Heston Stochastic Volatility Model (Heston, 1993). Furthermore, with a reference from (Agić-Šabeta, 2016), it would be interesting to apply the Cox Ingersoll Ross (CIR) stochastic process to describe the risk-free rate evolution, instead of using a constant risk-free rate over the whole insurance period. Thirdly, we cannot apply the real data to the VaRPI due to the lack of real data to calculate the VaR for the strategy. Furthermore, although ESPI is a very promising portfolio insurance strategy, we do not investigate its performance in this study. Fourthly, in this study, we already check the sensitivity of the performance measures in all strategies to the rebalancing frequency. However, it may be useful to deepen our research to find out the most suitable frequency for each strategy. Fifthly, as we observe the weakness of highly path dependence of the standard SLPI in weekly and daily rebalancing frequencies, in further research, we plan to include the modified version of SLPI, proposed by (Bird, Dennis, & Tippett, 1988). We will evaluate the performance of the modified version to see if it actually outperforms the standard SLPI and other strategies in the Vietnamese case. Last but not least, in this study and in most of the researches in the field, to the best of our knowledge, the authors concentrate on only the large-cap stocks, while overlooking the small-cap stocks with their inherent volatile property, which indeed need the portfolio insurance strategies more than their counterparts.
Figure A.1. Annualized mean distributions with market scenario 1 (μ = -15%, σ = 20%)
Figure A. 2. Annualized mean distributions with market scenario 2 ($\mu = -15\%, \sigma = 40\%)
Figure A. 3. Annualized mean distributions with market scenario 3 ($\mu = 8\%, \sigma = 20\%$)
Figure A. 4. Annualized mean distributions with market scenario 4 ($\mu = 8\%, \sigma = 40\%$)
Figure A. 5. Annualized mean distributions with market scenario 5 ($\mu = 15\%, \sigma = 20\%$)
Figure A. 6. Annualized mean distributions with market scenario 6 (μ = 15%, σ = 40%)
APPENDIX B

<table>
<thead>
<tr>
<th>Strat.</th>
<th>MC1</th>
<th>MC2</th>
<th>MC3</th>
<th>MC4</th>
<th>MC5</th>
<th>MC6</th>
<th>BT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>0.0181</td>
<td>0.0232</td>
<td>0.0527</td>
<td>0.0628</td>
<td>0.0710</td>
<td>0.0826</td>
<td>0.0577</td>
</tr>
<tr>
<td>CPPI3</td>
<td>0.0027</td>
<td>0.0076</td>
<td>0.0673</td>
<td>0.0667</td>
<td>0.1233</td>
<td>0.1076</td>
<td>0.0634</td>
</tr>
<tr>
<td>CPPI5</td>
<td>0.0004</td>
<td>0.0053</td>
<td>0.0693</td>
<td>0.0603</td>
<td>0.1321</td>
<td>0.0983</td>
<td>0.0725</td>
</tr>
<tr>
<td>OBPI</td>
<td>0.0088</td>
<td>-0.0049</td>
<td>0.0668</td>
<td>0.0743</td>
<td>0.1216</td>
<td>0.1200</td>
<td>0.082</td>
</tr>
<tr>
<td>VaRPI</td>
<td>0.0074</td>
<td>0.0164</td>
<td>0.0624</td>
<td>0.0632</td>
<td>0.1487</td>
<td>0.0955</td>
<td>-</td>
</tr>
<tr>
<td>SLPI</td>
<td>-0.0069</td>
<td>-0.0045</td>
<td>0.0686</td>
<td>0.0593</td>
<td>0.1320</td>
<td>0.0993</td>
<td>0.1178</td>
</tr>
</tbody>
</table>

Table A. 1. The comparison of the AMean of the strategies across market scenarios

<table>
<thead>
<tr>
<th>Strat.</th>
<th>MC1</th>
<th>MC2</th>
<th>MC3</th>
<th>MC4</th>
<th>MC5</th>
<th>MC6</th>
<th>BT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>0.0274</td>
<td>0.0603</td>
<td>0.0419</td>
<td>0.0909</td>
<td>0.0484</td>
<td>0.1034</td>
<td>0.0446</td>
</tr>
<tr>
<td>CPPI3</td>
<td>0.0566</td>
<td>0.1219</td>
<td>0.1228</td>
<td>0.2028</td>
<td>0.1491</td>
<td>0.2345</td>
<td>0.1274</td>
</tr>
<tr>
<td>CPPI5</td>
<td>0.0738</td>
<td>0.1431</td>
<td>0.1450</td>
<td>0.2109</td>
<td>0.1689</td>
<td>0.2371</td>
<td>0.1797</td>
</tr>
<tr>
<td>OBPI</td>
<td>0.0521</td>
<td>0.1436</td>
<td>0.1223</td>
<td>0.2302</td>
<td>0.1469</td>
<td>0.2583</td>
<td>0.1388</td>
</tr>
<tr>
<td>VaRPI</td>
<td>0.0412</td>
<td>0.0815</td>
<td>0.1736</td>
<td>0.1849</td>
<td>0.3431</td>
<td>0.2466</td>
<td>-</td>
</tr>
<tr>
<td>SLPI</td>
<td>0.0972</td>
<td>0.1740</td>
<td>0.1582</td>
<td>0.2351</td>
<td>0.1749</td>
<td>0.2563</td>
<td>0.2125</td>
</tr>
</tbody>
</table>

Table A. 2. The comparison of the AStd of the strategies across market scenarios

<table>
<thead>
<tr>
<th>Strat.</th>
<th>MC1</th>
<th>MC2</th>
<th>MC3</th>
<th>MC4</th>
<th>MC5</th>
<th>MC6</th>
<th>BT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>0.0626</td>
<td>0.1070</td>
<td>0.0174</td>
<td>0.0562</td>
<td>0.0244</td>
<td>0.0560</td>
<td>0.0047</td>
</tr>
<tr>
<td>CPPI3</td>
<td>0.5292</td>
<td>0.9643</td>
<td>0.1107</td>
<td>0.4910</td>
<td>0.0616</td>
<td>0.3673</td>
<td>0.0123</td>
</tr>
<tr>
<td>CPPI5</td>
<td>1.0734</td>
<td>1.7261</td>
<td>0.2544</td>
<td>1.1345</td>
<td>0.1066</td>
<td>0.9516</td>
<td>0.0047</td>
</tr>
<tr>
<td>OBPI</td>
<td>0.5643</td>
<td>0.3720</td>
<td>0.1229</td>
<td>0.1532</td>
<td>0.0708</td>
<td>0.1138</td>
<td>-0.0130</td>
</tr>
<tr>
<td>VaRPI</td>
<td>0.1546</td>
<td>0.2218</td>
<td>0.4475</td>
<td>0.3874</td>
<td>0.4651</td>
<td>0.4370</td>
<td>-</td>
</tr>
<tr>
<td>SLPI</td>
<td>1.0427</td>
<td>1.6510</td>
<td>0.3454</td>
<td>1.1511</td>
<td>0.2037</td>
<td>1.0064</td>
<td>0.0229</td>
</tr>
</tbody>
</table>

Table A. 3. The comparison of the AKurt of the strategies across market scenarios
### ASkew

<table>
<thead>
<tr>
<th>Strat.</th>
<th>MC1</th>
<th>MC2</th>
<th>MC3</th>
<th>MC4</th>
<th>MC5</th>
<th>MC6</th>
<th>BT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>-0.0405</td>
<td>-0.0435</td>
<td>0.0036</td>
<td>0.0026</td>
<td>0.0113</td>
<td>0.0121</td>
<td>0.0845</td>
</tr>
<tr>
<td>CPPI3</td>
<td>-0.3778</td>
<td>-0.5461</td>
<td>-0.0848</td>
<td>-0.2792</td>
<td>-0.0362</td>
<td>-0.2051</td>
<td>-0.0293</td>
</tr>
<tr>
<td>CPPI5</td>
<td>-0.6567</td>
<td>-0.8665</td>
<td>-0.1637</td>
<td>-0.5669</td>
<td>-0.0688</td>
<td>-0.4745</td>
<td>-0.0064</td>
</tr>
<tr>
<td>OBPI</td>
<td>-0.4185</td>
<td>-0.2493</td>
<td>-0.0779</td>
<td>-0.0899</td>
<td>-0.0208</td>
<td>-0.0548</td>
<td>0.0068</td>
</tr>
<tr>
<td>VaRPI</td>
<td>-0.0902</td>
<td>-0.0861</td>
<td>-0.1404</td>
<td>-0.1023</td>
<td>-0.1947</td>
<td>-0.1156</td>
<td>-</td>
</tr>
<tr>
<td>SLPI</td>
<td>-0.6670</td>
<td>-0.8391</td>
<td>-0.2130</td>
<td>-0.5746</td>
<td>-0.1262</td>
<td>-0.5043</td>
<td>0.1094</td>
</tr>
</tbody>
</table>

Table A. 4. The comparison of the ASkew of the strategies across market scenarios

### Sharpe ratio

<table>
<thead>
<tr>
<th>Strat.</th>
<th>MC1</th>
<th>MC2</th>
<th>MC3</th>
<th>MC4</th>
<th>MC5</th>
<th>MC6</th>
<th>BT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>-0.8504</td>
<td>-0.3018</td>
<td>0.2697</td>
<td>0.2354</td>
<td>0.6116</td>
<td>0.3985</td>
<td>0.3656</td>
</tr>
<tr>
<td>CPPI3</td>
<td>-0.6837</td>
<td>-0.2773</td>
<td>0.2109</td>
<td>0.1248</td>
<td>0.5493</td>
<td>0.2823</td>
<td>0.1729</td>
</tr>
<tr>
<td>CPPI5</td>
<td>-0.5556</td>
<td>-0.2523</td>
<td>0.1924</td>
<td>0.0896</td>
<td>0.5370</td>
<td>0.2400</td>
<td>0.1731</td>
</tr>
<tr>
<td>OBPI</td>
<td>-0.6257</td>
<td>-0.3224</td>
<td>0.2077</td>
<td>0.1429</td>
<td>0.5459</td>
<td>0.3043</td>
<td>0.2923</td>
</tr>
<tr>
<td>VaRPI</td>
<td>-0.8252</td>
<td>-0.3067</td>
<td>0.1210</td>
<td>0.1179</td>
<td>0.3127</td>
<td>0.2194</td>
<td>-</td>
</tr>
<tr>
<td>SLPI</td>
<td>-0.4969</td>
<td>-0.2638</td>
<td>0.1719</td>
<td>0.0761</td>
<td>0.5180</td>
<td>0.2259</td>
<td>0.3595</td>
</tr>
</tbody>
</table>

Table A. 5. The comparison of the Sharpe ratio of the strategies across market scenarios

### Sortino ratio

<table>
<thead>
<tr>
<th>Strat.</th>
<th>MC1</th>
<th>MC2</th>
<th>MC3</th>
<th>MC4</th>
<th>MC5</th>
<th>MC6</th>
<th>BT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>-0.5710</td>
<td>-0.4290</td>
<td>0.3024</td>
<td>0.4441</td>
<td>0.7950</td>
<td>0.8640</td>
<td>0.4208</td>
</tr>
<tr>
<td>CPPI3</td>
<td>-0.8740</td>
<td>-0.7048</td>
<td>0.4766</td>
<td>0.1682</td>
<td>1.6119</td>
<td>0.7326</td>
<td>0.4366</td>
</tr>
<tr>
<td>CPPI5</td>
<td>-0.8853</td>
<td>-0.7131</td>
<td>0.4747</td>
<td>0.0269</td>
<td>1.7022</td>
<td>0.5208</td>
<td>0.5245</td>
</tr>
<tr>
<td>OBPI</td>
<td>-0.7497</td>
<td>-0.8770</td>
<td>0.4853</td>
<td>0.3357</td>
<td>1.6183</td>
<td>0.9849</td>
<td>0.7919</td>
</tr>
<tr>
<td>VaRPI</td>
<td>-0.7969</td>
<td>-0.5592</td>
<td>0.2929</td>
<td>0.2562</td>
<td>1.0673</td>
<td>0.6543</td>
<td>-</td>
</tr>
<tr>
<td>SLPI</td>
<td>-0.9676</td>
<td>-0.8133</td>
<td>0.4558</td>
<td>0.0041</td>
<td>1.6693</td>
<td>0.5059</td>
<td>1.2463</td>
</tr>
</tbody>
</table>

Table A. 6. The comparison of the Sortino ratio of the strategies across market scenarios
### Table A. 7. The comparison of the Omega ratio of the strategies across market scenarios

<table>
<thead>
<tr>
<th>Strat.</th>
<th>MC1</th>
<th>MC2</th>
<th>MC3</th>
<th>MC4</th>
<th>MC5</th>
<th>MC6</th>
<th>BT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>0.4188</td>
<td>0.5593</td>
<td>1.3332</td>
<td>1.6278</td>
<td>1.8816</td>
<td>2.1949</td>
<td>1.4461</td>
</tr>
<tr>
<td>CPP13</td>
<td>0.0672</td>
<td>0.2015</td>
<td>1.7536</td>
<td>1.4721</td>
<td>3.3885</td>
<td>2.3673</td>
<td>1.5558</td>
</tr>
<tr>
<td>CPP15</td>
<td>0.0251</td>
<td>0.1569</td>
<td>1.7988</td>
<td>1.2829</td>
<td>3.5956</td>
<td>2.0841</td>
<td>1.7289</td>
</tr>
<tr>
<td>OBPI</td>
<td>0.2038</td>
<td>-0.0240</td>
<td>1.7553</td>
<td>1.7082</td>
<td>3.3637</td>
<td>2.7260</td>
<td>2.0337</td>
</tr>
<tr>
<td>VaRP1</td>
<td>0.1721</td>
<td>0.4039</td>
<td>1.5462</td>
<td>1.5172</td>
<td>3.0607</td>
<td>2.1926</td>
<td>-</td>
</tr>
<tr>
<td>SLPI</td>
<td>-0.1145</td>
<td>-0.0183</td>
<td>1.7995</td>
<td>1.2512</td>
<td>3.5954</td>
<td>2.0755</td>
<td>2.7644</td>
</tr>
</tbody>
</table>

### Table A. 8. The comparison of the Floor Protection ratio of the strategies across market scenarios

<table>
<thead>
<tr>
<th>Strat.</th>
<th>MC1</th>
<th>MC2</th>
<th>MC3</th>
<th>MC4</th>
<th>MC5</th>
<th>MC6</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>CPP13</td>
<td>0.6684</td>
<td>0.3007</td>
<td>0.9985</td>
<td>0.7602</td>
<td>1.0000</td>
<td>0.8581</td>
</tr>
<tr>
<td>CPP15</td>
<td>0.1153</td>
<td>0.0678</td>
<td>0.9026</td>
<td>0.3240</td>
<td>0.9815</td>
<td>0.4406</td>
</tr>
<tr>
<td>OBPI</td>
<td>0.9970</td>
<td>0.1264</td>
<td>0.9931</td>
<td>0.5539</td>
<td>0.9972</td>
<td>0.7004</td>
</tr>
<tr>
<td>VaRP1</td>
<td>0.9510</td>
<td>0.9523</td>
<td>0.9418</td>
<td>0.9491</td>
<td>0.9360</td>
<td>0.9463</td>
</tr>
<tr>
<td>SLPI</td>
<td>0.0193</td>
<td>0.0631</td>
<td>0.5518</td>
<td>0.2995</td>
<td>0.7604</td>
<td>0.4031</td>
</tr>
</tbody>
</table>

### Table A. 9. The comparison of the rank of the strategies across market scenarios and the Overall rank

<table>
<thead>
<tr>
<th>Strat.</th>
<th>MC1</th>
<th>MC2</th>
<th>MC3</th>
<th>MC4</th>
<th>MC5</th>
<th>MC6</th>
<th>Overall rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>BH</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CPP13</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>CPP15</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>OBPI</td>
<td>2</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>VaRP1</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>3</td>
<td>6</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>SLPI</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>
APPENDIX C

# PYTHON CODES FOR THESIS OF TRIEU TIEN DAT
# PORTFOLIO INSURANCE STRATEGIES: An empirical analysis of a frontier market - the Vietnamese case
#
# ***********************************************************************************************
# IMPORTING PACKAGES AND DATA
#
# Packages import
import numpy as np
import numpy.random as nrand
import pandas as pd
import scipy.stats as st
import plotly
plotly.too ls.set_credentials_file(username='myusername', api_key='myapikey')
import plotly.plotly as py
import plotly.figure_factory as ff
import plotly.graph_objs as go

##########
# Data import from Excel file and some basic calculations for vneq30
file = "Data for Thesis.xlsx"
data = pd.ExcelFile(file)
vneq30_monthly = data.parse(2)
bt_path_returns = np.array(vneq30_monthly)[1:,2]

# Some parameters for the whole Thesis
riskfree = 4.14/100 # risk-free rate is fixed
T = 5 # investment horizon (5 years ahead)
number_of_paths = 10000
V0 = 100000000 # initial value of insured portfolio
floor = 1000000000 # the floor at maturity
S = vneq30_monthly["Index"][0] # starts at 03 April 2012
K = S*np.exp(riskfree*T)
sigma_bt = 0.214329462

# Monte Carlo simulated data of vneq30 with predetermined parameters for 6 market scenarios
# The model parameters class
class monte_carlo_parameters:
    def __init__(self, mu, sigma, time_length, time_steps, paths):
        # the expected return of the Monte Carlo simulation
        self.mu = mu
        # the standard deviation of the Monte Carlo simulation
        self.sigma = sigma
        # investment period (number of years)
        self.time_length = time_length
        # numbers of rebalancing times in the whole investment period
        self.time_steps = time_steps
        # number of paths for Monte Carlo simulation
        self.paths = paths

    # Set parameters for 6 market scenarios mc1, mc2, mc3, mc4, mc5, mc6.
    mc1 = monte_carlo_parameters(mu=-0.15, sigma=0.2, time_length=5, time_steps=60, paths=number_of_paths)
    mc2 = monte_carlo_parameters(mu=-0.15, sigma=0.4, time_length=5, time_steps=60, paths=number_of_paths)
    mc3 = monte_carlo_parameters(mu=0.08, sigma=0.2, time_length=5, time_steps=60, paths=number_of_paths)
    mc4 = monte_carlo_parameters(mu=0.08, sigma=0.4, time_length=5, time_steps=60, paths=number_of_paths)
    mc5 = monte_carlo_parameters(mu=0.15, sigma=0.2, time_length=5, time_steps=60, paths=number_of_paths)
    mc6 = monte_carlo_parameters(mu=0.15, sigma=0.4, time_length=5, time_steps=60, paths=number_of_paths)

    # Function to create an array of Monte Carlo simulated paths of **returns** in the whole investment 5-year period
    def mc_array_path_returns(param):
        dt = param.time_length/param.time_steps
        mc_array_path_returns = np.zeros((param.time_steps,param.paths))
        for i in range(param.paths):
            nrand.seed(i)
            for j in range(param.time_steps):
                mc_array_path_returns[j,i] = param.mu*dt + param.sigma*np.sqrt(dt)*nrand.normal(0,1)  # p. 309, formula 14.7 Hull 9
        return mc_array_path_returns

    # Function for the Monte Carlo simulation of the BH (CPPI with m =1) and CPPI strategies (with m=3, m=5).
    def cppi_mc(floor, m, risk_free_local,T_local,mc_array_path_returns_local):
        # create an array of simulated data with 61 rows of steps and 10000 columns of paths
        value_of_insured_portfolio = np.zeros(((len(mc_array_path_returns_local[:,0])+1),len(mc_array_path_returns_local[0,:])))
        dt = T_local/len(mc_array_path_returns_local [:,0])
for i in range(len(mc_array_path_returns_local[0,:])): # run for each path, starting from path 0 to path 9999
    value_of_insured_portfolio[0,i] = V0 # set the initial value of the insured portfolio for all the paths (100,000,000 VND)
    pv_floor = floor*np.exp(-risk_free_local*T_local) # check if the formula for riskfree rate is correct

for j in range(len(mc_array_path_returns_local[:,0])): # run for each month, starting from month 1 to month 60
    if value_of_insured_portfolio[j,i] > pv_floor: # when the insured portfolio doesn't pierce through the floor (see equation (1.3) in Thesis)
        cushion = value_of_insured_portfolio[j,i] - pv_floor
        exposure_to_vn30eq = cushion*m
        rf_asset = value_of_insured_portfolio[j,i] - exposure_to_vn30eq
        if exposure_to_vn30eq > value_of_insured_portfolio[j,i]: # in strong bull market, exceed the limit (see equation 1.2)
            value_of_insured_portfolio[j+1,i] = value_of_insured_portfolio[j,i]*np.exp(mc_array_path_returns_local[j,i])
        else: # when vice versa, exposure to risky assets is less than the money we have (see equation 1.2)
            value_of_insured_portfolio[j+1,i] = exposure_to_vn30eq*np.exp(mc_array_path_returns_local[j,i])+rf_asset*np.exp(risk_free_local*dt)
    else: # when the insured portfolio pierces through the floor, all the money we have to be invested in risk-free assets (see equation (1.3))
        value_of_insured_portfolio[j+1,i] = value_of_insured_portfolio[j,i] * np.exp(risk_free_local*(T_local-j*dt)) # the floor value grows up at risk-free rate after each month

return value_of_insured_portfolio

# Function for the backtesting of the BH (CPPI with m =1) and CPPI strategies (with m=3, m=5).
def cppi_bt(floor, m, risk_free_local,T_local,bt_path_returns_local):
    # create a list of zeros then populate it with prices of 61 months (month 0 with V0 = 100000000 and those of the next 60 months)
    value_of_insured_portfolio = np.zeros(((len(bt_path_returns_local)+1),1))
    dt = T_local/len(value_of_insured_portfolio)

    value_of_insured_portfolio[0] = V0 # set the initial value of the insured portfolio for all the paths (100,000,000 VND)
    pv_floor = floor*np.exp(-risk_free_local*T_local) # check if the formula for riskfree rate is correct

    for j in range(len(bt_path_returns_local)): # run for each month, starting from month 1 to month 60
        if value_of_insured_portfolio[j] > pv_floor: # when the insured portfolio doesn't pierce through the floor (see equation (1.3))
            cushion = value_of_insured_portfolio[j] - pv_floor
            exposure_to_vn30eq = cushion*m
            rf_asset = value_of_insured_portfolio[j] - exposure_to_vn30eq

            if exposure_to_vn30eq > value_of_insured_portfolio[j]: # in strong bull market, exceed the limit (see equation 1.2)
                value_of_insured_portfolio[j+1] = value_of_insured_portfolio[j]*np.exp(bt_path_returns_local[j])
            else: # when vice versa, exposure to risky assets is less than the money we have (see equation 1.2)
                value_of_insured_portfolio[j+1] = exposure_to_vn30eq*np.exp(bt_path_returns_local[j])+rf_asset*np.exp(risk_free_local*dt)
        else:

    return value_of_insured_portfolio
(see equation (1.3))

\[ \text{value of insured portfolio}[j + 1] = \text{value of insured portfolio}[j] \times \exp(\text{risk free local} \times dt) \]

\[ \text{pv floor} = \text{floor} \times \exp(-\text{risk free local} \times (T_{local} - j \times dt)) \]

# the floor value grows up at risk-free rate after each month

return value of insured portfolio

# Function for the Monte Carlo simulation of the OBPI

def obpi_mc(K_local, S_local, T_local, sigma_local, risk_free_local, mc_array_path_returns_local):

    # create an array of simulated data with 61 rows and 10000 columns of paths
    value_of_insured_portfolio = np.zeros((len(mc_array_path_returns_local[:, 0]) + 1, len(mc_array_path_returns_local[0, :])))
    dt = T_local / len(mc_array_path_returns_local[:, 0])

    for i in range(len(mc_array_path_returns_local[0, :])):  # run for each path, starting from path 0 to path 9999
        value_of_insured_portfolio[0, i] = V0  # set the initial value of the insured portfolio for all the paths (100,000,000 VND)
        Index = S_local
        for j in range(len(mc_array_path_returns_local[:, 0])):  # run for each month
            d1 = (np.log(Index/K_local)+(risk_free_local+np.power(sigma_local, 2)/2)*(T_local-j*dt))/(sigma_local*np.sqrt(T_local-j*dt))
            d2 = d1 - sigma_local*np.sqrt(T_local-j*dt)
            N1 = st.norm.cdf(d1, 0, 1)
            N2 = st.norm.cdf(-d2, 0, 1)
            wRisky = (Index*N1)/(Index*N1+K_local*np.exp(-risk_free_local*(T_local-j*dt))*N2)
            wRiskfree = 1 - wRisky
            value_of_insured_portfolio[j+1, i] = wRisky*value_of_insured_portfolio[j, i]*np.exp(mc_array_path_returns_local[j, i]) + wRiskfree*value_of_insured_portfolio[j, i]*np.exp(risk_free_local*dt)
            Index = Index *np.exp(mc_array_path_returns_local[j, i])

    return value_of_insured_portfolio

# Function for the backtesting of the OBPI

def obpi_bt(K_local, S_local, T_local, sigma_local, risk_free_local, bt_path_returns_local):

    # create an array of simulated data with 61 rows and 10000 columns of paths
    value_of_insured_portfolio = np.zeros((len(bt_path_returns_local)+1, 1))
    dt = T_local / len(bt_path_returns_local)

    for j in range(len(bt_path_returns_local)):  # set the initial value of the insured portfolio for all the paths (100,000,000 VND)
        Index = S_local
        for i in range(len(bt_path_returns_local[:, 0])):  # run for each month
            d1 = (np.log(Index/K_local)+(risk_free_local+np.power(sigma_local, 2)/2)*(T_local-j*dt))/(sigma_local*np.sqrt(T_local-j*dt))
            d2 = d1 - sigma_local*np.sqrt(T_local-j*dt)
            N1 = st.norm.cdf(d1, 0, 1)
            N2 = st.norm.cdf(-d2, 0, 1)
            wRisky = (Index*N1)/(Index*N1+K_local*np.exp(-risk_free_local*(T_local-j*dt))*N2)
            wRiskfree = 1 - wRisky
            value_of_insured_portfolio[j+1, i] = wRisky*value_of_insured_portfolio[j, i]*np.exp(bt_path_returns_local[j, i]) + wRiskfree*value_of_insured_portfolio[j, i]*np.exp(risk_free_local*dt)
            Index = Index *np.exp(bt_path_returns_local[j, i])

    return value_of_insured_portfolio
N1 = st.norm.cdf(d1, 0, 1)
N2 = st.norm.cdf(-d2, 0, 1)

wRisky = (Index*N1)/(Index*N1+K_local*np.exp(-risk_free_local*(T_local-j*dt))*N2)
wRiskfree = 1 - wRisky

value_of_insured_portfolio[j+1]= wRisky*value_of_insured_portfolio[j]*np.exp(bt_path_returns_local[j]) \
  + wRiskfree*value_of_insured_portfolio[j]*np.exp(risk_free_local*dt)

Index = Index *np.exp(bt_path_returns_local[j])

# Function for the Monte Carlo simulation of the VaRPI
def varpi_mc(alpha, floor, risk_free_local, T_local, mc_array_path_returns_local, param):
    #alpha is confidence level
    # create an array of simulated data with 61 rows and 10000 columns of paths
    value_of_insured_portfolio = np.zeros(((len(mc_array_path_returns_local[:,0]) + 1), len(mc_array_path_returns_local[0,:])))
dt = T_local/len(mc_array_path_returns_local[:,0])

    for i in range(len(mc_array_path_returns_local[0,:])): # run for each path, starting from path 0 to path 9999
        #set the initial value of the insured portfolio for all the paths (100,000,000 VND)
        value_of_insured_portfolio[0,i]= V0

        for j in range(len(mc_array_path_returns_local[:,0])): # run for each month
            wRiskfree = (floor - value_of_insured_portfolio[j,i]*np.exp((param.mu-(1/2)*np.power(param.sigma,2))*(T_local-j*dt))-
                         st.norm.pdf(alpha)*param.sigma*st.norm.pdf(T_local-j*dt))/
                         (value_of_insured_portfolio[j,i]*np.exp(risk_free_local*(T_local-j*dt))-value_of_insured_portfolio[j,i]*np.exp((param.mu-(1/2)*np.power(param.sigma,2))*(T_local-j*dt))-
                         st.norm.pdf(alpha)*param.sigma*st.norm.pdf(T_local-j*dt))

            wRisky = 1 - wRiskfree

            value_of_insured_portfolio[j+1,i]= wRisky*value_of_insured_portfolio[j,i]*np.exp(mc_array_path_returns_local[j,i]) \
  + wRiskfree*value_of_insured_portfolio[j,i]*np.exp(risk_free_local*dt)

    return value_of_insured_portfolio

# Function for the Monte Carlo simulation of SLPI
def slpi_mc(floor, risk_free_local, T_local, mc_array_path_returns_local):
    # create an array of simulated data with 61 rows of steps and 10000 columns of paths
    value_of_insured_portfolio = np.zeros(((len(mc_array_path_returns_local[:,0]) + 1), len(mc_array_path_returns_local[0,:])))
dt = T_local/len(mc_array_path_returns_local[:,0])

    for i in range(len(mc_array_path_returns_local[0,:])): # run for each path, starting from path 0 to path 9999
        #set the initial value of the insured portfolio for all the paths (100,000,000 VND)
        value_of_insured_portfolio[0,i]= V0

        for j in range(len(mc_array_path_returns_local[:,0])): # run for each month (60 months)
            pv_floor = floor*np.exp(-risk_free_local*T_local) # check if the formula for riskfree rate is correct
            if value_of_insured_portfolio[j,i] > pv_floor: #when the insured portfolio doesn't pierce through the floor (see p. 24
value_of_insured_portfolio[j + 1, i] = value_of_insured_portfolio[j, i] * np.exp(mc_array_path_returns_local[j, i])

else:
    # when the insured portfolio pierces through the floor, all the money we have to be invested in risk-free assets
    value_of_insured_portfolio[j + 1, i] = value_of_insured_portfolio[j, i] * np.exp(risk_free_local * dt)

pv_floor = floor*np.exp(-risk_free_local*(T_local-j*dt))

# the floor value grows up at risk-free rate after each month
return value_of_insured_portfolio

# Function for the backtesting of SLPI

def slpi_bt(floor, risk_free_local,T_local, bt_path_returns_local):
    value_of_insured_portfolio = np.zeros(((len(bt_path_returns_local) + 1), 1)))
    dt = T_local/len(bt_path_returns_local)
    value_of_insured_portfolio[0] = V0  # set the initial value of the insured portfolio for all the paths (100,000,000 VND)
    pv_floor = floor*np.exp(-risk_free_local*T_local)
    for j in range(len(bt_path_returns_local)):  # run for each month, starting from month 1 to month 60
        if value_of_insured_portfolio[j] > pv_floor:  #when the insured portfolio doesn't pierce through the floor (see p. 24 Thesis)
        else:  # when the insured portfolio pierces through the floor, all the money we have to be invested in risk-free assets
            pv_floor = floor*np.exp(-risk_free_local*(T_local-j*dt))
        return value_of_insured_portfolio

# Function to convert the array of value of the insured portfolio to that of its returns with the Monte Carlo simulation

def convert_value_to_returns (array_value):
    array_value_to_returns = np.zeros(((len(array_value[:,0])-1),len(array_value[0,:])))
    for i in range(len(array_value[0,:])):  # run for each path
        for j in range(len(array_value[:,0])-1):  # run for each month
            array_value_to_returns[j,i] = np.log(array_value[j+1,i]/array_value[j,i])
    return array_value_to_returns

# 1.1. Buy and hold strategy (BH)
# a. Monter Carlo simulation method

array of value paths#floor= 100,000,000 VND #riskfree= 0.0414 #T=5
bh_mc1 = cppi_mc(floor,1,riskfree,T,mc_array_path_returns(mc1))
bh_mc2 = cppi_mc(floor,1,riskfree,T,mc_array_path_returns(mc2))
bh_mc3 = cppi_mc(floor,1,riskfree,T,mc_array_path_returns(mc3))
bh_mc4 = cppi_mc(floor,1,riskfree,T,mc_array_path_returns(mc4))
bh_mc5 = cppi_mc(floor,1,riskfree,T,mc_array_path_returns(mc5))
bh_mc6 = cppi_mc(floor,1,riskfree,T,mc_array_path_returns(mc6))
#array of returns paths
returns_bh_mc1 = convert_value_to_returns(bh_mc1)
returns_bh_mc2 = convert_value_to_returns(bh_mc2)
returns_bh_mc3 = convert_value_to_returns(bh_mc3)
returns_bh_mc4 = convert_value_to_returns(bh_mc4)
returns_bh_mc5 = convert_value_to_returns(bh_mc5)
returns_bh_mc6 = convert_value_to_returns(bh_mc6)

# b. Backtesting
bh_bt = cppi_bt(floor,1,riskfree,T,bt_path_returns)
returns_bh_bt = convert_value_to_returns(bh_bt)

# 1.2. Constant Proportion portfolio insurance strategy (CPPI)
# 1.2.1. m = 3
# a. Monte Carlo simulation method
#array of value paths
floor= 100,000,000 VND #riskfree= 0.0414 #T=5
cppi3_mc1 = cppi_mc(floor,3,riskfree,T,mc_array_path_returns(mc1))
cppi3_mc2 = cppi_mc(floor,3,riskfree,T,mc_array_path_returns(mc2))
cppi3_mc3 = cppi_mc(floor,3,riskfree,T,mc_array_path_returns(mc3))
cppi3_mc4 = cppi_mc(floor,3,riskfree,T,mc_array_path_returns(mc4))
cppi3_mc5 = cppi_mc(floor,3,riskfree,T,mc_array_path_returns(mc5))
cppi3_mc6 = cppi_mc(floor,3,riskfree,T,mc_array_path_returns(mc6))

#array of returns paths
returns_cppi3_mc1 = convert_value_to_returns(cppi3_mc1)
returns_cppi3_mc2 = convert_value_to_returns(cppi3_mc2)
returns_cppi3_mc3 = convert_value_to_returns(cppi3_mc3)
returns_cppi3_mc4 = convert_value_to_returns(cppi3_mc4)
returns_cppi3_mc5 = convert_value_to_returns(cppi3_mc5)
returns_cppi3_mc6 = convert_value_to_returns(cppi3_mc6)

# b. Backtesting
cppi3_bt = cppi_bt(floor,3,riskfree,T,bt_path_returns)
returns_cppi3_bt = convert_value_to_returns(cppi3_bt)

# 1.2.2. m = 5
# a. Monte Carlo simulation method
#array of value paths
floor= 100,000,000 VND #riskfree= 0.0414 #T=5
cppi5_mc1 = cppi_mc(floor,5,riskfree,T,mc_array_path_returns(mc1))
cppi5_mc2 = cppi_mc(floor,5,riskfree,T,mc_array_path_returns(mc2))
cppi5_mc3 = cppi_mc(floor,5,riskfree,T,mc_array_path_returns(mc3))
cppi5_mc4 = cppi_mc(floor,5,riskfree,T,mc_array_path_returns(mc4))
cppi5_mc5 = cppi_mc(floor, 5, riskfree, T, mc_array_path_returns(mc5))
cppi5_mc6 = cppi_mc(floor, 5, riskfree, T, mc_array_path_returns(mc6))

# array of returns paths
returns_cppi5_mc1 = convert_value_to_returns(cppi5_mc1)
returns_cppi5_mc2 = convert_value_to_returns(cppi5_mc2)
returns_cppi5_mc3 = convert_value_to_returns(cppi5_mc3)
returns_cppi5_mc4 = convert_value_to_returns(cppi5_mc4)
returns_cppi5_mc5 = convert_value_to_returns(cppi5_mc5)
returns_cppi5_mc6 = convert_value_to_returns(cppi5_mc6)

# b. Backtesting
cppi5_bt = cppi_bt(floor, 5, riskfree, T, bt_path_returns)
returns_cppi5_bt = convert_value_to_returns(cppi5_bt)

# 1.3. Option-based portfolio insurance strategy (OBPI)
# a. Monte Carlo simulation method
# array of value paths
K = S * np.exp(riskfree * T), # S = 11245673.329, # riskfree = 0.0414 # T = 5
obpi_mc1 = obpi_mc(K, S, T, mc1.sigma, riskfree, mc_array_path_returns(mc1))
obpi_mc2 = obpi_mc(K, S, T, mc2.sigma, riskfree, mc_array_path_returns(mc2))
obpi_mc3 = obpi_mc(K, S, T, mc3.sigma, riskfree, mc_array_path_returns(mc3))
obpi_mc4 = obpi_mc(K, S, T, mc4.sigma, riskfree, mc_array_path_returns(mc4))
obpi_mc5 = obpi_mc(K, S, T, mc5.sigma, riskfree, mc_array_path_returns(mc5))
obpi_mc6 = obpi_mc(K, S, T, mc6.sigma, riskfree, mc_array_path_returns(mc6))

# array of returns paths
returns_obpi_mc1 = convert_value_to_returns(obpi_mc1)
returns_obpi_mc2 = convert_value_to_returns(obpi_mc2)
returns_obpi_mc3 = convert_value_to_returns(obpi_mc3)
returns_obpi_mc4 = convert_value_to_returns(obpi_mc4)
returns_obpi_mc5 = convert_value_to_returns(obpi_mc5)
returns_obpi_mc6 = convert_value_to_returns(obpi_mc6)

# b. Backtesting
obpi_bt = obpi_bt(K, S, T, sigma_bt, riskfree, bt_path_returns)
returns_obpi_bt = convert_value_to_returns(obpi_bt)

# 1.4. VaR-based portfolio insurance strategy (VaRPI)

# Monte Carlo simulation method
# array of value paths
alpha = 0.95, # riskfree = 0.0414 # T = 5
varpi_mc1 = varpi_mc(alpha, floor, riskfree, T, mc_array_path_returns(mc1), mc1)
varpi_mc2 = varpi_mc(alpha, floor, riskfree, T, mc_array_path_returns(mc2), mc2)
varpi_mc3 = varpi_mc(0.95, floor, riskfree, T, mc_array_path_returns(mc3), mc3)
varpi_mc4 = varpi_mc(0.95, floor, riskfree, T, mc_array_path_returns(mc4), mc4)
varpi_mc5 = varpi_mc(0.95, floor, riskfree, T, mc_array_path_returns(mc5), mc5)
varpi_mc6 = varpi_mc(0.95, floor, riskfree, T, mc_array_path_returns(mc6), mc6)

# array of returns paths
returns_varpi_mc1 = convert_value_to_returns(varpi_mc1)
returns_varpi_mc2 = convert_value_to_returns(varpi_mc2)
returns_varpi_mc3 = convert_value_to_returns(varpi_mc3)
returns_varpi_mc4 = convert_value_to_returns(varpi_mc4)
returns_varpi_mc5 = convert_value_to_returns(varpi_mc5)
returns_varpi_mc6 = convert_value_to_returns(varpi_mc6)

# 1.5. Stop-loss portfolio insurance strategy (SLPI)
# a. Monter Carlo simulation method
slpi_mc1 = slpi_mc(floor, riskfree, T, mc_array_path_returns(mc1))
slpi_mc2 = slpi_mc(floor, riskfree, T, mc_array_path_returns(mc2))
slpi_mc3 = slpi_mc(floor, riskfree, T, mc_array_path_returns(mc3))
slpi_mc4 = slpi_mc(floor, riskfree, T, mc_array_path_returns(mc4))
slpi_mc5 = slpi_mc(floor, riskfree, T, mc_array_path_returns(mc5))
slpi_mc6 = slpi_mc(floor, riskfree, T, mc_array_path_returns(mc6))

# array of returns paths
returns_slpi_mc1 = convert_value_to_returns(slpi_mc1)
returns_slpi_mc2 = convert_value_to_returns(slpi_mc2)
returns_slpi_mc3 = convert_value_to_returns(slpi_mc3)
returns_slpi_mc4 = convert_value_to_returns(slpi_mc4)
returns_slpi_mc5 = convert_value_to_returns(slpi_mc5)
returns_slpi_mc6 = convert_value_to_returns(slpi_mc6)

# b. Backtesting
slpi_bt = slpi_bt(floor, riskfree, T, bt_path_returns)
returns_slpi.bt = convert_value_to_returns(slpi_bt)

# Chapter 2: Portfolio Insurance Performance Measures
# Series of functions to calculate the performance measures
# Annualized Mean

def amean (returns):
    return np.nanmean(returns)*12
# Annualized Standard Deviation
def astd (returns):
    return np.nanstd(returns)*np.sqrt(12)
# Annualized Skewness

def askew (returns):
    return pd.Series(returns).skew()/np.sqrt(12) # see the proof for this formula in word file
# Annualized Kurtosis

def akurt(returns):
    return pd.Series(returns).kurtosis()/12 # see the proof for this formula in word file
# The Sharpe ratio
# Reference of codes http://www.turingfinance.com/computational-investing-with-python-week-one/
def sharpe(amean_local, riskfree_local, astd_local):
    return (amean_local - riskfree_local) / astd_local
# Definition of Lower partial moment of the returns
# Reference of codes http://www.turingfinance.com/computational-investing-with-python-week-one/
def lpm(returns, threshold, order):
    # Calculate the difference between the threshold and the returns
    diff = threshold - returns
    # Set the minimum of each to 0
    diff = diff.clip(min=0)
    # Return the sum of the different to the power of order
    return np.sum(diff ** order) / len(returns)
# The Sortino ratio
# Reference of codes http://www.turingfinance.com/computational-investing-with-python-week-one/
def sortino(amean_local, riskfree_local, returns_local):
    return (amean_local - riskfree_local) / np.sqrt(lpm(returns_local, riskfree_local, 2))
# The Omega ratio
# Reference of codes http://www.turingfinance.com/computational-investing-with-python-week-one/
def omega(amean_local, riskfree_local, returns_local):
    return (((amean_local - riskfree_local) / lpm(returns_local, riskfree_local, 1)) + 1)
# The Floor Protection ratio

def prot(array_value):
    list_of_last_value = []
    for i in range(len(array_value[0, :])):
        list_of_last_value.append(array_value[-1, i])
    return sum(1 for x in list_of_last_value if x>= floor)/len(array_value[0, :])

# Function to create the list of performance measures.
def function_list_perf_measure (array_returns,perf_measure):
    list_perf_measure = []
    if perf_measure == "mean":
        for i in range(len(array_returns[0,:])):
```python
list_perf_measure.append(amean(array_returns[:,i]))
elif perf_measure == "std":
    for i in range(len(array_returns[0,:])):
        list_perf_measure.append(astd(array_returns[:,i]))
elif perf_measure == "skew":
    for i in range(len(array_returns[0,:])):
        list_perf_measure.append(askew(array_returns[:,i]))
elif perf_measure == "kurt":
    for i in range(len(array_returns[0,:])):
        list_perf_measure.append(akurt(array_returns[:,i]))
elif perf_measure == "sharpe":
    for i in range(len(array_returns[0,:])):
        list_perf_measure.append(sharpe(amean(array_returns[:,i]),riskfree,astd(array_returns[:,i])))
elif perf_measure == "sortino":
    for i in range(len(array_returns[0,:])):
        list_perf_measure.append(sortino(amean(array_returns[:,i]),riskfree,array_returns[:,i]))
elif perf_measure == "omega":
    for i in range(len(array_returns[0,:])):
        list_perf_measure.append(omega(amean(array_returns[:,i]),riskfree,array_returns[:,i]))
else:
    print("The performance measure you choose is not available. Please read the comment in the first line of the function")
return list_perf_measure
```

Sample average of the performance measures with \( N = 10,000 \) paths

# 2.1. First four moments
# 2.1.1. Expected Returns
# a. Buy and hold strategy
amean_bh_mc1 = np.nanmean(function_list_perf_measure(returns_bh_mc1,"mean"))
name_bh_mc2 = np.nanmean(function_list_perf_measure(returns_bh_mc2,"mean"))
name_bh_mc3 = np.nanmean(function_list_perf_measure(returns_bh_mc3,"mean"))
name_bh_mc4 = np.nanmean(function_list_perf_measure(returns_bh_mc4,"mean"))
name_bh_mc5 = np.nanmean(function_list_perf_measure(returns_bh_mc5,"mean"))
name_bh_mc6 = np.nanmean(function_list_perf_measure(returns_bh_mc6,"mean"))

name_bh_bt = np.nanmean(function_list_perf_measure(returns_bh_bt,"mean"))
# b. Constant Proportion portfolio insurance strategy (cppi)
# b.1. m = 3
name_cppi3_mc1 = np.nanmean(function_list_perf_measure(returns_cppi3_mc1,"mean"))
name_cppi3_mc2 = np.nanmean(function_list_perf_measure(returns_cppi3_mc2,"mean"))
name_cppi3_mc3 = np.nanmean(function_list_perf_measure(returns_cppi3_mc3,"mean"))
name_cppi3_mc4 = np.nanmean(function_list_perf_measure(returns_cppi3_mc4,"mean"))
name_cppi3_mc5 = np.nanmean(function_list_perf_measure(returns_cppi3_mc5,"mean"))
```
amean_cppi3_mc6 = np.nanmean(function_list_perf_measure(returns_cppi3_mc6, "mean"))

amean_cppi3_bt = np.nanmean(function_list_perf_measure(returns_cppi3_bt, "mean"))

# b.1. m = 5
amean_cppi5_mc1 = np.nanmean(function_list_perf_measure(returns_cppi5_mc1, "mean"))
amean_cppi5_mc2 = np.nanmean(function_list_perf_measure(returns_cppi5_mc2, "mean"))
amean_cppi5_mc3 = np.nanmean(function_list_perf_measure(returns_cppi5_mc3, "mean"))
amean_cppi5_mc4 = np.nanmean(function_list_perf_measure(returns_cppi5_mc4, "mean"))
amean_cppi5_mc5 = np.nanmean(function_list_perf_measure(returns_cppi5_mc5, "mean"))
amean_cppi5_mc6 = np.nanmean(function_list_perf_measure(returns_cppi5_mc6, "mean"))

amean_cppi5_bt = np.nanmean(function_list_perf_measure(returns_cppi5_bt, "mean"))

# c. Option-based portfolio insurance strategy (OBPI)
amean_obpi_mc1 = np.nanmean(function_list_perf_measure(returns_obpi_mc1, "mean"))
amean_obpi_mc2 = np.nanmean(function_list_perf_measure(returns_obpi_mc2, "mean"))
amean_obpi_mc3 = np.nanmean(function_list_perf_measure(returns_obpi_mc3, "mean"))
amean_obpi_mc4 = np.nanmean(function_list_perf_measure(returns_obpi_mc4, "mean"))
amean_obpi_mc5 = np.nanmean(function_list_perf_measure(returns_obpi_mc5, "mean"))
amean_obpi_mc6 = np.nanmean(function_list_perf_measure(returns_obpi_mc6, "mean"))

amean_obpi_bt = np.nanmean(function_list_perf_measure(returns_obpi_bt, "mean"))

# d VaR-based portfolio insurance strategy (VaRPI)
amean_varpi_mc1 = np.nanmean(function_list_perf_measure(returns_varpi_mc1, "mean"))
amean_varpi_mc2 = np.nanmean(function_list_perf_measure(returns_varpi_mc2, "mean"))
amean_varpi_mc3 = np.nanmean(function_list_perf_measure(returns_varpi_mc3, "mean"))
amean_varpi_mc4 = np.nanmean(function_list_perf_measure(returns_varpi_mc4, "mean"))
amean_varpi_mc5 = np.nanmean(function_list_perf_measure(returns_varpi_mc5, "mean"))
amean_varpi_mc6 = np.nanmean(function_list_perf_measure(returns_varpi_mc6, "mean"))

# Not available for backtesting
# e. Stop-loss portfolio insurance strategy (SLPI)
amean_slpi_mc1 = np.nanmean(function_list_perf_measure(returns_slpi_mc1, "mean"))
amean_slpi_mc2 = np.nanmean(function_list_perf_measure(returns_slpi_mc2, "mean"))
amean_slpi_mc3 = np.nanmean(function_list_perf_measure(returns_slpi_mc3, "mean"))
amean_slpi_mc4 = np.nanmean(function_list_perf_measure(returns_slpi_mc4, "mean"))
amean_slpi_mc5 = np.nanmean(function_list_perf_measure(returns_slpi_mc5, "mean"))
amean_slpi_mc6 = np.nanmean(function_list_perf_measure(returns_slpi_mc6, "mean"))

amean_slpi_bt = np.nanmean(function_list_perf_measure(returns_slpi_bt, "mean"))

# 2.1.2. Standard Deviation
# a. Buy and hold strategy

```python
astd_bh_mc1 = np.nanmean(function_list_perf_measure(returns_bh_mc1, "std"))
astd_bh_mc2 = np.nanmean(function_list_perf_measure(returns_bh_mc2, "std"))
astd_bh_mc3 = np.nanmean(function_list_perf_measure(returns_bh_mc3, "std"))
astd_bh_mc4 = np.nanmean(function_list_perf_measure(returns_bh_mc4, "std"))
astd_bh_mc5 = np.nanmean(function_list_perf_measure(returns_bh_mc5, "std"))
astd_bh_mc6 = np.nanmean(function_list_perf_measure(returns_bh_mc6, "std"))
```

# Should think about the annualized later when we finish the basic daily one

# b. Constant Proportion portfolio insurance strategy (cppi)

## b.1. m = 3

```python
astd_cppi3_mc1 = np.nanmean(function_list_perf_measure(returns_cppi3_mc1, "std"))
astd_cppi3_mc2 = np.nanmean(function_list_perf_measure(returns_cppi3_mc2, "std"))
astd_cppi3_mc3 = np.nanmean(function_list_perf_measure(returns_cppi3_mc3, "std"))
astd_cppi3_mc4 = np.nanmean(function_list_perf_measure(returns_cppi3_mc4, "std"))
astd_cppi3_mc5 = np.nanmean(function_list_perf_measure(returns_cppi3_mc5, "std"))
astd_cppi3_mc6 = np.nanmean(function_list_perf_measure(returns_cppi3_mc6, "std"))
```

## b.2. m = 5

```python
astd_cppi5_mc1 = np.nanmean(function_list_perf_measure(returns_cppi5_mc1, "std"))
astd_cppi5_mc2 = np.nanmean(function_list_perf_measure(returns_cppi5_mc2, "std"))
astd_cppi5_mc3 = np.nanmean(function_list_perf_measure(returns_cppi5_mc3, "std"))
astd_cppi5_mc4 = np.nanmean(function_list_perf_measure(returns_cppi5_mc4, "std"))
astd_cppi5_mc5 = np.nanmean(function_list_perf_measure(returns_cppi5_mc5, "std"))
astd_cppi5_mc6 = np.nanmean(function_list_perf_measure(returns_cppi5_mc6, "std"))
```

# c. Option-based portfolio insurance strategy (OBPI)

```python
astd_obpi_mc1 = np.nanmean(function_list_perf_measure(returns_obpi_mc1, "std"))
astd_obpi_mc2 = np.nanmean(function_list_perf_measure(returns_obpi_mc2, "std"))
astd_obpi_mc3 = np.nanmean(function_list_perf_measure(returns_obpi_mc3, "std"))
astd_obpi_mc4 = np.nanmean(function_list_perf_measure(returns_obpi_mc4, "std"))
astd_obpi_mc5 = np.nanmean(function_list_perf_measure(returns_obpi_mc5, "std"))
astd_obpi_mc6 = np.nanmean(function_list_perf_measure(returns_obpi_mc6, "std"))
```

# d VaR-based portfolio insurance strategy (VaRPI)

```python
astd_varpi_mc1 = np.nanmean(function_list_perf_measure(returns_varpi_mc1, "std"))
astd_varpi_mc2 = np.nanmean(function_list_perf_measure(returns_varpi_mc2, "std"))
```
\begin{verbatim}
astd_varpi_mc3 = np.nanmean(function_list_perf_measure(returns_varpi_mc3,"std"))
astd_varpi_mc4 = np.nanmean(function_list_perf_measure(returns_varpi_mc4,"std"))
astd_varpi_mc5 = np.nanmean(function_list_perf_measure(returns_varpi_mc5,"std"))
astd_varpi_mc6 = np.nanmean(function_list_perf_measure(returns_varpi_mc6,"std"))

# e. Stop-loss portfolio insurance strategy (SLPI)
astd_slpi_mc1 = np.nanmean(function_list_perf_measure(returns_slpi_mc1,"std"))
astd_slpi_mc2 = np.nanmean(function_list_perf_measure(returns_slpi_mc2,"std"))
astd_slpi_mc3 = np.nanmean(function_list_perf_measure(returns_slpi_mc3,"std"))
astd_slpi_mc4 = np.nanmean(function_list_perf_measure(returns_slpi_mc4,"std"))
astd_slpi_mc5 = np.nanmean(function_list_perf_measure(returns_slpi_mc5,"std"))
astd_slpi_mc6 = np.nanmean(function_list_perf_measure(returns_slpi_mc6,"std"))

astd_slpi_bt = np.nanmean(function_list_perf_measure(returns_slpi_bt,"std"))

# 2.1.3. Skewness

# a. Buy and hold strategy
askew_bh_mc1 = np.nanmean(function_list_perf_measure(returns_bh_mc1,"skew"))
askew_bh_mc2 = np.nanmean(function_list_perf_measure(returns_bh_mc2,"skew"))
askew_bh_mc3 = np.nanmean(function_list_perf_measure(returns_bh_mc3,"skew"))
askew_bh_mc4 = np.nanmean(function_list_perf_measure(returns_bh_mc4,"skew"))
askew_bh_mc5 = np.nanmean(function_list_perf_measure(returns_bh_mc5,"skew"))
askew_bh_mc6 = np.nanmean(function_list_perf_measure(returns_bh_mc6,"skew"))

askew_bh_bt = np.nanmean(function_list_perf_measure(returns_bh_bt,"skew"))

# b. Constant Proportion portfolio insurance strategy (cppi)
  # b.1. m = 3
askew_cppi3_mc1 = np.nanmean(function_list_perf_measure(returns_cppi3_mc1,"skew"))
askew_cppi3_mc2 = np.nanmean(function_list_perf_measure(returns_cppi3_mc2,"skew"))
askew_cppi3_mc3 = np.nanmean(function_list_perf_measure(returns_cppi3_mc3,"skew"))
askew_cppi3_mc4 = np.nanmean(function_list_perf_measure(returns_cppi3_mc4,"skew"))
askew_cppi3_mc5 = np.nanmean(function_list_perf_measure(returns_cppi3_mc5,"skew"))
askew_cppi3_mc6 = np.nanmean(function_list_perf_measure(returns_cppi3_mc6,"skew"))

askew_cppi3_bt = np.nanmean(function_list_perf_measure(returns_cppi3_bt,"skew"))
  # b.2. m = 5
askew_cppi5_mc1 = np.nanmean(function_list_perf_measure(returns_cppi5_mc1,"skew"))
askew_cppi5_mc2 = np.nanmean(function_list_perf_measure(returns_cppi5_mc2,"skew"))
askew_cppi5_mc3 = np.nanmean(function_list_perf_measure(returns_cppi5_mc3,"skew"))
askew_cppi5_mc4 = np.nanmean(function_list_perf_measure(returns_cppi5_mc4,"skew"))
askew_cppi5_mc5 = np.nanmean(function_list_perf_measure(returns_cppi5_mc5,"skew"))
\end{verbatim}
askew_cppi5_mc6 = np.nanmean(function_list_perf_measure(returns_cppi5_mc6,"skew"))

askew_cppi5_bt = np.nanmean(function_list_perf_measure(returns_cppi5.bt,"skew"))
# c. Option-based portfolio insurance strategy (OBPI)
askew_obpi_mc1 = np.nanmean(function_list_perf_measure(returns_obpi_mc1,"skew"))
askew_obpi_mc2 = np.nanmean(function_list_perf_measure(returns_obpi_mc2,"skew"))
askew_obpi_mc3 = np.nanmean(function_list_perf_measure(returns_obpi_mc3,"skew"))
askew_obpi_mc4 = np.nanmean(function_list_perf_measure(returns_obpi_mc4,"skew"))
askew_obpi_mc5 = np.nanmean(function_list_perf_measure(returns_obpi_mc5,"skew"))
askew_obpi_mc6 = np.nanmean(function_list_perf_measure(returns_obpi_mc6,"skew"))

askew_obpi bt = np.nanmean(function_list_perf_measure(returns_obpi bt,"skew"))
# d. VaR-based portfolio insurance strategy (VaRPI)
askew_varpi mc1 = np.nanmean(function_list_perf_measure(returns_varpi mc1,"skew"))
askew_varpi mc2 = np.nanmean(function_list_perf_measure(returns_varpi mc2,"skew"))
askew_varpi mc3 = np.nanmean(function_list_perf_measure(returns_varpi mc3,"skew"))
askew_varpi mc4 = np.nanmean(function_list_perf_measure(returns_varpi mc4,"skew"))
askew_varpi mc5 = np.nanmean(function_list_perf_measure(returns_varpi mc5,"skew"))
askew_varpi mc6 = np.nanmean(function_list_perf_measure(returns_varpi mc6,"skew"))

# Not available for backtesting
# e. Stop-loss portfolio insurance strategy (SLPI)
askew_slpi mc1 = np.nanmean(function_list_perf_measure(returns_slpi mc1,"skew"))
askew_slpi mc2 = np.nanmean(function_list_perf_measure(returns_slpi mc2,"skew"))
askew_slpi mc3 = np.nanmean(function_list_perf_measure(returns_slpi mc3,"skew"))
askew_slpi mc4 = np.nanmean(function_list_perf_measure(returns_slpi mc4,"skew"))
askew_slpi mc5 = np.nanmean(function_list_perf_measure(returns_slpi mc5,"skew"))
askew_slpi mc6 = np.nanmean(function_list_perf_measure(returns_slpi mc6,"skew"))

askew_slpi bt = np.nanmean(function_list_perf_measure(returns_slpi bt,"skew"))
# 2.1.4. Kurtosis
# a. Buy and hold strategy
akurt_bh mc1 = np.nanmean(function_list_perf_measure(returns_bh mc1,"kurt"))
akurt_bh mc2 = np.nanmean(function_list_perf_measure(returns_bh mc2,"kurt"))
akurt_bh mc3 = np.nanmean(function_list_perf_measure(returns_bh mc3,"kurt"))
akurt_bh mc4 = np.nanmean(function_list_perf_measure(returns_bh mc4,"kurt"))
akurt_bh mc5 = np.nanmean(function_list_perf_measure(returns_bh mc5,"kurt"))
akurt_bh mc6 = np.nanmean(function_list_perf_measure(returns_bh mc6,"kurt"))

akurt_bh bt = np.nanmean(function_list_perf_measure(returns_bh bt,"kurt"))
# b. Constant Proportion portfolio insurance strategy (cppi)
# b.1. m = 3
akurt_cppi3_mc1 = np.nanmean(function_list_perf_measure(returns_cppi3_mc1,"kurt"))
akurt_cppi3_mc2 = np.nanmean(function_list_perf_measure(returns_cppi3_mc2,"kurt"))
akurt_cppi3_mc3 = np.nanmean(function_list_perf_measure(returns_cppi3_mc3,"kurt"))
akurt_cppi3_mc4 = np.nanmean(function_list_perf_measure(returns_cppi3_mc4,"kurt"))
akurt_cppi3_mc5 = np.nanmean(function_list_perf_measure(returns_cppi3_mc5,"kurt"))
akurt_cppi3_mc6 = np.nanmean(function_list_perf_measure(returns_cppi3_mc6,"kurt"))

akurt_cppi3_bt = np.nanmean(function_list_perf_measure(returns_cppi3_bt,"kurt"))
# b.2. m = 5
akurt_cppi5_mc1 = np.nanmean(function_list_perf_measure(returns_cppi5_mc1,"kurt"))
akurt_cppi5_mc2 = np.nanmean(function_list_perf_measure(returns_cppi5_mc2,"kurt"))
akurt_cppi5_mc3 = np.nanmean(function_list_perf_measure(returns_cppi5_mc3,"kurt"))
akurt_cppi5_mc4 = np.nanmean(function_list_perf_measure(returns_cppi5_mc4,"kurt"))
akurt_cppi5_mc5 = np.nanmean(function_list_perf_measure(returns_cppi5_mc5,"kurt"))
akurt_cppi5_mc6 = np.nanmean(function_list_perf_measure(returns_cppi5_mc6,"kurt"))

akurt_cppi5_bt = np.nanmean(function_list_perf_measure(returns_cppi5_bt,"kurt"))
# c. Option-based portfolio insurance strategy (OBPI)
akurt_obpi_mc1 = np.nanmean(function_list_perf_measure(returns_obpi_mc1,"kurt"))
akurt_obpi_mc2 = np.nanmean(function_list_perf_measure(returns_obpi_mc2,"kurt"))
akurt_obpi_mc3 = np.nanmean(function_list_perf_measure(returns_obpi_mc3,"kurt"))
akurt_obpi_mc4 = np.nanmean(function_list_perf_measure(returns_obpi_mc4,"kurt"))
akurt_obpi_mc5 = np.nanmean(function_list_perf_measure(returns_obpi_mc5,"kurt"))
akurt_obpi_mc6 = np.nanmean(function_list_perf_measure(returns_obpi_mc6,"kurt"))

akurt_obpi_bt = np.nanmean(function_list_perf_measure(returns_obpi_bt,"kurt"))
# d. VaR-based portfolio insurance strategy (VaRPI)
akurt_varpi_mc1 = np.nanmean(function_list_perf_measure(returns_varpi_mc1,"kurt"))
akurt_varpi_mc2 = np.nanmean(function_list_perf_measure(returns_varpi_mc2,"kurt"))
akurt_varpi_mc3 = np.nanmean(function_list_perf_measure(returns_varpi_mc3,"kurt"))
akurt_varpi_mc4 = np.nanmean(function_list_perf_measure(returns_varpi_mc4,"kurt"))
akurt_varpi_mc5 = np.nanmean(function_list_perf_measure(returns_varpi_mc5,"kurt"))
akurt_varpi_mc6 = np.nanmean(function_list_perf_measure(returns_varpi_mc6,"kurt"))

# Not available for backtesting
# e. Stop-loss portfolio insurance strategy (SLPI)
akurt_slpi_mc1 = np.nanmean(function_list_perf_measure(returns_slpi_mc1,"kurt"))
akurt_slpi_mc2 = np.nanmean(function_list_perf_measure(returns_slpi_mc2,"kurt"))
akurt_slpi_mc3 = np.nanmean(function_list_perf_measure(returns_slpi_mc3,"kurt"))
akurt_slpi_mc4 = np.nanmean(function_list_perf_measure(returns_slpi_mc4,"kurt"))
akurt_slpi_mc5 = np.nanmean(function_list_perf_measure(returns_slpi_mc5,"kurt"))
akurt_slpi_mc6 = np.nanmean(function_list_perf_measure(returns_slpi_mc6,"kurt"))

akurt_slpi_bt = np.nanmean(function_list_perf_measure(returns_slpi_bt,"kurt"))

# 2.2. Performance ratios
# 2.2.1. The Sharpe ratio #Be sure to use the VaRiable Riskfree consistently

# a. Buy and hold strategy
sharpe_bh_mc1 = np.nanmean(function_list_perf_measure(returns_bh_mc1,"sharpe"))
sharpe_bh_mc2 = np.nanmean(function_list_perf_measure(returns_bh_mc2,"sharpe"))
sharpe_bh_mc3 = np.nanmean(function_list_perf_measure(returns_bh_mc3,"sharpe"))
sharpe_bh_mc4 = np.nanmean(function_list_perf_measure(returns_bh_mc4,"sharpe"))
sharpe_bh_mc5 = np.nanmean(function_list_perf_measure(returns_bh_mc5,"sharpe"))
sharpe_bh_mc6 = np.nanmean(function_list_perf_measure(returns_bh_mc6,"sharpe"))

sharpe_bh_bt = np.nanmean(function_list_perf_measure(returns_bh_bt,"sharpe"))
#Sharpe ratio here should be constant or will be an array of rates

# b. Constant Proportion portfolio insurance strategy (cppi)
# b.1. m = 3
sharpe_cppi3_mc1 = np.nanmean(function_list_perf_measure(returns_cppi3_mc1,"sharpe"))
sharpe_cppi3_mc2 = np.nanmean(function_list_perf_measure(returns_cppi3_mc2,"sharpe"))
sharpe_cppi3_mc3 = np.nanmean(function_list_perf_measure(returns_cppi3_mc3,"sharpe"))
sharpe_cppi3_mc4 = np.nanmean(function_list_perf_measure(returns_cppi3_mc4,"sharpe"))
sharpe_cppi3_mc5 = np.nanmean(function_list_perf_measure(returns_cppi3_mc5,"sharpe"))
sharpe_cppi3_mc6 = np.nanmean(function_list_perf_measure(returns_cppi3_mc6,"sharpe"))

sharpe_cppi3_bt = np.nanmean(function_list_perf_measure(returns_cppi3_bt,"sharpe"))
# b.2. m = 5
sharpe_cppi5_mc1 = np.nanmean(function_list_perf_measure(returns_cppi5_mc1,"sharpe"))
sharpe_cppi5_mc2 = np.nanmean(function_list_perf_measure(returns_cppi5_mc2,"sharpe"))
sharpe_cppi5_mc3 = np.nanmean(function_list_perf_measure(returns_cppi5_mc3,"sharpe"))
sharpe_cppi5_mc4 = np.nanmean(function_list_perf_measure(returns_cppi5_mc4,"sharpe"))
sharpe_cppi5_mc5 = np.nanmean(function_list_perf_measure(returns_cppi5_mc5,"sharpe"))
sharpe_cppi5_mc6 = np.nanmean(function_list_perf_measure(returns_cppi5_mc6,"sharpe"))

sharpe_cppi5_bt = np.nanmean(function_list_perf_measure(returns_cppi5_bt,"sharpe"))
# c. Option-based portfolio insurance strategy (OBPI)
sharpe_obpi_mc1 = np.nanmean(function_list_perf_measure(returns_obpi_mc1,"sharpe"))
sharpe_obpi_mc2 = np.nanmean(function_list_perf_measure(returns_obpi_mc2, "sharpe"))
sharpe_obpi_mc3 = np.nanmean(function_list_perf_measure(returns_obpi_mc3, "sharpe"))
sharpe_obpi_mc4 = np.nanmean(function_list_perf_measure(returns_obpi_mc4, "sharpe"))
sharpe_obpi_mc5 = np.nanmean(function_list_perf_measure(returns_obpi_mc5, "sharpe"))
sharpe_obpi_mc6 = np.nanmean(function_list_perf_measure(returns_obpi_mc6, "sharpe"))

sharpe_obpi_bt = np.nanmean(function_list_perf_measure(returns_obpi_bt, "sharpe"))

# d. VaR-based portfolio insurance strategy (VaRPI)
sharpe_varpi_mc1 = np.nanmean(function_list_perf_measure(returns_varpi_mc1, "sharpe"))
sharpe_varpi_mc2 = np.nanmean(function_list_perf_measure(returns_varpi_mc2, "sharpe"))
sharpe_varpi_mc3 = np.nanmean(function_list_perf_measure(returns_varpi_mc3, "sharpe"))
sharpe_varpi_mc4 = np.nanmean(function_list_perf_measure(returns_varpi_mc4, "sharpe"))
sharpe_varpi_mc5 = np.nanmean(function_list_perf_measure(returns_varpi_mc5, "sharpe"))
sharpe_varpi_mc6 = np.nanmean(function_list_perf_measure(returns_varpi_mc6, "sharpe"))

# Not available for backtesting
sharpe_espi_bt = np.nanmean(function_list_perf_measure(returns_espi_bt, "sharpe"))

# e. Stop-loss portfolio insurance strategy (SLPI)
sharpe_slpi_mc1 = np.nanmean(function_list_perf_measure(returns_slpi_mc1, "sharpe"))
sharpe_slpi_mc2 = np.nanmean(function_list_perf_measure(returns_slpi_mc2, "sharpe"))
sharpe_slpi_mc3 = np.nanmean(function_list_perf_measure(returns_slpi_mc3, "sharpe"))
sharpe_slpi_mc4 = np.nanmean(function_list_perf_measure(returns_slpi_mc4, "sharpe"))
sharpe_slpi_mc5 = np.nanmean(function_list_perf_measure(returns_slpi_mc5, "sharpe"))
sharpe_slpi_mc6 = np.nanmean(function_list_perf_measure(returns_slpi_mc6, "sharpe"))

sharpe_slpi_bt = np.nanmean(function_list_perf_measure(returns_slpi_bt, "sharpe"))

##########

# 2.2.2. The Sortino ratio

# a. Buy and hold strategy
sortino_bh_mc1 = np.nanmean(function_list_perf_measure(returns_bh_mc1, "sortino"))
sortino_bh_mc2 = np.nanmean(function_list_perf_measure(returns_bh_mc2, "sortino"))
sortino_bh_mc3 = np.nanmean(function_list_perf_measure(returns_bh_mc3, "sortino"))
sortino_bh_mc4 = np.nanmean(function_list_perf_measure(returns_bh_mc4, "sortino"))
sortino_bh_mc5 = np.nanmean(function_list_perf_measure(returns_bh_mc5, "sortino"))
sortino_bh_mc6 = np.nanmean(function_list_perf_measure(returns_bh_mc6, "sortino"))

sortino_bh_bt = np.nanmean(function_list_perf_measure(returns_bh_bt, "sortino"))

# b. Constant Proportion portfolio insurance strategy (cppi)
# b.1. m = 3
sortino_cppi3_mc1 = np.nanmean(function_list_perf_measure(returns_cppi3_mc1,"sortino"))
sortino_cppi3_mc2 = np.nanmean(function_list_perf_measure(returns_cppi3_mc2,"sortino"))
sortino_cppi3_mc3 = np.nanmean(function_list_perf_measure(returns_cppi3_mc3,"sortino"))
sortino_cppi3_mc4 = np.nanmean(function_list_perf_measure(returns_cppi3_mc4,"sortino"))
sortino_cppi3_mc5 = np.nanmean(function_list_perf_measure(returns_cppi3_mc5,"sortino"))
sortino_cppi3_mc6 = np.nanmean(function_list_perf_measure(returns_cppi3_mc6,"sortino"))
sortino_cppi3_bt = np.nanmean(function_list_perf_measure(returns_cppi3_bt,"sortino"))

# b.2. m = 5
sortino_cppi5_mc1 = np.nanmean(function_list_perf_measure(returns_cppi5_mc1,"sortino"))
sortino_cppi5_mc2 = np.nanmean(function_list_perf_measure(returns_cppi5_mc2,"sortino"))
sortino_cppi5_mc3 = np.nanmean(function_list_perf_measure(returns_cppi5_mc3,"sortino"))
sortino_cppi5_mc4 = np.nanmean(function_list_perf_measure(returns_cppi5_mc4,"sortino"))
sortino_cppi5_mc5 = np.nanmean(function_list_perf_measure(returns_cppi5_mc5,"sortino"))
sortino_cppi5_mc6 = np.nanmean(function_list_perf_measure(returns_cppi5_mc6,"sortino"))
sortino_cppi5_bt = np.nanmean(function_list_perf_measure(returns_cppi5_bt,"sortino"))

# c. Option-based portfolio insurance strategy (OBPI)
sortino_obpi_mc1 = np.nanmean(function_list_perf_measure(returns_obpi_mc1,"sortino"))
sortino_obpi_mc2 = np.nanmean(function_list_perf_measure(returns_obpi_mc2,"sortino"))
sortino_obpi_mc3 = np.nanmean(function_list_perf_measure(returns_obpi_mc3,"sortino"))
sortino_obpi_mc4 = np.nanmean(function_list_perf_measure(returns_obpi_mc4,"sortino"))
sortino_obpi_mc5 = np.nanmean(function_list_perf_measure(returns_obpi_mc5,"sortino"))
sortino_obpi_mc6 = np.nanmean(function_list_perf_measure(returns_obpi_mc6,"sortino"))
sortino_obpi_bt = np.nanmean(function_list_perf_measure(returns_obpi_bt,"sortino"))

# d. VaR-based portfolio insurance strategy (VaRPI)
sortino_varpi_mc1 = np.nanmean(function_list_perf_measure(returns_varpi_mc1,"sortino"))
sortino_varpi_mc2 = np.nanmean(function_list_perf_measure(returns_varpi_mc2,"sortino"))
sortino_varpi_mc3 = np.nanmean(function_list_perf_measure(returns_varpi_mc3,"sortino"))
sortino_varpi_mc4 = np.nanmean(function_list_perf_measure(returns_varpi_mc4,"sortino"))
sortino_varpi_mc5 = np.nanmean(function_list_perf_measure(returns_varpi_mc5,"sortino"))
sortino_varpi_mc6 = np.nanmean(function_list_perf_measure(returns_varpi_mc6,"sortino"))

# Not available for backtesting sortino_varpi_bt = np.nanmean(function_list_perf_measure(returns_varpi_bt,"sortino"))

# e. Stop-loss portfolio insurance strategy (SLPI)
sortino_slpi_mc1 = np.nanmean(function_list_perf_measure(returns_slpi_mc1,"sortino"))
sortino_slpi_mc2 = np.nanmean(function_list_perf_measure(returns_slpi_mc2,"sortino"))
sortino_slpi_mc3 = np.nanmean(function_list_perf_measure(returns_slpi_mc3,"sortino"))
sortino_slpi_mc4 = np.nanmean(function_list_perf_measure(returns_slpi_mc4,"sortino"))
sortino_slpi_mc5 = np.nanmean(function_list_perf_measure(returns_slpi_mc5,"sortino"))
sortino_slpi_mc6 = np.nanmean(function_list_perf_measure(returns_slpi_mc6,"sortino"))

sortino_slpi_bt = np.nanmean(function_list_perf_measure(returns_slpi_bt,"sortino"))
# 2.2.3. The Omega ratio

# a. Buy and hold strategy
omega_bh_mc1 = np.nanmean(function_list_perf_measure(returns_bh_mc1,"omega"))
omega_bh_mc2 = np.nanmean(function_list_perf_measure(returns_bh_mc2,"omega"))
omega_bh_mc3 = np.nanmean(function_list_perf_measure(returns_bh_mc3,"omega"))
omega_bh_mc4 = np.nanmean(function_list_perf_measure(returns_bh_mc4,"omega"))
omega_bh_mc5 = np.nanmean(function_list_perf_measure(returns_bh_mc5,"omega"))
omega_bh_mc6 = np.nanmean(function_list_perf_measure(returns_bh_mc6,"omega"))

omega_bh_bt = np.nanmean(function_list_perf_measure(returns_bh_bt,"omega"))
# b. Constant Proportion portfolio insurance strategy (cppi)
# b.1. m = 3
omega_cppi3_mc1 = np.nanmean(function_list_perf_measure(returns_cppi3_mc1,"omega"))
omega_cppi3_mc2 = np.nanmean(function_list_perf_measure(returns_cppi3_mc2,"omega"))
omega_cppi3_mc3 = np.nanmean(function_list_perf_measure(returns_cppi3_mc3,"omega"))
omega_cppi3_mc4 = np.nanmean(function_list_perf_measure(returns_cppi3_mc4,"omega"))
omega_cppi3_mc5 = np.nanmean(function_list_perf_measure(returns_cppi3_mc5,"omega"))
omega_cppi3_mc6 = np.nanmean(function_list_perf_measure(returns_cppi3_mc6,"omega"))

omega_cppi3_bt = np.nanmean(function_list_perf_measure(returns_cppi3_bt,"omega"))
# b.2. m = 5
omega_cppi5_mc1 = np.nanmean(function_list_perf_measure(returns_cppi5_mc1,"omega"))
omega_cppi5_mc2 = np.nanmean(function_list_perf_measure(returns_cppi5_mc2,"omega"))
omega_cppi5_mc3 = np.nanmean(function_list_perf_measure(returns_cppi5_mc3,"omega"))
omega_cppi5_mc4 = np.nanmean(function_list_perf_measure(returns_cppi5_mc4,"omega"))
omega_cppi5_mc5 = np.nanmean(function_list_perf_measure(returns_cppi5_mc5,"omega"))
omega_cppi5_mc6 = np.nanmean(function_list_perf_measure(returns_cppi5_mc6,"omega"))

omega_cppi5_bt = np.nanmean(function_list_perf_measure(returns_cppi5_bt,"omega"))
# c. Option-based portfolio insurance strategy (OBPI)
omega_obpi_mc1 = np.nanmean(function_list_perf_measure(returns_obpi_mc1,"omega"))
omega_obpi_mc2 = np.nanmean(function_list_perf_measure(returns_obpi_mc2,"omega"))
omega_obpi_mc3 = np.nanmean(function_list_perf_measure(returns_obpi_mc3,"omega"))
omega_obpi_mc4 = np.nanmean(function_list_perf_measure(returns_obpi_mc4,"omega"))
omega_obpi_mc5 = np.nanmean(function_list_perf_measure(returns_obpi_mc5,"omega"))
omega_obpi_mc6 = np.nanmean(function_list_perf_measure(returns_obpi_mc6,"omega"))
omega_obpi_bt = np.nanmean(function_list_perf_measure(returns_obpi_bt, "omega"))
# d. VaR-based portfolio insurance strategy (VaRPI)
omega_varpi_MC1 = np.nanmean(function_list_perf_measure(returns_varpi_MC1, "omega"))
omega_varpi_MC2 = np.nanmean(function_list_perf_measure(returns_varpi_MC2, "omega"))
omega_varpi_MC3 = np.nanmean(function_list_perf_measure(returns_varpi_MC3, "omega"))
omega_varpi_MC4 = np.nanmean(function_list_perf_measure(returns_varpi_MC4, "omega"))
omega_varpi_MC5 = np.nanmean(function_list_perf_measure(returns_varpi_MC5, "omega"))
omega_varpi_MC6 = np.nanmean(function_list_perf_measure(returns_varpi_MC6, "omega"))

# Not available for backtesting omega_varpi_BT = np.nanmean(function_list_perf_measure(returns_varpi_BT, "omega"))
# e. Stop-loss portfolio insurance strategy (SLPI)
omega_slpi_MC1 = np.nanmean(function_list_perf_measure(returns_slpi_MC1, "omega"))
omega_slpi_MC2 = np.nanmean(function_list_perf_measure(returns_slpi_MC2, "omega"))
omega_slpi_MC3 = np.nanmean(function_list_perf_measure(returns_slpi_MC3, "omega"))
omega_slpi_MC4 = np.nanmean(function_list_perf_measure(returns_slpi_MC4, "omega"))
omega_slpi_MC5 = np.nanmean(function_list_perf_measure(returns_slpi_MC5, "omega"))
omega_slpi_MC6 = np.nanmean(function_list_perf_measure(returns_slpi_MC6, "omega"))
omega_slpi_BT = np.nanmean(function_list_perf_measure(returns_slpi_BT, "omega"))

# 2.2.3. The Floor Protection ratio
# a. Buy and hold strategy
prot_bh_MC1 = prot(bh_MC1)
prot_bh_MC2 = prot(bh_MC2)
prot_bh_MC3 = prot(bh_MC3)
prot_bh_MC4 = prot(bh_MC4)
prot_bh_MC5 = prot(bh_MC5)
prot_bh_MC6 = prot(bh_MC6)
# b. Constant Proportion portfolio insurance strategy (cppi)
# b.1. m = 3
prot_cppi3_MC1 = prot(cppi3_MC1)
prot_cppi3_MC2 = prot(cppi3_MC2)
prot_cppi3_MC3 = prot(cppi3_MC3)
prot_cppi3_MC4 = prot(cppi3_MC4)
prot_cppi3_MC5 = prot(cppi3_MC5)
prot_cppi3_MC6 = prot(cppi3_MC6)
# b.2. m = 5
prot_cppi5_MC1 = prot(cppi5_MC1)
prot_cppi5_MC2 = prot(cppi5_MC2)
prot_cppi5_MC3 = prot(cppi5_MC3)
prot_cppi5_MC4 = prot(cppi5_MC4)
prot_cppi5_mc5 = prot(cppi5_mc5)
prot_cppi5_mc6 = prot(cppi5_mc6)

# c. Option-based portfolio insurance strategy (OBPI)
prot_obpi_mc1 = prot(obpi_mc1)
prot_obpi_mc2 = prot(obpi_mc2)
prot_obpi_mc3 = prot(obpi_mc3)
prot_obpi_mc4 = prot(obpi_mc4)
prot_obpi_mc5 = prot(obpi_mc5)
prot_obpi_mc6 = prot(obpi_mc6)

# d. VaR-based portfolio insurance strategy (VaRPI)
prot_varpi_mc1 = prot(varpi_mc1)
prot_varpi_mc2 = prot(varpi_mc2)
prot_varpi_mc3 = prot(varpi_mc3)
prot_varpi_mc4 = prot(varpi_mc4)
prot_varpi_mc5 = prot(varpi_mc5)
prot_varpi_mc6 = prot(varpi_mc6)

# e. Stop-loss portfolio insurance strategy (SLPI)
prot_slpi_mc1 = prot(slpi_mc1)
prot_slpi_mc2 = prot(slpi_mc2)
prot_slpi_mc3 = prot(slpi_mc3)
prot_slpi_mc4 = prot(slpi_mc4)
prot_slpi_mc5 = prot(slpi_mc5)
prot_slpi_mc6 = prot(slpi_mc6)

# CHAPTER 3: RESULTS AND DEFINITION OF THE DOMINANT STRATEGY

### Monte Carlo simulation method

#### Definition of function for ranking (except Sharpe and Sortino ratios) for Monte Carlo

```python
def rank_strat(lowest_is_highest, position, bh_local, cppi3_local, cppi5_local, obpi_local, varpi_local, slpi_local):
    a = st.rankdata([bh_local, cppi3_local, cppi5_local, obpi_local, varpi_local, slpi_local], 'min')
    b = st.rankdata([bh_local, cppi3_local, cppi5_local, obpi_local, varpi_local, slpi_local], 'max')
    if lowest_is_highest == True:
        rank_result = a[position]
    else:
        rank_result = (len(b) - b + 1)[position]
    return rank_result
```

#### Definition of function for ranking the Sharpe and Sortino ratios for Monte Carlo

```python
def rank_strat_pos (lowest_is_highest, position, bh_local, cppi3_local, cppi5_local, obpi_local, varpi_local, slpi_local):
    if bh_local >= 0 and cppi3_local >= 0 and cppi5_local >= 0 and obpi_local >= 0 and varpi_local >= 0 and slpi_local >= 0:
        a = st.rankdata([bh_local, cppi3_local, cppi5_local, obpi_local, varpi_local, slpi_local], 'min')
        b = st.rankdata([bh_local, cppi3_local, cppi5_local, obpi_local, varpi_local, slpi_local], 'max')
```

108
if lowest_is_highest == True:
    rank_result = a[position]
else:
    rank_result = (len(b) - b + 1)[position]
else:
    rank_result = 0
return rank_result

# Reference of codes https://plot.ly/python/table/
def result_table(name, mean_local_bh, std_local_bh, skew_local_bh, kurt_local_bh, sharpe_local_bh, sortino_local_bh, omega_local_bh, prot_local_bh, mean_local_cppi3, std_local_cppi3, skew_local_cppi3, kurt_local_cppi3, sharpe_local_cppi3, sortino_local_cppi3, omega_local_cppi3, prot_local_cppi3, mean_local_cppi5, std_local_cppi5, skew_local_cppi5, kurt_local_cppi5, sharpe_local_cppi5, sortino_local_cppi5, omega_local_cppi5, prot_local_cppi5, mean_local_obpi, std_local_obpi, skew_local_obpi, kurt_local_obpi, sharpe_local_obpi, sortino_local_obpi, omega_local_obpi, prot_local_obpi, mean_local_varpi, std_local_varpi, skew_local_varpi, kurt_local_varpi, sharpe_local_varpi, sortino_local_varpi, omega_local_varpi, prot_local_varpi, mean_local_slpi, std_local_slpi, skew_local_slpi, kurt_local_slpi, sharpe_local_slpi, sortino_local_slpi, omega_local_slpi, prot_local_slpi):

    rank_bh_mean = rank_strat(False, 0, mean_local_bh, mean_local_cppi3, mean_local_cppi5, mean_local_obpi, mean_local_varpi, mean_local_slpi)
    rank_cppi3_mean = rank_strat(False, 1, mean_local_bh, mean_local_cppi3, mean_local_cppi5, mean_local_obpi, mean_local_varpi, mean_local_slpi)
    rank_cppi5_mean = rank_strat(False, 2, mean_local_bh, mean_local_cppi3, mean_local_cppi5, mean_local_obpi, mean_local_varpi, mean_local_slpi)
    rank_obpi_mean = rank_strat(False, 3, mean_local_bh, mean_local_cppi3, mean_local_cppi5, mean_local_obpi, mean_local_varpi, mean_local_slpi)
    rank_varpi_mean = rank_strat(False, 4, mean_local_bh, mean_local_cppi3, mean_local_cppi5, mean_local_obpi, mean_local_varpi, mean_local_slpi)
    rank_slpi_mean = rank_strat(False, 5, mean_local_bh, mean_local_cppi3, mean_local_cppi5, mean_local_obpi, mean_local_varpi, mean_local_slpi)

    rank_bh_std = rank_strat(True, 0, std_local_bh, std_local_cppi3, std_local_cppi5, std_local_obpi, std_local_varpi, std_local_slpi)
    rank_cppi3_std = rank_strat(True, 1, std_local_bh, std_local_cppi3, std_local_cppi5, std_local_obpi, std_local_varpi, std_local_slpi)
    rank_cppi5_std = rank_strat(True, 2, std_local_bh, std_local_cppi3, std_local_cppi5, std_local_obpi, std_local_varpi, std_local_slpi)
    rank_obpi_std = rank_strat(True, 3, std_local_bh, std_local_cppi3, std_local_cppi5, std_local_obpi, std_local_varpi, std_local_slpi)
    rank_varpi_std = rank_strat(True, 4, std_local_bh, std_local_cppi3, std_local_cppi5, std_local_obpi, std_local_varpi, std_local_slpi)
std_local_bh, std_local_cppi3, std_local_cppi5, std_local_obpi, std_local_varpi, std_local_slpi

rank_slpi_std = rank_strat(True, 5, std_local_bh, std_local_cppi3, std_local_cppi5, std_local_obpi, std_local_varpi, std_local_slpi)

rank_bh_sharpe = rank_strat_pos(False, 0, sharpe_local_bh, sharpe_local_cppi3, sharpe_local_cppi5, sharpe_local_obpi, sharpe_local_varpi, sharpe_local_slpi)
rank_cppi3_sharpe = rank_strat_pos(False, 1, sharpe_local_bh, sharpe_local_cppi3, sharpe_local_cppi5, sharpe_local_obpi, sharpe_local_varpi, sharpe_local_slpi)
rank_cppi5_sharpe = rank_strat_pos(False, 2, sharpe_local_bh, sharpe_local_cppi3, sharpe_local_cppi5, sharpe_local_obpi, sharpe_local_varpi, sharpe_local_slpi)
rank_obpi_sharpe = rank_strat_pos(False, 3, sharpe_local_bh, sharpe_local_cppi3, sharpe_local_cppi5, sharpe_local_obpi, sharpe_local_varpi, sharpe_local_slpi)
rank_varpi_sharpe = rank_strat_pos(False, 4, sharpe_local_bh, sharpe_local_cppi3, sharpe_local_cppi5, sharpe_local_obpi, sharpe_local_varpi, sharpe_local_slpi)
rank_slpi_sharpe = rank_strat_pos(False, 5, sharpe_local_bh, sharpe_local_cppi3, sharpe_local_cppi5, sharpe_local_obpi, sharpe_local_varpi, sharpe_local_slpi)

rank_bh_sortino = rank_strat_pos(False, 0, sortino_local_bh, sortino_local_cppi3, sortino_local_cppi5, sortino_local_obpi, sortino_local_varpi, sortino_local_slpi)
rank_cppi3_sortino = rank_strat_pos(False, 1, sortino_local_bh, sortino_local_cppi3, sortino_local_cppi5, sortino_local_obpi, sortino_local_varpi, sortino_local_slpi)
rank_cppi5_sortino = rank_strat_pos(False, 2, sortino_local_bh, sortino_local_cppi3, sortino_local_cppi5, sortino_local_obpi, sortino_local_varpi, sortino_local_slpi)
rank_obpi_sortino = rank_strat_pos(False, 3, sortino_local_bh, sortino_local_cppi3, sortino_local_cppi5, sortino_local_obpi, sortino_local_varpi, sortino_local_slpi)
rank_varpi_sortino = rank_strat_pos(False, 4, sortino_local_bh, sortino_local_cppi3, sortino_local_cppi5, sortino_local_obpi, sortino_local_varpi, sortino_local_slpi)
rank_slpi_sortino = rank_strat_pos(False, 5, sortino_local_bh, sortino_local_cppi3, sortino_local_cppi5, sortino_local_obpi, sortino_local_varpi, sortino_local_slpi)

rank_bh_omega = rank_strat(False, 0, omega_local_bh, omega_local_cppi3, omega_local_cppi5, omega_local_obpi, omega_local_varpi, omega_local_slpi)
rank_cppi3_omega = rank_strat(False, 1, omega_local_bh, omega_local_cppi3, omega_local_cppi5, omega_local_obpi, omega_local_varpi, omega_local_slpi)
rank_cppi5_omega = rank_strat(False, 2, omega_local_bh, omega_local_cppi3, omega_local_cppi5, omega_local_obpi, omega_local_varpi, omega_local_slpi)
rank_obpi_omega = rank_strat(False, 3, omega_local_bh, omega_local_cppi3, omega_local_cppi5, omega_local_obpi, omega_local_varpi, omega_local_slpi)
rank_varpi_omega = rank_strat(False, 4, omega_local_bh, omega_local_cppi3, omega_local_cppi5, omega_local_obpi, omega_local_varpi, omega_local_slpi)
rank_slpi_omega = rank_strat(False, 5, omega_local_bh, omega_local_cppi3, omega_local_cppi5, omega_local_obpi, omega_local_varpi, omega_local_slpi)
aver_rank_bh = np.nanmean([rank_bh_mean, rank_bh_std, rank_bh_sharpe, rank_bh_sortino, rank_bh_omega])
aver_rank_cppi3 = np.nanmean([rank_cppi3_mean, rank_cppi3_std, rank_cppi3_sharpe, rank_cppi3_sortino, rank_cppi3_omega])
aver_rank_cppi5 = np.nanmean([rank_cppi5_mean, rank_cppi5_std, rank_cppi5_sharpe, rank_cppi5_sortino, rank_cppi5_omega])
aver_rank_obpi = np.nanmean([rank_obpi_mean, rank_obpi_std, rank_obpi_sharpe, rank_obpi_sortino, rank_obpi_omega])
aver_rank_varpi = np.nanmean([rank_varpi_mean, rank_varpi_std, rank_varpi_sharpe, rank_varpi_sortino, rank_varpi_omega])
aver_rank_slpi = np.nanmean([rank_slpi_mean, rank_slpi_std, rank_slpi_sharpe, rank_slpi_sortino, rank_slpi_omega])

aver_rank_bh_overall = rank_strat(True, 0, aver_rank_bh, aver_rank_cppi3, aver_rank_cppi5, aver_rank_obpi, aver_rank_varpi, aver_rank_slpi)
aver_rank_cppi3_overall = rank_strat(True, 1, aver_rank_bh, aver_rank_cppi3, aver_rank_cppi5, aver_rank_obpi, aver_rank_varpi, aver_rank_slpi)
aver_rank_cppi5_overall = rank_strat(True, 2, aver_rank_bh, aver_rank_cppi3, aver_rank_cppi5, aver_rank_obpi, aver_rank_varpi, aver_rank_slpi)
aver_rank_obpi_overall = rank_strat(True, 3, aver_rank_bh, aver_rank_cppi3, aver_rank_cppi5, aver_rank_obpi, aver_rank_varpi, aver_rank_slpi)
aver_rank_varpi_overall = rank_strat(True, 4, aver_rank_bh, aver_rank_cppi3, aver_rank_cppi5, aver_rank_obpi, aver_rank_varpi, aver_rank_slpi)
aver_rank_slpi_overall = rank_strat(True, 5, aver_rank_bh, aver_rank_cppi3, aver_rank_cppi5, aver_rank_obpi, aver_rank_varpi, aver_rank_slpi)

['BH', round(mean_local_bh, 4), rank_bh_mean, round(std_local_bh, 4), rank_bh_std, round(skew_local_bh, 4), round(kurt_local_bh, 4), round(sharpe_local_bh, 4), rank_bh_sharpe, round(sortino_local_bh, 4), rank_bh_sortino, round(omega_local_bh, 4), rank_bh_omega, round(prot_local_bh, 4), rank_bh_overall],
['CPPI3', round(mean_local_cppi3, 4), rank_cppi3_mean, round(std_local_cppi3, 4), rank_cppi3_std, round(skew_local_cppi3, 4), round(kurt_local_cppi3, 4), round(sharpe_local_cppi3, 4), rank_cppi3_sharpe, round(sortino_local_cppi3, 4), rank_cppi3_sortino, round(omega_local_cppi3, 4), rank_cppi3_omega, round(prot_local_cppi3, 4), rank_cppi3_overall],
['CPPI5', round(mean_local_cppi5, 4), rank_cppi5_mean, round(std_local_cppi5, 4), rank_cppi5_std, round(skew_local_cppi5, 4), round(kurt_local_cppi5, 4), round(sharpe_local_cppi5, 4), rank_cppi5_sharpe, round(sortino_local_cppi5, 4), rank_cppi5_sortino, round(omega_local_cppi5, 4), rank_cppi5_omega, round(prot_local_cppi5, 4), rank_cppi5_overall],
['OBPI', round(mean_local_obpi, 4), rank_obpi_mean, round(std_local_obpi, 4), rank_obpi_std, round(skew_local_obpi, 4), round(kurt_local_obpi, 4), round(sharpe_local_obpi, 4), rank_obpi_sharpe, round(sortino_local_obpi, 4), rank_obpi_sortino, round(omega_local_obpi, 4), rank_obpi_omega, round(prot_local_obpi, 4), rank_obpi_overall],
['VarPI', round(mean_local_varpi, 4), rank_varpi_mean, round(std_local_varpi, 4), rank_varpi_std, round(skew_local_varpi, 4), round(kurt_local_varpi, 4), round(sharpe_local_varpi, 4), rank_varpi_sharpe, round(sortino_local_varpi, 4), rank_varpi_sortino, round(omega_local_varpi, 4), rank_varpi_omega, round(prot_local_varpi, 4), rank_varpi_overall],
['SLPI', round(mean_local_slpi, 4), rank_slpi_mean, round(std_local_slpi, 4), rank_slpi_std, round(skew_local_slpi, 4), round(kurt_local_slpi, 4), round(sharpe_local_slpi, 4), rank_slpi_sharpe, round(sortino_local_slpi, 4), rank_slpi_sortino, round(omega_local_slpi, 4), rank_slpi_omega, round(prot_local_slpi, 4), rank_slpi_overall]]
round(sortino_local_varpi, 4), rank_varpi_sortino, round(omega_local_varpi, 4), rank_varpi_omega, round(prot_local_varpi, 4), rank_varpi_overall),
['SLPI', round(mean_local_slpi, 4), rank_slpi_mean, round(std_local_slpi, 4), rank_slpi_std, round(skew_local_slpi, 4), round(kurt_local_slpi, 4), round(sharpe_local_slpi, 4), rank_slpi_sharpe, round(sortino_local_slpi, 4), rank_slpi_sortino, round(omega_local_slpi, 4), rank_slpi_omega, round(prot_local_slpi, 4), rank_slpi_overall]]

table = ff.create_table(data_matrix)
return py.plot(table, filename='Table ' + name)

# 3.1.1. Bear market scenario
# a. Scenario 1 ($\mu = -15\%, \sigma = 20\%$)
result_table_mc1 = result_table("MC1", amean_bh_mc1, astd_bh_mc1, askew_bh_mc1, akurt_bh_mc1, sharpe_bh_mc1, sortino_bh_mc1, omega_bh_mc1, prot_bh_mc1, amean_cppi3_mc1, astd_cppi3_mc1, askew_cppi3_mc1, akurt_cppi3_mc1, sharpe_cppi3_mc1, sortino_cppi3_mc1, omega_cppi3_mc1, prot_cppi3_mc1, amean_cppi5_mc1, astd_cppi5_mc1, askew_cppi5_mc1, akurt_cppi5_mc1, sharpe_cppi5_mc1, sortino_cppi5_mc1, omega_cppi5_mc1, prot_cppi5_mc1, amean_obpi_mc1, astd_obpi_mc1, askew_obpi_mc1, akurt_obpi_mc1, sharpe_obpi_mc1, sortino_obpi_mc1, omega_obpi_mc1, prot_obpi_mc1, amean_varpi_mc1, astd_varpi_mc1, askew_varpi_mc1, akurt_varpi_mc1, sharpe_varpi_mc1, sortino_varpi_mc1, omega_varpi_mc1, prot_varpi_mc1, amean_slpi_mc1, astd_slpi_mc1, askew_slpi_mc1, akurt_slpi_mc1, sharpe_slpi_mc1, sortino_slpi_mc1, omega_slpi_mc1, prot_slpi_mc1)

# b. Scenario 2 ($\mu = -15\%, \sigma = 40\%$)
result_table_mc2 = result_table("MC2", amean_bh_mc2, astd_bh_mc2, askew_bh_mc2, akurt_bh_mc2, sharpe_bh_mc2, sortino_bh_mc2, omega_bh_mc2, prot_bh_mc2, amean_cppi3_mc2, astd_cppi3_mc2, askew_cppi3_mc2, akurt_cppi3_mc2, sharpe_cppi3_mc2, sortino_cppi3_mc2, omega_cppi3_mc2, prot_cppi3_mc2, amean_cppi5_mc2, astd_cppi5_mc2, askew_cppi5_mc2, akurt_cppi5_mc2, sharpe_cppi5_mc2, sortino_cppi5_mc2, omega_cppi5_mc2, prot_cppi5_mc2, amean_obpi_mc2, astd_obpi_mc2, askew_obpi_mc2, akurt_obpi_mc2, sharpe_obpi_mc2, sortino_obpi_mc2, omega_obpi_mc2, prot_obpi_mc2, amean_varpi_mc2, astd_varpi_mc2, askew_varpi_mc2, akurt_varpi_mc2, sharpe_varpi_mc2, sortino_varpi_mc2, omega_varpi_mc2, prot_varpi_mc2, amean_slpi_mc2, astd_slpi_mc2, askew_slpi_mc2, akurt_slpi_mc2, sharpe_slpi_mc2, sortino_slpi_mc2, omega_slpi_mc2, prot_slpi_mc2)

# 3.1.2. Normal market scenario
# a. Scenario 3 ($\mu = 8\%, \sigma = 20\%$)
result_table_mc3 =
result_table("MC3", amean_bh_mc3, astd_bh_mc3, askew_bh_mc3, akurt_bh_mc3, sharpe_bh_mc3, sortino_bh_mc3, omega_bh_mc3, prot_bh_mc3,
sharpe_cpii3_mc3, sortino_cpii3_mc3, omega_cpii3_mc3, prot_cpii3_mc3,
sharpe_cpii5_mc3, sortino_cpii5_mc3, omega_cpii5_mc3, prot_cpii5_mc3,
omega_obpi_mc3, prot_obpi_mc3,
sharpe_varpi_mc3, sortino_varpi_mc3, omega_varpi_mc3, prot_varpi_mc3,
omega_slpi_mc3, prot_slpi_mc3)

# b. Scenario 4 (μ = 8%, σ = 40%)
result_table_mc4 =
result_table("MC4", amean_bh_mc4, astd_bh_mc4, askew_bh_mc4, akurt_bh_mc4, sharpe_bh_mc4, sortino_bh_mc4, omega_bh_mc4, prot_bh_mc4,
sharpe_cpii3_mc4, sortino_cpii3_mc4, omega_cpii3_mc4, prot_cpii3_mc4,
sharpe_cpii5_mc4, sortino_cpii5_mc4, omega_cpii5_mc4, prot_cpii5_mc4,
omega_obpi_mc4, prot_obpi_mc4,
sharpe_varpi_mc4, sortino_varpi_mc4, omega_varpi_mc4, prot_varpi_mc4,
omega_slpi_mc4, prot_slpi_mc4)

# 3.1.3. Bull market scenario
# a. Scenario 5 (μ = 15%, σ = 20%)
result_table_mc5 =
result_table("MC5", amean_bh_mc5, astd_bh_mc5, askew_bh_mc5, akurt_bh_mc5, sharpe_bh_mc5, sortino_bh_mc5, omega_bh_mc5, prot_bh_mc5,
sharpe_cpii3_mc5, sortino_cpii3_mc5, omega_cpii3_mc5, prot_cpii3_mc5,
sharpe_cpii5_mc5, sortino_cpii5_mc5, omega_cpii5_mc5, prot_cpii5_mc5,
omega_obpi_mc5, prot_obpi_mc5,
sharpe_varpi_mc5, sortino_varpi_mc5, omega_varpi_mc5, prot_varpi_mc5,
omega_slpi_mc5, prot_slpi_mc5)
omega_slpi_mc5, prot_slpi_mc5)
# b. Scenario 6 (15%, σ = 40%)
result_table_mc6 =
result_table("MC6", amean_bh_mc6, astd_bh_mc6, skew_bh_mc6, akurt_bh_mc6, sharpe_bh_mc6, sortino_bh_mc6, omega_bh_mc6, prot_bh_mc6,
amean_cppi3_mc6, astd_cppi3_mc6, skew_cppi3_mc6, akurt_cppi3_mc6,
sharpe_cppi3_mc6, sortino_cppi3_mc6, omega_cppi3_mc6, prot_cppi3_mc6,
amean_cppi5_mc6, astd_cppi5_mc6, skew_cppi5_mc6, akurt_cppi5_mc6,
sharpe_cppi5_mc6, sortino_cppi5_mc6, omega_cppi5_mc6, prot_cppi5_mc6,
omega_obpi_mc6, prot_obpi_mc6,
amean_varpi_mc6, astd_varpi_mc6, skew_varpi_mc6, akurt_varpi_mc6,
sharpe_varpi_mc6, sortino_varpi_mc6, omega_varpi_mc6, prot_varpi_mc6,
amean_slpi_mc6, astd_slpi_mc6, skew_slpi_mc6, akurt_slpi_mc6,
sharpe_slpi_mc6, sortino_slpi_mc6, omega_slpi_mc6, prot_slpi_mc6)

###########
# 3.2. Backtesting
# Definition of function for ranking (except Sharpe and Sortino ratios) for Backtesting

def rank_strat_bt(lowest_is_highest, position, bh_local, cppi3_local, cppi5_local, obpi_local, slpi_local):
    a = st.rankdata([bh_local, cppi3_local, cppi5_local, obpi_local, slpi_local], 'min')
    b = st.rankdata([bh_local, cppi3_local, cppi5_local, obpi_local, slpi_local], 'max')
    if lowest_is_highest == True:
        rank_result = a[position]
    else:
        rank_result = (len(b) - b + 1)[position]
    return rank_result

# Definition of function for ranking the Sharpe and Sortino ratios for Backtesting

def rank_strat_pos_bt(lowest_is_highest, position, bh_local, cppi3_local, cppi5_local, obpi_local, slpi_local):
    if bh_local >= 0 and cppi3_local >= 0 and cppi5_local >= 0 and obpi_local >= 0 and slpi_local >= 0:
        a = st.rankdata([bh_local, cppi3_local, cppi5_local, obpi_local, slpi_local], 'min')
        b = st.rankdata([bh_local, cppi3_local, cppi5_local, obpi_local, slpi_local], 'max')
        if lowest_is_highest == True:
            rank_result = a[position]
        else:
            rank_result = (len(b) - b + 1)[position]
    else:
        rank_result = 0
    return rank_result

# a. Table

def result_table_bt(name, mean_local_bh, std_local_bh, skew_local_bh, kurt_local_bh, sharpe_local_bh, sortino_local_bh,
omega_local_bh, mean_local_cppi3, std_local_cppi3, skew_local_cppi3, kurt_local_cppi3, sharpe_local_cppi3, sortino_local_cppi3, omega_local_cppi3, mean_local_cppi5, std_local_cppi5, skew_local_cppi5, kurt_local_cppi5, sharpe_local_cppi5, sortino_local_cppi5, omega_local_cppi5, mean_local_obpi, std_local_obpi, skew_local_obpi, kurt_local_obpi, sharpe_local_obpi, sortino_local_obpi, omega_local_obpi, mean_local_varpi, std_local_varpi, skew_local_varpi, kurt_local_varpi, sharpe_local_varpi, sortino_local_varpi, omega_local_varpi, mean_local_espi, std_local_espi, skew_local_espi, kurt_local_espi, sharpe_local_espi, sortino_local_espi, omega_local_espi, mean_local_slpi, std_local_slpi, skew_local_slpi, kurt_local_slpi, sharpe_local_slpi, sortino_local_slpi, omega_local_slpi):
  rank_bh_mean = rank_strat_bt(False, 0, mean_local_bh, mean_local_cppi3, mean_local_cppi5, mean_local_obpi, mean_local_slpi)
  rank_cppi3_mean = rank_strat_bt(False, 1, mean_local_bh, mean_local_cppi3, mean_local_cppi5, mean_local_obpi, mean_local_slpi)
  rank_cppi5_mean = rank_strat_bt(False, 2, mean_local_bh, mean_local_cppi3, mean_local_cppi5, mean_local_obpi, mean_local_slpi)
  rank_obpi_mean = rank_strat_bt(False, 3, mean_local_bh, mean_local_cppi3, mean_local_cppi5, mean_local_obpi, mean_local_slpi)
  rank_slpi_mean = rank_strat_bt(False, 4, mean_local_bh, mean_local_cppi3, mean_local_cppi5, mean_local_obpi, mean_local_slpi)
  rank_bh_std = rank_strat_bt(True, 0, std_local_bh, std_local_cppi3, std_local_cppi5, std_local_obpi, std_local_slpi)
  rank_cppi3_std = rank_strat_bt(True, 1, std_local_bh, std_local_cppi3, std_local_cppi5, std_local_obpi, std_local_slpi)
  rank_cppi5_std = rank_strat_bt(True, 2, std_local_bh, std_local_cppi3, std_local_cppi5, std_local_obpi, std_local_slpi)
  rank_obpi_std = rank_strat bt(True, 3, std_local_bh, std_local_cppi3, std_local_cppi5, std_local_obpi, std_local_slpi)
  rank_slpi_std = rank_strat bt(True, 4, std_local_bh, std_local_cppi3, std_local_cppi5, std_local_obpi, std_local_slpi)
  rank_bh_sharpe = rank_strat_pos_bt(False, 0, sharpe_local_bh, sharpe_local_cppi3, sharpe_local_cppi5, sharpe_local_obpi, sharpe_local_slpi)
  rank_cppi3_sharpe = rank_strat_pos bt(False, 1, sharpe_local_bh, sharpe_local_cppi3, sharpe_local_cppi5, sharpe_local_obpi, sharpe_local_slpi)
  rank_cppi5_sharpe = rank_strat_pos bt(False, 2, sharpe_local_bh, sharpe_local_cppi3, sharpe_local_cppi5, sharpe_local_obpi, sharpe_local_slpi)
  rank_obpi_sharpe = rank_strat_pos bt(False, 3, sharpe_local_bh, sharpe_local_cppi3, sharpe_local_cppi5, sharpe_local_obpi, sharpe_local_slpi)
  rank_slpi_sharpe = rank_strat_pos bt(False, 4, sharpe_local_bh, sharpe_local_cppi3, sharpe_local_cppi5, sharpe_local_obpi, sharpe_local_slpi)
  rank_bh_sortino = rank_strat_pos bt(False, 0, sortino_local_bh, sortino_local_cppi3, sortino_local_cppi5, sortino_local_obpi, sortino_local_slpi)
  rank_cppi3_sortino = rank_strat_pos bt(False, 1, sortino_local_bh, sortino_local_cppi3, sortino_local_cppi5, sortino_local_obpi, sortino_local_slpi)
  rank_cppi5_sortino = rank_strat_pos bt(False, 2, sortino_local_bh, sortino_local_cppi3, sortino_local_cppi5, sortino_local_obpi, sortino_local_slpi)
sortino_local_bh, sortino_local_cppi3, sortino_local_cppi5, sortino_local_obpi, sortino_local_slpi)
    rank_obpi_sortino = rank_strat_pos_bt(False, 3, sortino_local_bh, sortino_local_cppi3, sortino_local_cppi5, sortino_local_obpi, sortino_local_slpi)
    rank_slpi_sortino = rank_strat_pos_bt(False, 4, sortino_local_bh, sortino_local_cppi3, sortino_local_cppi5, sortino_local_obpi, sortino_local_slpi)
    rank_bh_omega = rank_strat_bt(False, 0, omega_local_bh, omega_local_cppi3, omega_local_cppi5, omega_local_obpi, omega_local_slpi)
    rank_cppi3_omega = rank_strat_bt(False, 1, omega_local_bh, omega_local_cppi3, omega_local_cppi5, omega_local_obpi, omega_local_slpi)
    rank_cppi5_omega = rank_strat_bt(False, 2, omega_local_bh, omega_local_cppi3, omega_local_cppi5, omega_local_obpi, omega_local_slpi)
    rank_obpi_omega = rank_strat_bt(False, 3, omega_local_bh, omega_local_cppi3, omega_local_cppi5, omega_local_obpi, omega_local_slpi)
    rank_slpi_omega = rank_strat_bt(False, 4, omega_local_bh, omega_local_cppi3, omega_local_cppi5, omega_local_obpi, omega_local_slpi)

    aver_rank_bh = np.nanmean([rank_bh_mean, rank_bh_std, rank_bh_sharpe, rank_bh_sortino, rank_bh_omega])
    aver_rank_cppi3 = np.nanmean([rank_cppi3_mean, rank_cppi3_std, rank_cppi3_sharpe, rank_cppi3_sortino, rank_cppi3_omega])
    aver_rank_cppi5 = np.nanmean([rank_cppi5_mean, rank_cppi5_std, rank_cppi5_sharpe, rank_cppi5_sortino, rank_cppi5_omega])
    aver_rank_obpi = np.nanmean([rank_obpi_mean, rank_obpi_std, rank_obpi_sharpe, rank_obpi_sortino, rank_obpi_omega])
    aver_rank_slpi = np.nanmean([rank_slpi_mean, rank_slpi_std, rank_slpi_sharpe, rank_slpi_sortino, rank_slpi_omega])

    rank_bh_overall = rank_strat_bt(True, 0, aver_rank_bh, aver_rank_cppi3, aver_rank_cppi5, aver_rank_obpi, aver_rank_slpi)
    rank_cppi3_overall = rank_strat_bt(True, 1, aver_rank_bh, aver_rank_cppi3, aver_rank_cppi5, aver_rank_obpi, aver_rank_slpi)
    rank_cppi5_overall = rank_strat_bt(True, 2, aver_rank_bh, aver_rank_cppi3, aver_rank_cppi5, aver_rank_obpi, aver_rank_slpi)
    rank_obpi_overall = rank_strat_bt(True, 3, aver_rank_bh, aver_rank_cppi3, aver_rank_cppi5, aver_rank_obpi, aver_rank_slpi)
    rank_slpi_overall = rank_strat_bt(True, 4, aver_rank_bh, aver_rank_cppi3, aver_rank_cppi5, aver_rank_obpi, aver_rank_slpi)

                    ['BH', round(mean_local_bh, 4), rank_bh_mean, round(std_local_bh, 4), rank_bh_std, round(skew_local_bh, 4),
                     round(kurt_local_bh, 4), round(sharpe_local_bh, 4), rank_bh_sharpe, round(sortino_local_bh, 4), rank_bh_sortino, round(omega_local_bh, 4), rank_bh_omega],
                    ['CPPI3', round(mean_local_cppi3, 4), rank_cppi3_mean, round(std_local_cppi3, 4),
                     round(skew_local_cppi3, 4), round(kurt_local_cppi3, 4),
                     round(sharpe_local_cppi3, 4), rank_cppi3_sharpe, round(sortino_local_cppi3, 4), rank_cppi3_sortino, round(omega_local_cppi3, 4), rank_cppi3_omega],
                    ['CPPI5', round(mean_local_cppi5, 4), rank_cppi5_mean, round(std_local_cppi5, 4),
                     round(skew_local_cppi5, 4), round(kurt_local_cppi5, 4)]]
round(sharpe_local_cppi5, 4), rank_cppi5_sharpe, round(sortino_local_cppi5, 4), rank_cppi5_sortino,
round(omega_local_cppi5, 4), rank_cppi5_omega,
rank_cppi5_overall],
['OBPI', round(mean_local_obpi, 4), rank_obpi_mean, round(std_local_obpi, 4),
rank_obpi_std, round(skew_local_obpi, 4), round(kurt_local_obpi, 4),
round(sharpe_local_obpi, 4), rank_obpi_sharpe, round(sortino_local_obpi, 4), rank_obpi_sortino,
round(omega_local_obpi, 4), rank_obpi_omega,
rank_obpi_overall],
['SLPI', round(mean_local_slpi, 4), rank_slpi_mean, round(std_local_slpi, 4),
rank_slpi_std, round(skew_local_slpi, 4), round(kurt_local_slpi, 4),
round(sharpe_local_slpi, 4), rank_slpi_sharpe, round(sortino_local_slpi, 4), rank_slpi_sortino,
round(omega_local_slpi, 4), rank_slpi_omega,
rank_slpi_overall]]

table = ff.create_table(data_matrix)
return py.plot(table, filename='Table ' + name)
result_table_bt = result_table_bt("BT", amean_bh_bt, astd_bh_bt, askew_bh_bt, akurt_bh_bt, sharpe_bh_bt, sortino_bh_bt, omega_bh_bt,
_ amean_cpsi3_bt, astd_cpsi3_bt, askew_cpsi3_bt, akurt_cpsi3_bt,
sharpe_cpsi3_bt, sortino_cpsi3_bt, omega_cpsi3_bt,
_ amean_cpsi5_bt, astd_cpsi5_bt, askew_cpsi5_bt, akurt_cpsi5_bt,
sharpe_cpsi5_bt, sortino_cpsi5_bt, omega_cpsi5_bt,
_ amean_obpi_bt, astd_obpi_bt, askew_obpi_bt, akurt_obpi_bt, sharpe_obpi_bt, sortino_obpi_bt,
omega_obpi_bt,
_ # amean_varpi_bt, astd_varpi_bt, askew_varpi_bt, akurt_varpi_bt, sharpe_varpi_bt, sortino_varpi_bt,
omega_varpi_bt,
_ # amean_espi_bt, astd_espi_bt, askew_espi_bt, akurt_espi_bt, sharpe_espi_bt, sortino_espi_bt,
omega_espi_bt,
_ amean_slpi_bt, astd_slpi_bt, askew_slpi_bt, akurt_slpi_bt, sharpe_slpi_bt, sortino_slpi_bt,
omega_slpi_bt)

# b. Graph
# Reference of codes https://plot.ly/python/line-charts/#style-line-plots
# Create the floor line
floor_line = []
for i in range(len(mc_array_path_returns(mc1)) + 1):
    floor_line.append(floor*np.exp(-riskfree*(T-i/12))) # we use 12 only for monthly. If use other frequency, need to change.
# Create the value of the insured portfolio according to the market scenarios for 1 out of 10000 paths
def vn30eq_monthly_mc (param):
    vn30eq_monthly_mc = np.zeros((len(mc_array_path_returns(param)[::, 0]) + 1))
    vn30eq_monthly_mc[0] = V0
    for j in range(len(mc_array_path_returns(param)[::, 0])):
vn30eq_monthly_mc[j + 1] = vn30eq_monthly_mc[j] * np.exp(mc_array_path_returns(param)[j, 0])

return vn30eq_monthly_mc

# Function to plot the values of portfolio insurance strategies through years for Backtesting

def plot_value_through_years(name, floor_local, index_local, bh_local, cppi3_local, cppi5_local, obpi_local, slpi_local):
    # to plot for backtesting, please indicate the attribute name with exact "Backtesting".
    year = ['4/12', '5/12', '6/12', '7/12', '8/12', '9/12', '10/12', '11/12', '12/12',
            '1/17', '2/17', '3/17', '4/17']

    # Create lines
    floor = go.Scatter(
        x=year,
        y=floor_local,
        opacity= 0.4,
        name='Floor',
        line=dict(
            color=('rgb(225, 0, 0)'),
            dash='dot'))

    line_index = go.Scatter(
        x=year,
        y=index_local * (V0/S),
        name='Index',
        line=dict(
            color=('rgb(225, 0, 0)')))  

    line_bh = go.Scatter(
        x=year,
        y=bh_local,
        name='BH',
        line=dict(
            color=('rgb(205, 12, 24)'),
            dash='dot'))

    line_cppi3 = go.Scatter(
        x=year,
        y=cppi3_local,
        name='CPPI 3',
        line=dict(
            color=('rgb(0, 225, 0)')))  

    line_cppi5 = go.Scatter(
def plot_value_through_years_mc(name, floor_local, index_local, bh_local, cppi3_local, cppi5_local, obpi_local, varpi_local, slpi_local):
    # to plot for backtesting, please indicate the attribute name with exact "Backtesting".
```python

# Create lines
floor = go.Scatter(
    x=year,
    y=floor_local,
    opacity=0.4,
    name='Floor',
    line=dict(
        color=('rgb(225, 0, 0)'),
        dash='dot'))

a = []
for i in range(len(index_local)):
    a.append(index_local[i])
line_index = go.Scatter(
    x=year,
    y=a,
    name='Index',
    line=dict(
        color=('rgb(225, 0, 0)')))

line_bh = go.Scatter(
    x=year,
    y=bh_local,
    name='BH',
    line=dict(
        color=('rgb(205, 12, 24)'),
        dash='dot'))

line_cppi3 = go.Scatter(
    x=year,
    y=cppi3_local,
    name='CPPI 3',
    line=dict(
        color=('rgb(0, 225, 0)')))

line_cppi5 = go.Scatter(
    x=year,
    y=cppi5_local,
    name='CPPI 5',
    line=dict(
        color=('rgb(0, 225, 0)')))```
line = dict(
    color=('rgb(0, 225, 0)'),
    dash = 'dash'))
line_obpi = go.Scatter(
    x=year,
    y=obpi_local,
    name='OBPI',
    line=dict(
        color=('rgb(0, 0, 225)'),
        dash='dot'))
line_varpi = go.Scatter(
    x=year,
    y=varpi_local,
    name='VaRPI',
    line=dict(
        color=('rgb(205, 12, 225)'),
        dash='dashdot'))
line_slpi = go.Scatter(
    x=year,
    y=slpi_local,
    name='SLPI',
    line=dict(
        color=('rgb(15, 20, 100)'),
        dash='dashdot'))
data = [floor, line_index, line_bh, line_obpi, line_cppi3, line_cppi5, line_varpi, line_slpi]

# Edit the layout
layout = dict(title='Performance of strategies with ' + name,
              xaxis=dict(title='Month/Year'),
              yaxis=dict(title='VND'))

fig = dict(data=data, layout=layout)
return py.plot(fig, filename='Strategies performance - ' + name)

#Function to plot histogram of the annualized mean of the portfolio insurance strategy (N = 10,000 paths)
def plot_histogram_MC(list_returns_to_plot):
    #the value of list_returns_to_plot should be as this form BH - MC1 (in the quotes)
    if list_returns_to_plot == "BH - MC1":
        name = 'BH - MC1'
    elif list_returns_to_plot == "BH - MC2":
        name = 'BH - MC2'
    elif list_returns_to_plot == "BH - MC3":
        name = 'BH - MC3'
name = 'BH - MC3'
elif list_returns_to_plot == "BH - MC4":
    name = 'BH - MC4'
elif list_returns_to_plot == "BH - MC5":
    name = 'BH - MC5'
elif list_returns_to_plot == "BH - MC6":
    name = 'BH - MC6'
elif list_returns_to_plot == "CPPI3 - MC1":
    name = 'CPPI3 - MC1'
elif list_returns_to_plot == "CPPI3 - MC2":
    name = 'CPPI3 - MC2'
elif list_returns_to_plot == "CPPI3 - MC3":
    name = 'CPPI3 - MC3'
elif list_returns_to_plot == "CPPI3 - MC4":
    name = 'CPPI3 - MC4'
elif list_returns_to_plot == "CPPI3 - MC5":
    name = 'CPPI3 - MC5'
elif list_returns_to_plot == "CPPI3 - MC6":
    name = 'CPPI3 - MC6'
elif list_returns_to_plot == "CPPI5 - MC1":
    name = 'CPPI5 - MC1'
elif list_returns_to_plot == "CPPI5 - MC2":
    name = 'CPPI5 - MC2'
elif list_returns_to_plot == "CPPI5 - MC3":
    name = 'CPPI5 - MC3'
elif list_returns_to_plot == "CPPI5 - MC4":
    name = 'CPPI5 - MC4'
elif list_returns_to_plot == "CPPI5 - MC5":
    name = 'CPPI5 - MC5'
elif list_returns_to_plot == "CPPI5 - MC6":
    name = 'CPPI5 - MC6'
elif list_returns_to_plot == "OBPI - MC1":
    name = 'OBPI - MC1'
elif list_returns_to_plot == "OBPI - MC2":
    name = 'OBPI - MC2'
elif list_returns_to_plot == "OBPI - MC3":
    name = 'OBPI - MC3'
elif list_returns_to_plot == "OBPI - MC4":
    name = 'OBPI - MC4'
elif list_returns_to_plot == "OBPI - MC5":
    name = 'OBPI - MC5'
elif list_returns_to_plot == "OBPI - MC6":
    name = 'OBPI - MC6'
elif list_returns_to_plot == "VaRPI - MC1":
    name = 'VaRPI - MC1'
elif list_returns_to_plot == "VaRPI - MC2":
    name = 'VaRPI - MC2'
elif list_returns_to_plot == "VaRPI - MC3":
    name = 'VaRPI - MC3'
elif list_returns_to_plot == "VaRPI - MC4":
    name = 'VaRPI - MC4'
elif list_returns_to_plot == "VaRPI - MC5":
    name = 'VaRPI - MC5'
elif list_returns_to_plot == "VaRPI - MC6":
    name = 'VaRPI - MC6'
elif list_returns_to_plot == "SLPI - MC1":
    name = 'SLPI - MC1'
elif list_returns_to_plot == "SLPI - MC2":
    name = 'SLPI - MC2'
elif list_returns_to_plot == "SLPI - MC3":
    name = 'SLPI - MC3'
elif list_returns_to_plot == "SLPI - MC4":
    name = 'SLPI - MC4'
elif list_returns_to_plot == "SLPI - MC5":
    name = 'SLPI - MC5'
elif list_returns_to_plot == "SLPI - MC6":
    name = 'SLPI - MC6'
else:
    print("The list of returns you choose is not available. Please read the comment in the first line of the function")

    layout = go.Layout(
        title=name,
        xaxis=dict(title='Annual mean of the portfolio strategy'),
        yaxis=dict(title='Probability'))

global figure

    if list_returns_to_plot == "BH - MC1":
        figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_bh_mc1, "mean"), histnorm='probability')],
                          layout=layout)
    elif list_returns_to_plot == "BH - MC2":
        figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_bh_mc2, "mean"), histnorm='probability')],
elif list_returns_to_plot == "BH - MC3":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_bh_mc3, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "BH - MC4":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_bh_mc4, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "BH - MC5":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_bh_mc5, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "BH - MC6":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_bh_mc6, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "CPPI3 - MC1":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_cppi3_mc1, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "CPPI3 - MC2":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_cppi3_mc2, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "CPPI3 - MC3":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_cppi3_mc3, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "CPPI3 - MC4":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_cppi3_mc4, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "CPPI3 - MC5":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_cppi3_mc5, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "CPPI3 - MC6":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_cppi3_mc6, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "CPPI5 - MC1":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_cppi5_mc1, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "CPPI5 - MC2":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_cppi5_mc2, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "CPPI5 - MC3":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_cppi5_mc3, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "CPPI5 - MC4":

```python
figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_cppi5_mc4, "mean"), histnorm='probability')],
layout=layout)
elif list_returns_to_plot == "CPPI5 - MC5":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_cppi5_mc5, "mean"), histnorm='probability')],
layout=layout)
elif list_returns_to_plot == "CPPI5 - MC6":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_cppi5_mc6, "mean"), histnorm='probability')],
layout=layout)
elif list_returns_to_plot == "OBPI - MC1":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_obpi_mc1, "mean"), histnorm='probability')],
layout=layout)
elif list_returns_to_plot == "OBPI - MC2":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_obpi_mc2, "mean"), histnorm='probability')],
layout=layout)
elif list_returns_to_plot == "OBPI - MC3":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_obpi_mc3, "mean"), histnorm='probability')],
layout=layout)
elif list_returns_to_plot == "OBPI - MC4":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_obpi_mc4, "mean"), histnorm='probability')],
layout=layout)
elif list_returns_to_plot == "OBPI - MC5":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_obpi_mc5, "mean"), histnorm='probability')],
layout=layout)
elif list_returns_to_plot == "OBPI - MC6":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_obpi_mc6, "mean"), histnorm='probability')],
layout=layout)
elif list_returns_to_plot == "VaRPI - MC1":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_varpi_mc1, "mean"), histnorm='probability')],
layout=layout)
elif list_returns_to_plot == "VaRPI - MC2":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_varpi_mc2, "mean"), histnorm='probability')],
layout=layout)
elif list_returns_to_plot == "VaRPI - MC3":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_varpi_mc3, "mean"), histnorm='probability')],
layout=layout)
elif list_returns_to_plot == "VaRPI - MC4":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_varpi_mc4, "mean"), histnorm='probability')],
layout=layout)
elif list_returns_to_plot == "VaRPI - MC5":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_varpi_mc5, "mean"), histnorm='probability')],
layout=layout)
```
elif list_returns_to_plot == "VaRPI - MC6":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_varpi_mc6, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "SLPI - MC1":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_slpi_mc1, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "SLPI - MC2":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_slpi_mc2, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "SLPI - MC3":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_slpi_mc3, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "SLPI - MC4":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_slpi_mc4, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "SLPI - MC5":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_slpi_mc5, "mean"), histnorm='probability')], layout=layout)
elif list_returns_to_plot == "SLPI - MC6":
    figure = go.Figure(data=[go.Histogram(x=function_list_perf_measure(returns_slpi_mc6, "mean"), histnorm='probability')], layout=layout)

return py.plot(figure, filename="Histogram_returns_" + name)

################## Print the results
# Tables (Already done above)

# Graphs of one out of 10000 paths of Monte Carlo simulation for 6 market scenarios and Backtesting method

print(plot_value_through_years_mc("MC1",floor_line, vn30eq_monthly_mc(mc1),bh_mc1[:,0],cppi3_mc1[:,0],cppi5_mc1[:,0],obpi_mc1[:,0],varpi_mc1[:,0], slpi_mc1[:,0]))
print(plot_value_through_years_mc("MC2",floor_line, vn30eq_monthly_mc(mc2),bh_mc2[:,0],cppi3_mc2[:,0],cppi5_mc2[:,0],obpi_mc2[:,0],varpi_mc2[:,0], slpi_mc2[:,0]))
print(plot_value_through_years_mc("MC3",floor_line, vn30eq_monthly_mc(mc3),bh_mc3[:,0],cppi3_mc3[:,0],cppi5_mc3[:,0],obpi_mc3[:,0],varpi_mc3[:,0], slpi_mc3[:,0]))
print(plot_value_through_years_mc("MC4",floor_line, vn30eq_monthly_mc(mc4),bh_mc4[:,0],cppi3_mc4[:,0],cppi5_mc4[:,0],obpi_mc4[:,0],varpi_mc4[:,0], slpi_mc4[:,0]))
print(plot_value_through_years_mc("MC5",floor_line, vn30eq_monthly_mc(mc5),bh_mc5[:,0],cppi3_mc5[:,0],cppi5_mc5[:,0],obpi_mc5[:,0],varpi_mc5[:,0], slpi_mc5[:,0]))
print(plot_value_through_years_mc("MC6",floor_line, vn30eq_monthly_mc(mc6),bh_mc6[:,0],cppi3_mc6[:,0],cppi5_mc6[:,0],obpi_mc6[:,0],varpi_mc6[:,0], slpi_mc6[:,0]))
print(plot_value_through_years("Backtesting",floor_line, vneq30_monthly["Index"],bh_bt,cppi3_bt,cppi5_bt,obpi_bt,slpi_bt))
# Histograms

print(plot_histogram_MC("BH - MC1"))
print(plot_histogram_MC("BH - MC2"))
print(plot_histogram_MC("BH - MC3"))
print(plot_histogram_MC("BH - MC4"))
print(plot_histogram_MC("BH - MC5"))
print(plot_histogram_MC("BH - MC6"))
print(plot_histogram_MC("CPPI3 - MC1"))
print(plot_histogram_MC("CPPI3 - MC2"))
print(plot_histogram_MC("CPPI3 - MC3"))
print(plot_histogram_MC("CPPI3 - MC4"))
print(plot_histogram_MC("CPPI3 - MC5"))
print(plot_histogram_MC("CPPI3 - MC6"))
print(plot_histogram_MC("CPPI5 - MC1"))
print(plot_histogram_MC("CPPI5 - MC2"))
print(plot_histogram_MC("CPPI5 - MC3"))
print(plot_histogram_MC("CPPI5 - MC4"))
print(plot_histogram_MC("CPPI5 - MC5"))
print(plot_histogram_MC("CPPI5 - MC6"))
print(plot_histogram_MC("OBPI - MC1"))
print(plot_histogram_MC("OBPI - MC2"))
print(plot_histogram_MC("OBPI - MC3"))
print(plot_histogram_MC("OBPI - MC4"))
print(plot_histogram_MC("OBPI - MC5"))
print(plot_histogram_MC("OBPI - MC6"))
print(plot_histogram_MC("VaRPI - MC1"))
print(plot_histogram_MC("VaRPI - MC2"))
print(plot_histogram_MC("VaRPI - MC3"))
print(plot_histogram_MC("VaRPI - MC4"))
print(plot_histogram_MC("VaRPI - MC5"))
print(plot_histogram_MC("VaRPI - MC6"))
print(plot_histogram_MC("SLPI - MC1"))
print(plot_histogram_MC("SLPI - MC2"))
print(plot_histogram_MC("SLPI - MC3"))
print(plot_histogram_MC("SLPI - MC4"))
print(plot_histogram_MC("SLPI - MC5"))
print(plot_histogram_MC("SLPI - MC6"))

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
END OF CODES ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
REFERENCES


Minister of Finance. (2011, 03 28). *The Circular about the rates of collection, the regime of collection, remittance, management and use of fees in the field of securities applicable securities business organizations*. Retrieved from Minister of Finance: http://www.mof.gov.vn/


