A Work Project, presented as part of the requirements for the Award of a Master's degree in

Finance from the Nova School of Business and Economics.

Banco Invest Consulting Project
«Delta-Gamma Value-at-Risk model for - »

Sven Yannik Peters (49692) - Portfolio of Autocall options

Work project carried out under the supervision of:

Daniele D'arienzo

11/01/2022

Abstract (100 words maximum)

Banco Invest offers various over-the-counter (OTC) derivatives to institutional clients as part of its structured investment solutions. These derivatives are managed within the bank's Proprietary Trading Book. The focus of this consulting project is developing a Delta-Gamma Value-at-Risk (VaR) model that Banco Invest can implement to actively manage its equity derivative portfolio`s underlying risks. The first part contains the estimation of the portfolio delta and gamma. The second part consists of the quadratic approximation to calculate the portfolio standard deviation. In the last section, the authors calculate the Delta-Gamma Value-at-Risk and provide recommendations to Banco Invest.

Keywords: Value-at-Risk, Autocall option, Portfolio Delta, Portfolio Gamma, Delta-Gamma Value-at-Risk

This work used infrastructure and resources funded by Fundação para a Ciência e a Tecnologia (UID/ECO/00124/2013, UID/ECO/00124/2019 and Social Sciences DataLab, Project 22209), POR Lisboa (LISBOA-01-0145-FEDER-007722 and Social Sciences DataLab, Project 22209) and POR Norte (Social Sciences DataLab, Project 22209).

Contents

1 Value-at-Risk – Group part 4

1.1 Defining Value-at-Risk 4

1.2 Pitfalls and limitations of Value at Risk 6

2 The “Greeks” - Group part 8

2.1 Delta Risk 8

2.2 Gamma Risk 9

2.3 Hedging the Greeks 11

3 Value-at-Risk for a Derivatives Portfolio - Group part 12

4 Methodology used in Python - Group part 15

5 Autocall option – Individual part (Yannik Peters) 18

5.1 Portfolio Delta 22

5.2 Portfolio Gamma 24

5.3 Non-linear Delta-Gamma-VaR 25

10 Recommendation - Group part 27

Bibliography 29

Appendix 31

1 Value-at-Risk – Group part

Market risk describes the risk of a possible loss in a risk position due to collective adverse movements of market rates and prices. It is one of the most critical risks for institutions that actively trade in financial markets; quantifying and monitoring this risk is crucial for allocating capital and reserves needed to cover potential losses and assess their overall solvency. Market risks are determined by institutions using standard procedures or internal risk models; one of these procedures is the Value-at-Risk model. (Deutsche Bundesbank 2022)

1.1 Defining Value-at-Risk

The Value-at-Risk expresses the maximum potential loss, in absolute terms or as a percentage in the respective currency the asset is held, that results under normal market conditions from an adverse movement in the relevant market of an investment over a specified time horizon (H) at a given degree of confidence (α) during a fixed holding period of a risk position. The estimated maximum potential loss of the model, the VaR estimate, is only expected to be exceeded (1-α) % of the time. (Castellacci and Siclari 2003, pp. 531-532) (Fallon 1996, p. 2) The time horizon of interest for a VaR estimate can be one day or even months and is determined by the nature of the portfolio. The horizon should correspond to the most prolonged period needed for an orderly liquidation or the time to hedge an investment portfolio. (Bodie, Kane, and Marcus 2021, p. 138) The VaR estimate's horizon is determined by the liquidity profile of the assets in the underlying investment portfolio; the length relates to the time needed to sell these assets at average transaction volumes so that they have little impact on the market. Since the market impact of the liquidation scenario is not disregarded when choosing the horizon, the VaR estimate will be an estimate of a realizable loss and not only a loss on paper. (Wilmott 1998, p. 548) The confidence (α) level for a VaR estimate corresponds to the institution's risk profile, determined by its degree of risk aversion or regulatory requirements. (Fallon 1996, p. 2)

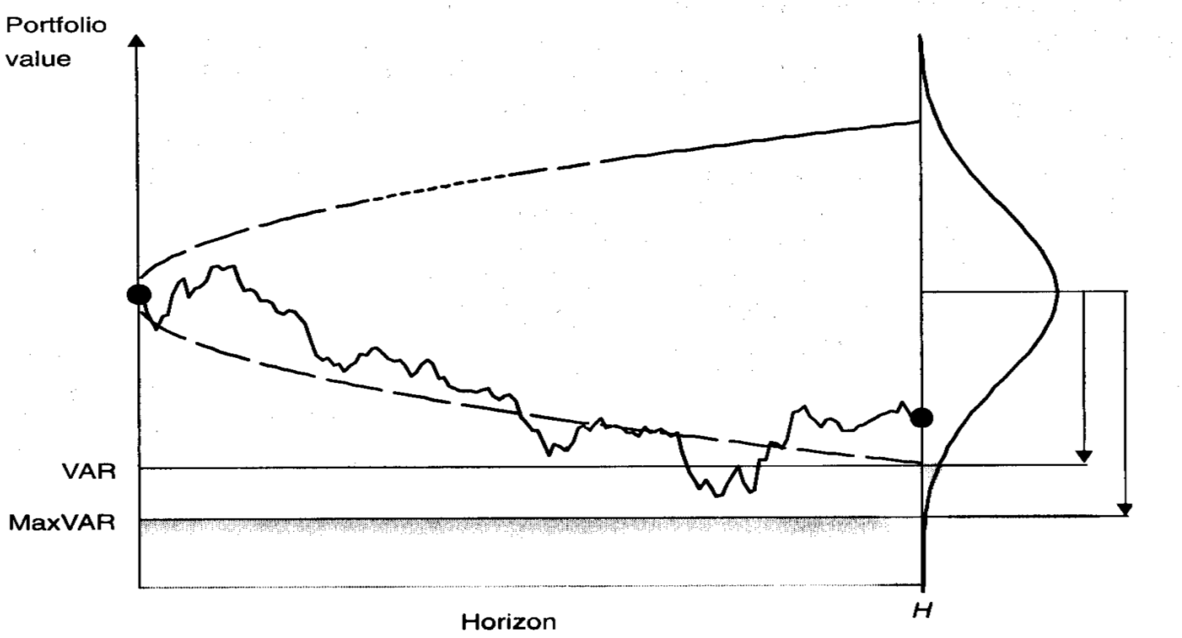


Figure 1: Development of VaR over time horizon H (Jorion 2007, p. 118)

A VaR calculation applies to all types of risky assets and can be applied to a single position and a whole portfolio of risky assets. Assessing VaR helps institutions evaluate the profitability of an investments in relation to the risk and identify investments with a higher-than-acceptable risk profile, allowing them to make changes or liquidate such investments. The VaR is used for active and passive risk measurement and defensive risk control. Ideally, it suits financial and non-financial institutions that engage in proprietary trading with significant exposure to market risks. (Jorion 2007, pp. 379-389) VaR estimates typically focus on 'tail events' where liquidity and large jumps are essential, as illustrated in ***Appendix 1*** below. (Wilmott 1998, p. 337) Therefore, confidence levels are typically set at 95%, 97.5%, and 99%. (Wilmott 1998, p. 547) An overview of which confidence levels translate into which z statics of the confidence interval can be found in ***Appendix 2***. The VAR statistic on portfolio losses is defined as a one-sided confidence interval:

$$Prob\ [\Delta\tilde{P}(\Delta t, \Delta\tilde{x}) > \ -VAR] = 1 - \alpha \tag{1}$$

In the above equation, $\Delta\tilde{P}(\Delta t, \Delta\tilde{x})$ stands for the change in the value of a portfolio that results from a function consisting of the forecasting period Δt and the vector $\Delta\tilde{x}$ of the random variables, with α being the confidence level. The equation can be interpreted as the portfolio's value will not fall by more than VAR over Δt number of trading days with α % confidence. (Fallon 1996, p. 2) The degree of complexity and the computational requirements of the calculation of a VaR estimate depends in particular on how the price of the instrument changes in relation to the underlying. ***Appendix 3*** depicts the two different relationships. (Romano 2017) The calculation of a VaR estimate for non-linear (i.e., derivatives) assets is more complex than for a linear asset (i.e., a stock or bond). In the context of an option: nonlinearity implies that a price movement in the underlying asset causes a non-linear change in the option price. There are three major methodologies to calculate Value-at-Risk, the historical approach, the parametric or model-building approach, and performing a Monte Carlo simulation. ***Figure 2*** below provides an overview of the different methodologies and their advantages and disadvantages. (Hull 2021, pp. 293-297 & 317-340)

Type	Description	Advantages	Disadvantages
Historical	Estimates VaR using past distribution of returns to predict future returns	- Easy way to calculate VaR - Takes into account possible skeweness and fat tails - Accurate for non-linear products - No distributional assumptions necessary	- Assumes future returns dependend on the past (impractical) - Large amount of daily rate history required - Slow reaction to recent market events
Parametric	Estimates VaR using prespecified variables (volatility & correlation)	- Quick and easy to compute - Accurate for simple & linear products	- Assumption of normal distribution impractical - Less quick and accurate for non-linear derivatives
Monte Carlo	Estimates VaR by simulating random scenarios	- Accurate for linear & non-linear products - Flexibility to choose different distributions - Flexibilty on the choice of variables - Outputs full distribution of potential product values	- Massive computational power required to revalue the portfolio in each scenario - Accuracy dependend on number of simulation performed

Figure 2: Overview of different approaches for VaR calculation (Hull 2021, pp. 293-297 & 317-340)

1.2 Pitfalls and limitations of Value at Risk

Despite the widespread use of the Value-at-Risk model, it has several drawbacks that will be briefly discussed in the following. First and foremost, all methods require making assumptions and using them as inputs for the mode; this can result in different outcomes even if the same modelling approach is used. Assumptions have to be made, e. g. about the applicable horizon and confidence level and the appropriate number of simulations. (Jorion 2007, pp. 542-557) Furthermore, all methods rely to some extent on historical data as a proxy to forecast future estimates. What has happened in the past does not necessarily imply that it will happen again in the future, so that estimation can be Inaccurate. (Jorion 2007, pp. 542-557) Second, there is yet to be an industry-wide standard to model VaR. The different approaches and models to calculate VaR can also lead to different estimates for the same portfolio. Hence, the correct interpretation is vital. (Jorion 2007, pp. 542-557) This brings us to the next limitation: a VaR estimate is calculated assuming normal market conditions, meaning extreme and rare events, such as so-called black swans, are not considered by the estimate. Because VaR only allows the risk manager to make statements about which value will not be exceeded with what degree of certainty, it does not tell anything about the worst outcome in case the VaR number is ex (Hull 2018, pp. 273-274) Additionally, the traditional VaR disregards intervening losses. These occur when the portfolio's value falls below VaR during the time horizon but eventually rises above it at the end of it. This can be an essential aspect for management if the portfolio is marked to market daily and faces potential margin calls that could result in liquidation in the worst-case scenario. (Jorion 2007, pp. 117-119) A VaR estimate provides the "big picture" of what is at risk regarding market risk effects. However, as it only accounts for this specific risk type, it has a narrow focus on what is really at risk. There are also risks which are not incorporated in the VaR framework, commonly referred to as "risks not in Value-at-risk" (RNIV): This can result in the actual Value at Risk of an investment being much higher than what the VaR model is predicting when capturing many of the other existing risk variables such as (geo-)political risks, liquidity risks, and regulatory risk. (Jorion 2007, pp. 542-557)

2 The “Greeks” - Group part

In option pricing, as well as for other derivatives, the "Greeks" are commonly used to measure the sensitivity of a derivative's value to factors that might affect the price of an options contract. ***Appendix 4*** gives an overview of the existing Greeks and their definitions. (Leoni 2014) Within the frame of this work, the focus will be set on two risk metrics, delta (Chapter 3.1) and gamma (Chapter 3.2) risk, in relation to option pricing, as they are the most fundamental.

2.1 Delta Risk

The delta, designated with the symbol Δ, is the first-order partial derivative of the option pricing function c with respect to the underlying asset S. Therefore, it expresses the sensitivity of the option contract’s price to changes in the price of the underlying asset while leaving all else constant (ceteris paribus). (Taleb 1997, p. 224) (Bouzoubaa and Osseiran 2010, p. 66)

$$\Delta = \frac{\partial c}{\partial S} \tag{2}$$

For vanilla options, the delta for long calls and short puts on standard options varies between 0 and 1. Vice versa, short calls and long puts have a delta ranging between 0 and -1. Graphically expressed is it the slope of the curve that links the option price to the underlying asset price. The higher the slope, the higher the delta and the more the derivative contract will change in response to price fluctuations of the underlying asset. ***Figure 3*** below depicts the change in delta with respect to the Strike price K and the time to maturity T for a European call option. With the option increasingly getting out of the money (OTM), a higher Strike K, and/or the option approaching its maturity date T, the delta tends to move towards 0. Conversely, with lower Strike K, the option being more in the money (ITM), and/or longer time until maturity T, delta approaches 1. (Hilpisch 2015, p. 78) The most significant change in delta can be observed with the option being at the money (ATM), S = K, close to its maturity date T. This is because theoretically, with the option being ATM a few seconds before it matures, one small move in either direction would result in the option being either in the money or out of the money, hence the considerable variation in delta. (Hilpisch 2015, p. 78)

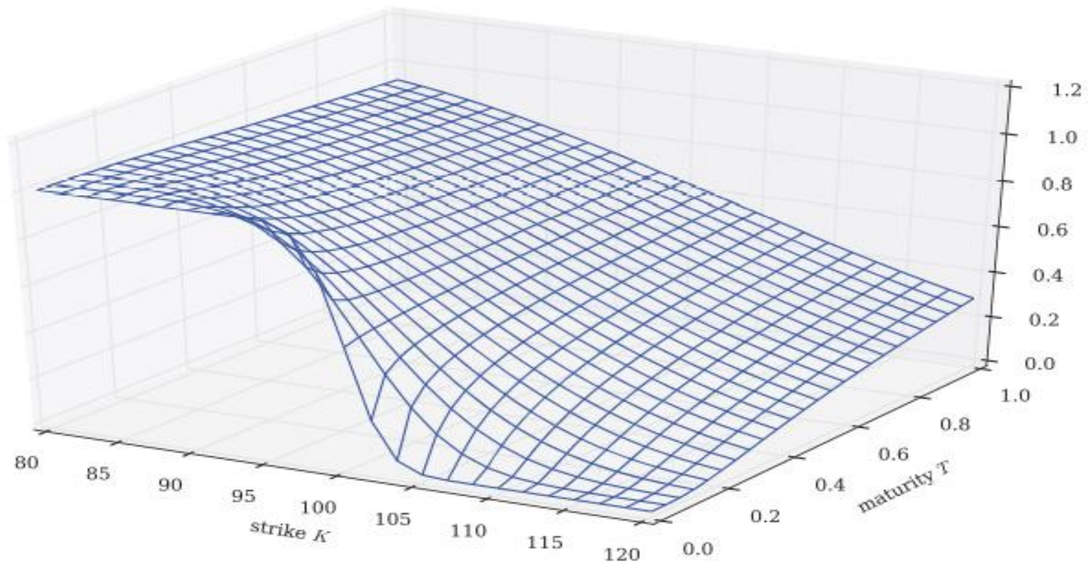


Figure 3: Delta of a European Call Option (Hilpisch 2015, p. 78)

Delta risk can be hedged to obtain a neutral position ($\Delta = 0$). How this can be achieved for a portfolio of derivatives will be explained in more detail in section 3.3, Hedging the Greeks.

2.2 Gamma Risk

For minor variations in the price of the underlying asset, delta proves to be good at estimating the change in the option's price. However, as soon as price changes become more severe, delta is extremely sensitive to changes in the underlying asset's price. This is because delta graphically represents a linear estimate for a non-linear option function. Hence, the actual option value might significantly differ from the proportion predicted by delta. (de Weert 2008, pp. 14-16) Gamma, Γ, measures by how much or how often a position or a portfolio of options needs to be re-hedged to maintain a delta-neutral position: it expresses by how much the Delta might change if the price of the underlying changes. It is the second-order derivative of the option pricing function c with respect to the underlying asset S.

$$\Gamma = \frac{\partial^2 c}{\partial^2 S} \tag{3}$$

The more curvature the option function entails, the higher the gamma and the more sensitive the delta is towards changes in the underlying's price. An increase in the underlying's price could significantly increase the delta and vice versa for a low gamma. Considering plain vanilla options, the gamma is always positive for long positions, whereas for short positions, it is negative. (Bouzoubaa and Osseiran 2010, p. 72) ***Figure 4*** below shows that the gamma value is stable for most of the option's life as it hovers near zero. The most notable value changes in gamma happen around ATM options close to maturity. As previously stated in the preceding section, it is for at-the-money options close to maturity where one move in either direction has the most significant influence on delta as it determines whether the option is exercised. Hence, the high value in gamma. (Yen Jerome and Lai 2015, pp. 84-85).

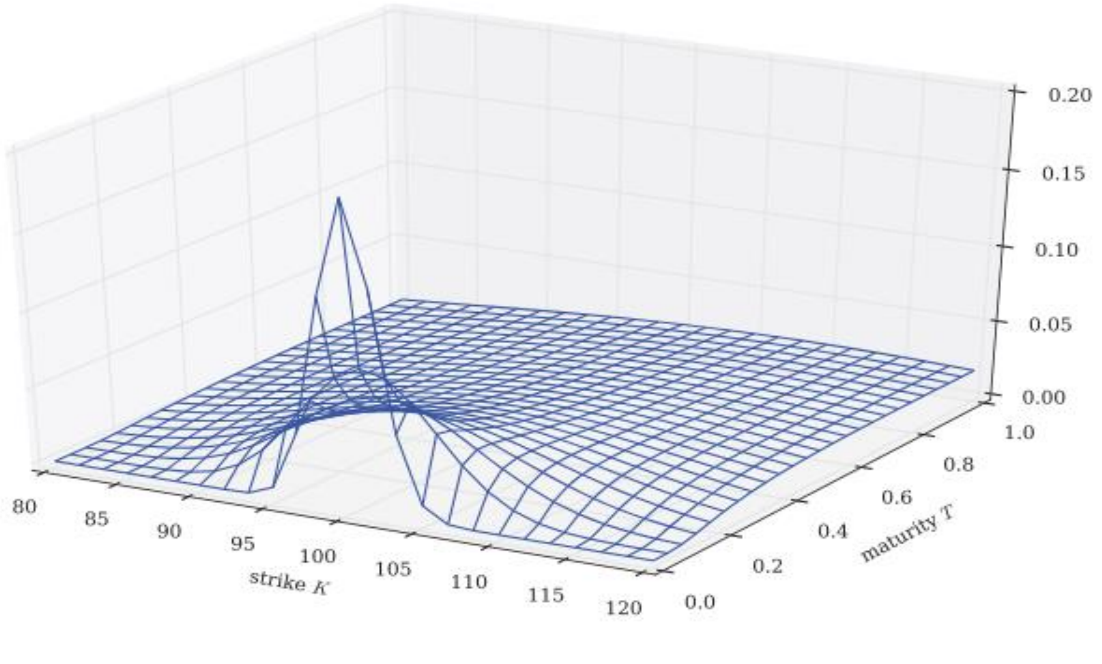


Figure 4: Gamma of a European Call option (Hilpisch 2015, p. 79)

How gamma is incorporated when hedging the respective portfolio's VaR will be explained in more detail in the next section.

2.3 Hedging the Greeks

As previously described, a portfolio's sensitivity to such is captured by the "Greek letters". The risk framework captures thresholds for each to ensure that these risks stay within the company's tolerance. Exceeding the limits initializes a process known as hedging. This is where counter positions in the market are established to ensure that the exposure to a particular risk factor stays within its predefined limit. In the following, it will be presented how a portfolio is hedged against delta and gamma. (Hull 2018, p. 161) Hedging delta consists of establishing a counter position equal to Δ amount of the underlying. By combining the existing portfolio and the hedging trade, the new portfolio's exposure to delta is neutralized. (Hull 2018, pp. 161-162) For linear products, hedging delta turns out to be static as it protects against both small and large changes in the value of the underlying. Further, once a linear hedge is implemented, there is no need to adjust it over time. The delta for a linear portfolio stays constant. (Hull 2018, pp. 163-164) Neutralizing delta exposure for non-linear products such as options proves to be a more complex procedure due to the non-linear relationship between the price of the underlying and the options contract. As mentioned earlier in this work, eliminating a portfolio's delta only offers protection from small fluctuations in the price of the underlying. Additionally, once it is set up, the delta hedge has to be adjusted frequently, also known as dynamic hedging or "rebalancing". This is because Delta constantly evolves throughout a non-linear product's lifetime. (Hull 2018, pp. 165-168) In practice, rebalancing is costly as, e.g., hedging a long position on an option involves buying the underlying when its price increased and selling it when it dropped to consistently create a synthetical position opposite of that to neutralize the option's delta. This is usually reflected in the premiums that option buyers have to pay. (Hull 2018, p. 169) With more significant changes in the prices of the underlyings, a portfolio's gamma comes into play. There are two ways of adjusting for the additional gamma exposure of a non-linear portfolio that will be briefly described below. (Hull 2018, pp. 169-170) Firstly, the portfolio is made gamma neutral by trading options with opposite gammas on the same underlyings as the options in the existing portfolio. Non-linear products are needed as linear products do not have exposure to gamma. By doing this, the new and combined portfolio's delta also changes and would have to be re-adjusted by trading opposite positions in the underlyings (Hull 2018, pp. 170-171) Implementing this in practice can be challenging as trading non-linear derivatives in the amounts needed often is impossible. Further, re-adjusting for the new delta of the combined portfolio is costly as it involves many transactions. (Hull 2018, p. 177) However, as described earlier, it makes economically more sense to see the gamma as a determinant of how often a portfolio needs to be re-hedged. In general, a portfolio with larger gamma would imply more frequent delta neutralization, whereas a smaller gamma results in less often adjustments to the portfolio, as changes in delta only tend to be small. (Hull 2018, pp. 169-170) Banco Invest hedges its equity derivatives portfolio with underlyings (delta neutralization) rather than options (gamma neutralization). The Bank does not take directional market risk, keeping the difference between the deltas (theoretical quantities) and the quantities held in the portfolio as close to zero as possible. These portfolio quantities are adjusted daily, at 30-minute intervals, based on market conditions, namely the evolution of the underlying shares.

3 Value-at-Risk for a Derivatives Portfolio - Group part

To begin with, calculating Value-at-Risk for a single asset is a straightforward process. Assuming linearity in the change of the portfolio's value to changes in the underlying and normally distributed returns, VaR is calculated as follows:

$$VaR = w_i S_i \left(\mu \, \delta t - \sigma_i \, (\delta t^{\frac{1}{2}}) \, \alpha(1-c) \right) \tag{4}$$

where w_i is the quantity of the asset i owned with price S_i. This is multiplied by the asset's drift over a predefined time horizon δt, with $\alpha(1-c)$ being the inverse cumulative distribution function of the standard normal distribution. This process is called delta approximation. (Wilmott 1998, pp. 548-550) Regarding a portfolio of assets, the calculation of VaR becomes more complex. First, the volatilities and covariances of all assets in the portfolio have to be computed. If this is done, the formula to calculate the VaR of a portfolio with M assets consisting of w_i amount of asset i and w_j amount of asset j is:

$$VaR_{Portfolio} = -M\left(\alpha(1-c)(\delta t^{\frac{1}{2}})\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{M} w_i w_j \sigma_i \sigma_j \rho_{ij}}\right) \tag{5}$$

with σ_i being the volatility of asset i and ρ_{ij} the correlation between asset i and j. (Wilmott 1998, pp. 551) Estimating VaR for a portfolio of derivatives, as mentioned earlier, the delta approximation would only be sufficient for portfolios where the underlyings show small movements in price. This is because the relationship between the portfolio's value and price changes in the underlyings can no longer be regarded as linear. For non-linear portfolios, the sensitivity to gamma additionally has to be considered. This is visually demonstrated in ***Figure 5*** below. It depicts the relationship between the price of an underlying asset to the corresponding value of a long call option on the same. While the underlying's price function is normally distributed, the option has a positively skewed probability distribution with a smaller tail on the left. (Hull 2018, pp. 333-334) This violates the initial premise that probabilities are normally distributed. If VaR were calculated based on this assumption, it would be excessively high. As a result, approximations for the portfolio's sensitivity to changes in the underlyings need to be reevaluated. (Wilmott 1998, pp. 550-551)

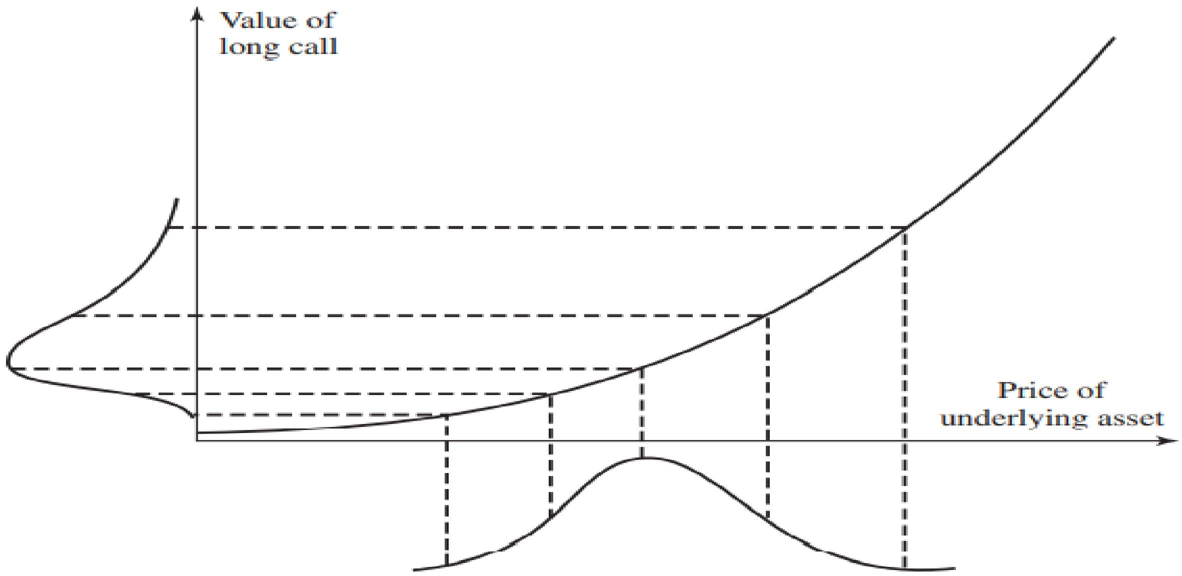


Figure 5: Translation of an Asset's normal probability distribution into that of a long call option (Hull 2018, p. 333)

To recapture, with larger swings in the prices of the underlyings of an options portfolio, the previous delta approximation to calculate VaR turns out to be inappropriate. A better estimation is achieved by incorporating the portfolio's sensitivity to gamma. Gamma exposure is particularly challenging as a second-order approximation is required. (Wilmott 1998, p. 551) This will be shown below. Assume a portfolio M consisting of a single option on an asset with price S. The change in the value of the portfolio δM compared to changes in the price of the underlying δS can be expressed as follows:

$$\delta M = \frac{\partial P}{\partial S}\delta S + \frac{1}{2}\frac{\partial^2 P}{\partial S^2}(\delta S)^2 + \frac{\partial P}{\partial \sigma}\delta t + \cdots \tag{6}$$

This can ultimately be reformulated into:

$$\delta M = \Delta\sigma S\ \delta t^{\frac{1}{2}}\,\phi + \delta t\left(\Delta\mu S + \frac{1}{2}\Gamma\sigma^2 S^2\phi^2 + \Theta\right) + \cdots \tag{7}$$

where Θ is the time drift of the option (Theta). (Wilmott 1998, p. 551) The quadratic term, the portfolio's exposure to gamma, is of specific interest above. ***Figure 6*** shows three different distribution functions. The distribution of the underlying with a standard deviation of $\sigma S\ \partial t^{\frac{1}{2}}$ is considered to be normal. The projected distribution for the change in the value of the options portfolio according to the delta approximation. It is normally distributed with a standard deviation of $\Delta\sigma S\ \partial t^{\frac{1}{2}}$. Finally, the options portfolio's distribution using the delta-gamma approximation. (Wilmott 1998, pp. 551-552)

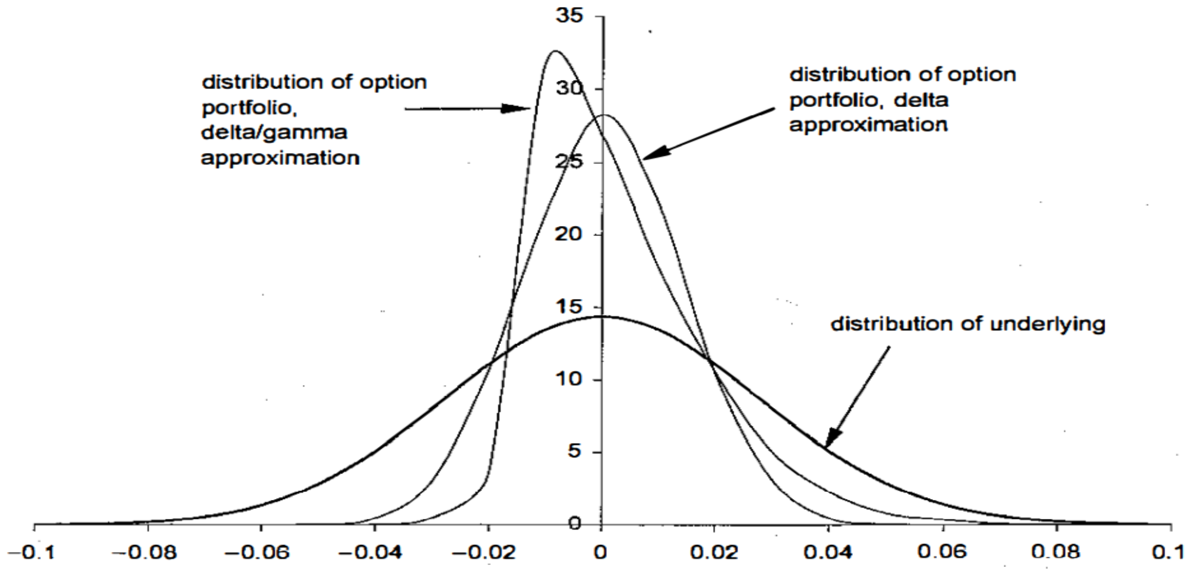


Figure 6: Relationship of an asset price's normal distribution to the distribution of an option portfolio according to the delta as well as the delta-gamma approximation (Wilmott 1998, p. 552)

By looking at the three different distributions, it is evident that the one for the delta-gamma approximation is not normally distributed compared to the other two. (Wilmott 1998, pp. 551-552)

4 Methodology used in Python - Group part

In the following, the Assumptions used to calculate the Delta-Gamma VaR in Python, as well as the fundamental parts of the code, are presented and explained. As the basis for all calculations of the various input statistics of the VaR model, the authors assume one year consisting of 252 trading days. Because of their ease of use for time series modelling, such as symmetry, time-additivity, and the log-normal distribution assumption, the various underlyings performances are transformed into logarithmic returns. Next, **e**ach option's volatility is calculated using equally weighted implied volatilities of the option's underlyings. In the absence of implied volatility, the underlying's historical volatility on a 30-day basis is used. Furthermore, to determine the correlation, variance, and covariance of the different underlyings, a maximum lookback window of 2 years is assumed, the same as the option's time to maturity on the trade date. From there on, for each day that has progressed, the option's remaining time to maturity is used to calculate the above statistics until a predefined minimum of 30 days was reached. Below this, correlation, variance, and covariance are calculated on a 30-day basis until the option matures. At this point, it is referred to ***Appendix 5-6*** for the code example. The options in Banco Invest's portfolio are valued as of **30/06/2022** using Monte Carlo simulations. The first step of Monte Carlo involved calculating the geometric Brownian Motion. In finance, this is a stochastic process to model random behavior over a specific time frame (δt) that consists of two main components, drift, and a randomly generated variable. (Yan 2017, pp. 421-428) Drift indicates the direction of an asset's historical returns, allowing predictions on an asset's expected return. It is calculated as shown in ***equation (8)*** using the same receding time horizon as explained for the underlying's statistics, except for the time series' minimum requirement of 30 days.

$$Drift = \left(Mean\ (stock\ returns) - \frac{Variance\ (stock\ returns)}{2} \right) *\ \delta t \tag{8}$$

Where underlyings are expected to pay dividends, the drift is adjusted further, as demonstrated in ***Appendix 7***. The next step is to obtain a random number by multiplying an asset's historical standard deviation with a random, standard normally distributed variable *(Z ([Rand (0;1)]))*.

$$Random\ variable = \left(Std.Dev.\ *\ Z([Rand(0;1)])\right) * \sqrt{\delta t} \tag{9}$$

As a result, the equation for predicting the future value of an asset *(S_{t+1})* sums up to the following:

$$S_{t+1} = S_t * e^{Drift\ +\ Random\ variable} \tag{10}$$

However, when pricing options comprised of baskets of underlyings, Cholesky Decomposition is performed as an extension of the Monte Carlo simulation to account for the correlation aspects between the various reference assets. A brief explanation of an example decomposition will be provided below. ***Appendix 8*** contains the code for the Cholesky decomposition performed for the different options. Assume a $2 * 2$ symmetric, positive definite correlation matrix Σ, where ρ is the correlation between ***X_1*** and ***X_2***.

$$\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \tag{11}$$

The correlation matrix can then be decomposed into a $2 * 2$ lower triangular matrix L, where $LL^T = \Sigma$. (Wilmott 1998, pp. 682-683) This appears to be as follows:

$$L = \begin{pmatrix} 1 & 0 \\ \rho & \sqrt{1-\rho^2} \end{pmatrix} \tag{12}$$

Following the generation of L, the random variables with desired correlation can be expressed as LZ, where Z is a column vector of the independent standard normal random variables:

$$Z = \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \tag{13}$$

As a result, by setting $XL = Z$, we can sample from a bivariate normal distribution, indicating that: (Yen Jerome and Lai 2015, pp. 99-100)

$$X_1 = Z_1 \tag{14}$$

$$X_2 = \rho * Z_1 + \sqrt{1-\rho^2} * Z_2 \tag{15}$$

To generate a sufficient sample of possible future asset values for the different underlyings to calculate the option's payoffs appropriately, 200.000 simulations are run. Following this, the averaged payoffs are discounted using the respective's maturity Euribor 3-month forward. Where no forward for the maturity of the option's payoffs is readily available, linear interpolation is performed to compute the discount rate for the respective maturity's payoff, as shown in ***Appendix 9-10***. Further, each underlying's delta is estimated by changing its price by 1%, while leaving the other's prices constant, and calculating the new price of the option. The difference in both derivative prices is then divided by the relative changes in the prices of the underlying. The option's delta is estimated as the weighted average of the underlying's deltas, assuming an equally weighted portfolio of underlyings. To calculate gamma, the above calculation is done a second time to get the change in delta. The difference in both deltas is then divided by the relative adjustment to obtain the gamma value. The equations used and the respective code for this can be found in ***Appendix 12-26***. In terms of VaR, the confidence level was set to 99,9 %. Calculations are performed initially for a one-day time horizon and then later multiplied by the square root of 252 to get the annualized VaR, as this is the requirement from the risk management department at Banco Invest. Detailed calculations performed for this in Python can be found in ***Appendix 26***.

5 Autocall option – Individual part (Yannik Peters)

Autocallables are an exotic barrier option that automatically triggers an event if prespecified conditions are satisfied. Gaining popularity after the 2008 financial crisis, they offer investors the possibility of yield enhancement in a low-yield environment based on the performance of underlying assets such as equities, commodities, or currencies, whether as a single asset or a basket of reference assets. (Guillaume 2015b, pp. 1-3) With this option, investors have the possibility to earn a higher coupon than currently offered in the market or face the risk of not receiving any coupon at all. This is contingent on the underlyings crossing a so-called coupon barrier level, and as the name already indicates, these options are automatically exercised once the autocall barrier is concurrently broken by the basket of underlying assets. Characteristics for Autocallables vary across the market. While some have differing coupon and autocall thresholds, others have them set to the same. A separate autocall and coupon barrier level implies that if the underlyings cross the latter but stay below the autocall threshold, a coupon is paid to the investor while the option remains in place and proceeds to the next observation date.

Only in case the autocall barrier is crossed, the option is exercised, and the investor receives the principal amount invested plus a coupon. In a scenario where both thresholds are equal, upon crossing the autocall level, the investor receives the initial amount invested and the coupon and the option cease to exist. (Z. Tong 2019, pp. 440-441) Autocalls can further be differentiated in being capital protected, investor's initial investment is guaranteed to be paid back at maturity, or not. The latter, also called down-and-in feature, results in the investor's redemption amount being linked to the worst-performing stock of the basked of underlyings. If the underlying's value declines by more than x %, the investor will face losses to the initial investment amount. (Bouzoubaa and Osseiran 2010, pp. 198-202) In terms of observation dates, the trigger levels are either observed continuously or discrete, meaning that for the latter, the underlyings performances are assessed on prespecified dates. If no early redemption occurs at an observation date, the option proceeds to the next date, where it is again assessed. Continuous Autocallables will be exercised at any time during their lifespan once the underlyings simultaneously cross the autocall barrier. This implies that discrete Autocallables are less likely to be called than continuous ones, leaving everything else constant. (Deng, Mallett, and McCann 2011, pp. 327-328) Some Autocallables come with an embedded memory function for coupon payments. A memory function guarantees that an investor receives all past coupons not paid on previous observation dates if all of the underlyings are above the autocall barrier on subsequent observation dates. (Guillaume 2015, pp. 73-74) The Autocallables offered by Banco Invest, and at the same time focus of this work, are capital-protected multi-asset equity options. The options in the bank's portfolio have a maximum maturity of two years with discrete, semi-annual coupon dates. While the potential coupon payments are designed to increase gradually with successive observation dates, Banco Invest's options do not have an embedded memory function. Autocall and coupon barrier are equal at 100 % of the strike price, meaning that only in the event of an Autocall does the investor receive the coupon plus the initial investment. ***Table 1*** shows the different possible payoff scenarios for an example Autocall option. Under the assumption that one year consists of 252 trading days, the first observation date is set exactly 126 trading days, the second one 252 trading days, the third 378 trading days, and the final observation date 504 trading days after the effective start date of the option. For simplicity reasons, the time value of money is not considered for this example. In the first scenario, the basket of underlyings breaches the autocall barrier on the first observation date after six months, resulting in the payment of the coupon of 0,30 % plus the initial investment and the option is exercised. The following scenario involves breaking the autocall level on the third observation date. The investor receives an increased coupon of 0,90 % on the initially invested amount after 18 months, and the option is then terminated.

Table 1: Payoff scenarios for a 2-year Autocall option with gradually increasing coupons

Payoff at observation date:		**Scenario 1**	**Scenario 2**	**Scenario 3**	**Scenario 4**
1.	6 months	100,30 %	0,00 %	0,00 %	0,00 %
2.	12 months	-	0,00 %	0,00 %	0,00 %
3.	18 months	-	100,90 %	0,00 %	0,00 %
4.	24 months	-	-	101,95 %	100,00 %

In scenarios three and four, the option remains in place until the final observation date. The underlyings do not breach the barrier level in scenario four, and the investor only receives back the original investment. Scenario three includes the underlyings triggering the autocall event on the final observation date after 24 months, with the investor receiving a coupon of 1,95 % and the principal amount invested. (Bouzoubaa and Osseiran 2010, pp. 187-189) Autocall options are highly innovative and customizable instruments attractive for investors looking for specific equity exposure while enjoying the benefits of capital protection and the possibility of attractive yields. While the investor favors a redemption earliest possible (1st observation date), the bank would like the underlyings to stay below the autocall barrier. The preferred scenario for the investor would earn him an above-market yield on a very short-term product in a rallying market. He could then re-deploy the initial investment he receives back from the bank into other products that are more bullish to participate in the market's upcycle fully. On the other hand, the bank would have been provided with interest-free capital from the investor and earned a premium from selling the option in their preferred case. ***Figure 7*** below provides a brief overview of Banco Invest's portfolio of fifteen Autocall options. The class created in Python to calculate each option's payoff can be found in ***Appendix 27-39***. On the right, the different coupons for the first, second, and third observation date, as well as Floor and Cap, are given. The latter two refer to the minimum and maximum possible return of each option in case the underlyings simultaneously do not exceed their strike during the option's life, respectively do so on the last observation date, which is also the maturity date of the option.

Product ID	Name	Effective date	Maturity date	Cap	Floor	Coupons on observation dates:		
						1st	2nd	3rd
1015	Invest Health & Tech Jun-20	30/06/2020	07/07/2022	2.40%	0%	0.60%	1.20%	1.80%
1033	Invest Personal Care Ago-20	01/09/2020	07/09/2022	2.40%	0%	0.60%	1.20%	1.80%
1041	Invest Back to School Set-20	30/09/2020	07/10/2022	2.80%	0%	0.70%	1.40%	2.10%
1130	Invest Digital 5G Fev-21	26/02/2021	06/03/2023	2.40%	0%	0.60%	1.20%	1.80%
1152	Invest Natural Resources Abr-21	30/04/2021	05/05/2023	2.00%	0%	0.50%	1.00%	1.50%
1190	Invest Fintech Jul-21	30/07/2021	07/08/2023	2.00%	0%	0.30%	0.60%	1.20%
1203	Invest Health Innovation Ago-21	31/08/2021	07/09/2023	2.00%	0%	0.30%	0.60%	1.20%
1204	Invest Back to School Set-21	30/09/2021	06/10/2023	2.00%	0%	0.35%	0.70%	1.20%
1229	Invest Infraestruturas Globais Out-21	29/10/2021	06/11/2023	1.85%	0%	0.30%	0.60%	1.05%
1244	Invest Hydrogen Nov-21	30/11/2021	07/12/2023	1.80%	0%	0.25%	0.40%	0.60%
1260	Invest Communication & Media Dez-21	30/12/2021	05/01/2024	2.20%	0%	0.40%	0.80%	1.20%
1348	Invest Metaverse Mar-22	31/03/2022	08/04/2024	1.95%	0%	0.30%	0.60%	0.90%
1392	Invest Basic Resources Abr-22	29/04/2022	06/05/2024	2.00%	0%	0.30%	0.60%	0.90%
1429	Invest Technology Jun-22	30/06/2022	08/07/2024	2.00%	0%	0.25%	0.50%	0.75%
1455	Invest Blockchain Jul-22	29/07/2022	05/08/2024	2.40%	0%	0.40%	0.80%	1.20%

Figure 7: Autocall portfolio overview of Banco Invest

On the right, the option's ID and name are given. As indicated by their respective names, Banco Invest categorizes each Autocall option into a different industry, from which the underlyings are composed. ***Appendix 40-54*** provides an overview of the different option's underlyings and their correlations within Banco Invest's portfolio. Correlation is measured one year prior to the respective option basket's setup until the option's trade date. It can be seen that the different stocks within an option are positively correlated with one another. Selling the options, Banco Invest is "short on the correlation" between the reference assets. With a lower correlation, the possibility of all five stocks ending above the autocall barrier will likely decrease. This is in favor of Banco Invest as it results in the bank not having to pay the coupons agreed.

5.1 Portfolio Delta

When Banco Invest sells Autocall options to its investors, the bank is short on the underlying. Assuming the market rises, the deltas of the bank's Autocall options would also rise, as the likelihood of the underlyings reaching their strikes would increase. Hence, hedging the spot risk in the form of the delta is a top priority. In order to neutralize the portfolio against delta, the bank would have to be buy the underlyings. ***Figure 8*** below summarizes the calculated delta of each Autocall option and the overall portfolio of Autocall options at Banco Invest. Furthermore, the notional value of the delta was also calculated to estimate the EUR amount that every option and the overall portfolio would have to be hedged with. Attention is drawn to the fact that the below deltas were calculated assuming a long position on the option, hence from the perspective of Banco Invest, they have to be taken as negative values. Looking at the figure below, it is noticeable that four deltas are at zero. This is, for example, the case for the delta for Autocall ***ID 1015***. Considering, at the time of pricing, the option only had five trading days left until its maturity date (***Figure 7***) and additionally looking at the strike prices of each underlying in relation to the prices the underlyings were trading as of 30/06/2022 (***Appendix 40***), the chances of the underlying SAP GY Equity ending up above its strike under normal market conditions are close to zero, hence the delta value.

ID	Notional	Weight	Delta	Notional value
1015	4.098.767,49 €	12,29%	0,0000	- €
1033	1.809.624,96 €	5,42%	0,0120	21.674,04 €
1041	2.100.436,10 €	6,30%	0,0000	- €
1130	2.339.921,22 €	7,01%	0,0118	27.551,94 €
1152	1.601.003,35 €	4,80%	0,0215	34.355,78 €
1190	2.733.564,22 €	8,19%	0,0019	5.287,06 €
1203	2.417.800,74 €	7,25%	0,0031	7.462,10 €
1204	1.760.549,78 €	5,28%	0,0000	- €
1229	1.737.793,20 €	5,21%	0,0090	15.601,05 €
1244	2.877.061,96 €	8,62%	0,0044	12.679,22 €
1260	1.622.300,08 €	4,86%	0,0000	- €
1348	2.179.670,84 €	6,53%	0,0091	19.902,81 €
1392	1.925.444,03 €	5,77%	0,0083	15.911,49 €
1429	2.653.565,40 €	7,95%	0,0024	6.282,54 €
1455	1.500.000,00 €	4,50%	0,0038	5.674,80 €

Delta (Δ) - Aggregated Autocall options Portfolio

Option type	Notional	Weight	Delta	Notional value
Autocall	**33.357.503,37 €**	**100,00%**	**0,0052**	**172.382,82 €**

Figure 8: Delta (Δ) of Banco Invest's Autocall options portfolio

The same can be said for the other IDs where the delta is estimated to be zero. In each case, either one or multiple underlyings are too far away from their respective strike price, implying the likelihood of the option being auto-called is close to zero. The highest delta can be observed for option ID 1152 (0,0215) and 1130 (0,0118), respectively. Both the option's underlyings are reasonably close to reaching their strikes, increasing the likelihood of the option being auto-called. In terms of the overall portfolio delta, multiplying each option's weight with its respective delta it was estimated to be 0,0052. Hence, buying the different underlyings worth a total of 172.382,82 € would neutralize the portfolio's delta. However, the deltas estimated are still very low compared to plain vanilla or other exotic options. This is because of the partly quite high strikes for some of the underlyings but can change pretty quickly around observation dates. Autocall options show discontinuities in their payoff profile on these dates, resulting in the "Greeks" and, in particular, delta being unstable and explosive, which makes hedging more challenging to maintain. Delta must be closely watched at these dates to avoid suddenly trading large quantities of the underlyings. Further, hedging delta on a daily basis avoids being forced to buy large quantities of the underlying when it's trading around the autocall barrier close to an observation date. This could push the price of the underlying unintendedly above its strike. Generally, it can be said that the closer to the observation date or maturity the options are, the more frequently it is advised to adjust the hedging. By buying the underlyings to hedge the Autocall portfolio's delta, Banco Invest is also long on dividends. In general, the more dividends are expected to be paid by the underlying, the more it will profit the bank on its hedge. However, as explained before, the drift is also adjusted for this, lowering the expected future movement of an underlying over the course of the option's life. Most dividend are announced for longer terms in the future, but only voted on once it gets close to the actual payout date. Close attention must therefore be paid to dividend announcements, especially companies lowering their dividend projections for the future. Ignoring this would falsely imply calculating the option's price on the basis of a lower expected movement in the underlying's price. This would ultimately result in a lower than usual Autocall option price or the strikes of the respective underlying being set too low. Maintaining delta neutrality for an individual option on an asset would be prohibitively expensive if the asset was traded daily. However, doing so for a portfolio of several options is feasible. This is because profits from a variety of trades offset the cost of daily rebalancing. Hence, there are significant economies of scale in trading derivatives.

5.2 Portfolio Gamma

In ***Figure 9*** below, the gamma for each option and also for the whole portfolio of Autocall options is summarized. As already seen for the deltas for Autocalls with IDs 1015, 1041, 1204, and 1260, their gammas are also zero. The options are deep out of the money, so it is not surprising as even bigger moves in the underlying's prices wouldn't impact the deltas of the options by much. In terms of rebalancing, these options do not frequently need to be delta hedged to neutralize directional exposure in the market. The biggest gamma value can be observed for the Autocall with ID 1392. This is due to all of the option's underlyings being close to or above their respective strike price with the option approaching its first observation date, hence a move in the price of the underlying significantly impacts the value of delta. This also means that this option has to be delta re-hedged the most frequent, in relative terms, as of 30/06/2022. For the overall portfolio, the gamma, calculated by taking the weighted average, was estimated to be 0,1963.

ID	Notional	Weight	Gamma
1015	4.098.767,49 €	12,29%	0,0000
1033	1.809.624,96 €	5,42%	0,2395
1041	2.100.436,10 €	6,30%	0,0000
1130	2.339.921,22 €	7,01%	0,0000
1152	1.601.003,35 €	4,80%	0,3414
1190	2.733.564,22 €	8,19%	0,1934
1203	2.417.800,74 €	7,25%	0,4244
1204	1.760.549,78 €	5,28%	0,0000
1229	1.737.793,20 €	5,21%	0,2352
1244	2.877.061,96 €	8,62%	0,3257
1260	1.622.300,08 €	4,86%	0,0000
1348	2.179.670,84 €	6,53%	0,4138
1392	1.925.444,03 €	5,77%	0,5699
1429	2.653.565,40 €	7,95%	0,1657
1455	1.500.000,00 €	4,50%	0,1513

Gamma (Γ) - Aggregated Autocall options Portfolio

Option type	Notional	Weight	Gamma
Autocall	**33.357.503,37 €**	**100,00%**	**0,1963**

Figure 9: Gamma (Γ) of Banco Invest's Autocall options portfolio

5.3 Non-linear Delta-Gamma-VaR

Figure 10 below summarizes the VaR for each Autocall option and the combined VaR for the undiversified and diversified portfolios. The diversified portfolio is expected to not lose more than EUR 23.173,93 over the course of one trading day with 99,9 % confidence. The undiversified VaR is almost three times higher as it calculated by adding the fifteen individual VaR numbers for each option. For the diversified VaR, the diversification effects of the different option's underlyings are additionally considered for the estimation. This is done by incorporating the variance-covariance matrix of the total portfolio's underlyings. Each option itself is focused on a specific industry, so individual VaR tends to be relatively high. It entails lump risk. Therefore, the portfolio VaR will always be smaller than summing up the individual VaR numbers of each position.

ID	Notional	Weight	VaR		Notional value
1015	4.098.767,49 €	12,29%	0,00%		- €
1033	1.809.624,96 €	5,42%	0,15%		2.736,75 €
1041	2.100.436,10 €	6,30%	0,00%		- €
1130	2.339.921,22 €	7,01%	0,42%		9.893,21 €
1152	1.601.003,35 €	4,80%	0,88%		14.055,86 €
1190	2.733.564,22 €	8,19%	0,05%		1.260,01 €
1203	2.417.800,74 €	7,25%	0,27%		6.434,21 €
1204	1.760.549,78 €	5,28%	0,00%		- €
1229	1.737.793,20 €	5,21%	0,45%		7.758,49 €
1244	2.877.061,96 €	8,62%	0,09%		2.478,01 €
1260	1.622.300,08 €	4,86%	0,00%		- €
1348	2.179.670,84 €	6,53%	0,65%		14.276,33 €
1392	1.925.444,03 €	5,77%	0,21%		3.995,37 €
1429	2.653.565,40 €	7,95%	0,03%		875,87 €
1455	1.500.000,00 €	4,50%	0,23%		3.427,30 €
				Undiversified VaR:	**67.191,41 €**

1-day VaR @ 99,9 % - Aggregated Autocall options Portfolio

Option type	Notional	Weight	VaR	Volatility	Diversified VaR
Autocall	**33.357.503,37 €**	**100,00%**	**0,07%**	**5,11%**	**23.173,93 €**

Figure 10: Delta-Gamma VaR for the Autocall options portfolio of Banco Invest

Considering a mix of these different industries in the overall Autocall options portfolio, the combined VaR is much lower. The estimated undiversified VaR above is almost three times larger than the diversified one, implying that, looking at the bigger picture, diversification is critical in limiting downside risk.

10 Recommendation - Group part

This chapter address how the bank's management should deal with the risk associated with the derivatives Portfolio. ***Figure 35*** below summarizes the delta, gamma, and Delta-Gamma Value-at-Risk for Banco Invest's overall options portfolio. The total derivatives portfolio of the bank has a notional of EUR 157.067.916, consisting of 53 different options. The 1-day Value-at-Risk at 99,9% confidence level for the bank's overall derivatives portfolio is EUR 372.773, implying a 99,9 % probability the portfolio will not lose more over the next trading day.

Banco Invest - Aggregated Derivatives Portfolio	
Notional	157.067.916,00 €
No. of option positions	53
Delta (Δ)	0,0163
Gamma (Γ)	0,3166
Volatility	4,91%
VaR (1d, @ 99,9%)	0,24%
VaR (1d, @ 99,9%)	372.773,00 €

Figure 35: Aggregated Portfolio Delta-Gamma VaR

As the bank does not take a directional risk on the market, the delta on combined option's portfolio must be neutralized with an appropriate hedging strategy. All five option types in the Banco Invest derivatives portfolio are basket options. The challenge of hedging, when facing options with a basket of underlying's, becomes evident in their correlated structure. This makes the evaluation of the contract's price but also the risks, e.g., delta, gamma, and their hedging a complex procedure. (Su 2006, pp. 3-5) This is because it is difficult to detangle the underlying basket's distribution. The correlation between the underlying tends to be volatile and can only be estimated. This further complicates the "perfect" hedging of basket options. As a result, in many cases, only a part of the underlying basket is used for hedging, or the payoffs of the basket are replicated "super-hedged". (Su 2008, pp. 19-23) Another difficulty arises from the number of underlying assets: When following a standard dynamic hedging strategy, a hedging portfolio for the basket options should be related to the underlying assets in the basket. The larger the amount of underlying's the more difficult it is to implement such a dynamic strategy and the larger the transactions cost, caused by the continuous rebalancing, become. Since most of the options are "near-zero-gamma", which means that the directionality, the delta of the option is not greatly affected by changes in the underlying market prices, a dynamic hedging strategy can be implemented as major changes in the delta are not expected to be caused by changes in the underlying market prices. Transaction costs for rebalancing will occur but will be manageable as they do not occur very frequently. Lamberton and Lapeyre (1992) showed that a dynamic hedge on even a subset of the underlying's works well: they developed a method using multiple regression analysis to create a dynamic approximate hedging portfolio of plain-vanilla options on only a subset of the underlying's. For our "near-zero-gamma" options, such a dynamic hedge could further reduce the already low cost of rebalancing. A static hedging strategy has the advantage that transaction costs caused by continuous rebalancing can be avoided, and therefore this strategy could have a better hedging performance. (Su 2008, pp 2-4) Su (2006) used the Principal Components Analysis (PCA) to demonstrate that also a static hedge on a subset of the underlying's performs well: The PCA was used to determine a dominant subset of assets of the basket. Since a dynamic hedge of a basket option often only approximates the optimal hedge, the complete neutralization of the delta can only be achieved by a static hedge. Since Banco Invest instructs it takes no directional risk in the market, the only hedging strategy that fits this case is a static strategy as described above. Moreover, since the assets in the respective basket options are all in the same thematic investment universe, it is worthwhile to follow the approach of Su (2006) to determine whether it is sufficient to apply a static hedge only to a subset of the underlying assets, due to the high correlation between them.

Bibliography

Bodie, Zvi, Alex Kane, and Alan J. Marcus. 2021. "Investments." New York.

Bouzoubaa, Mohamed, and Adel Osseiran. 2010. "Exotic Options and Hybrids." Chichester.

Castellacci, Giuseppe, and Michael J. Siclari. 2003. "The Practice of Delta–Gamma VaR: Implementing the Quadratic Portfolio Model." *European Journal of Operational Research* 150 (3): 529–45. https://doi.org/10.1016/S0377-2217(02)00782-8.

Deng, Geng, Joshua Mallett, and Craig McCann. 2011. "Modeling Autocallable Structured Products." *Journal of Derivatives and Hedge Funds* 17 (4): 326–40. https://doi.org/10.1057/jdhf.2011.25.

Deutsche Bundesbank. 2022. "Deutsche Bundesbank: Marktrisiko." 2022. https://www.bundesbank.de/de/aufgaben/bankenaufsicht/einzelaspekte/eigenmittelanforderungen/marktrisiko/marktrisiko-598476.

Fallon, William. 1996. "Calculating Value-at-Risk." Philadelphia. https://www.researchgate.net/publication/2428988_Calculating_Value-at-Risk.

Guillaume, Tristan. 2015a. "Autocallable Structured Products." *THE JOURNAL OF DERIVATIVES* 73. www.iijournals.com.

———. 2015b. "Analytical Valuation of Autocallable Notes." *International Journal of Financial Engineering* 02 (02): 1550016. https://doi.org/10.1142/s2424786315500164.

Hilpisch, Yves. 2015. "Derivatives Analytics with Python." Chichester.

Hull, John C. 2018. "Risk Management and Financial Institutions." New York.

Hull, John C. 2021. "OPTIONS, FUTURES, AND OTHER DERIVATIVES." New York.

Jorion, Philippe. 2007. *Philippe Jorion - Value at Risk, 3rd Ed._ The New Benchmark for Managing Financial Risk (2006, McGraw-Hill)*. Third. The McGraw-Hill Companies, Inc.

Leoni, Peter. 2014. *The Greeks and Hedging Explained*.

Romano. 2017. “Options Trading.” May 2017. https://romanornr.medium.com/options-trading-fd4d0bffb2c5.

Su, Xia. 2006. “Hedging Basket Options by Using a Subset of Underlying Assets.” 14. Bonn. https://www.econstor.eu/bitstream/10419/22959/1/bgse14_2006.pdf.

———. 2008. “Essays on Basket Options Hedging and Irreversible Investment Valuation.” Rheinische Friedrich-Wilhelms-Universität Bonn. https://bonndoc.ulb.uni-bonn.de/xmlui/handle/20.500.11811/3322.

Taleb, Nassim Nicholas. 1997. “Dynamic Hedging: Managing Vanilla and Exotic Options.” New York.

Weert, Frans de. 2008. “Exotic Options Trading.” Chichester.

Wilmott, Paul. 1998. *Derivatives : The Theory and Practice of Financial Engineering*. John Wiley & Sons Ltd.

Yan, Yuxing. 2017. *Python for Finance : Financial Modeling and Quantitative Analysis Explained*. Birmingham: Packt Publishing.

Yen Jerome, and Kin Keung Lai. 2015. “Emerging Financial Derivatives.” New York.

Z. Tong, Kevin. 2019. “A Recursive Pricing Method for Autocallables under Multivariate Subordination.” *Quantitative Finance and Economics* 3 (3): 440–55. https://doi.org/10.3934/QFE.2019.3.440.

Appendix

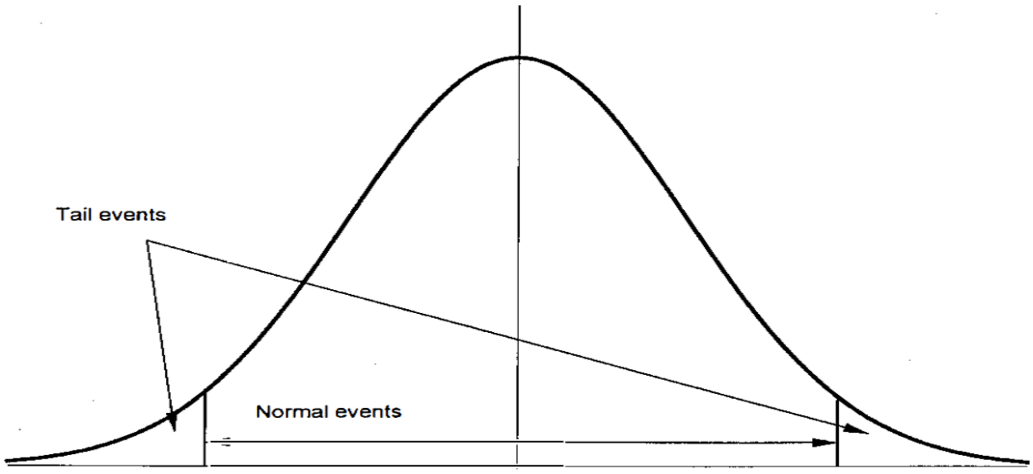
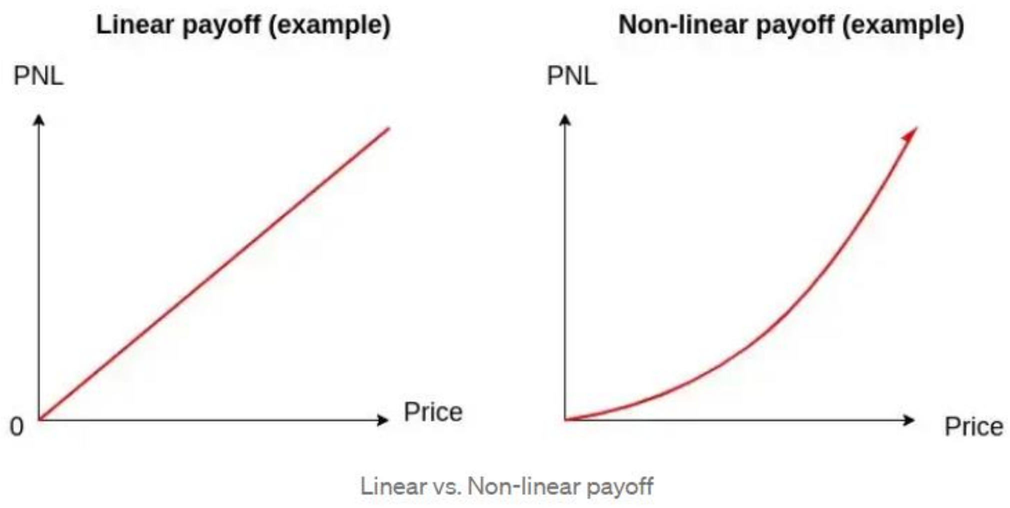


Figure 27.1 'Normal events' and 'tail events'.

Appendix 1: Value-at-Risk distribution showing possible tail events (Wilmott 1998, p. 338)

CI	Z
80%	1,282
85%	1,440
90%	1,645
95%	1,960
99%	2,576
99,5%	2,807
99,9%	3,291

Appendix 2: Overview of most common z-statistic for VaR calculation



Appendix 3: Linear / Non-linear VaR (Romano 2017)

Name	Symbol	Derivative	Measures	Definition
Delta	Δ	$\frac{\partial c}{\partial s}$	Equity Exposure	Measures how much an option's price is estimated to shift in response to a change of a one unit in the underlying security
Gamma	Γ	$\frac{\partial^2 c}{\partial^2 s}$	Payout Convexity	Measures the amount of change in Delta if the price of the underlying security changes by one unit
Theta	Θ	$\frac{\partial c}{\partial T}$	Time Decay	Measures the change in the option price induced by the decrease of 1 day of the remaining time to maturity
Vega	V	$\frac{\partial c}{\partial \sigma}$	Volatility Exposure	Measures how much an option's price will change in response to a 1% change in the volatility of the underlying securities
Rho	P	$\frac{\partial c}{\partial r}$	Interest Rate Exposure	Measures how much the value of an option changes based on a 1% change in the interest rate

Appendix 4: Overview Greeks – In accordance with (Leoni 2014, pp. 85-97)

```python
def drift_calc(data, return_type='log'):
    if return_type == 'log':
        lr = log_returns(data)
    elif reutrn_type == 'simple':
        lr = simple_returns(data)
    u = lr.mean()
    var = lr.var()
    drift = u-(0.5*var)
    try:
        return drift.values
    except:
        return drift

drift = drift_calc(modified_data)

div = portfolio[self.id]['div']
# Drift adjusting if dividend paying (for Brownsche Motion)
if div > 0:
    drift = drift - div
```

Appendix 5: Drift calculation in Python

```python
covar = log_ret.cov() #covariance matrix.

chol = np.linalg.cholesky(covar) #create cholesky matrix from covariance matrix

uncorr_x = norm.ppf(np.random.rand(num_stocks, simulated_days)) #stocks, days

corr_x = np.dot(chol, uncorr_x)

corr_2 = np.zeros_like(corr_x) #Return an array of zeros with the same shape and type as a given array.
for i in range(num_stocks):
    corr_2[i] = np.exp(drift[i] + corr_x[i])
corr_2[0]

stock0 = pd.DataFrame()                                     #create new data frame
for s in range(len(ticks)):
    ret_reshape = corr_2[s]
    ret_reshape = ret_reshape.reshape(simulated_days) #Gives a new shape to an array without changing its data
    price_list = np.zeros_like(ret_reshape)
    price_list[0] = data.iloc[-1, s] #iloc = Purely integer-location based indexing for selection by position
    for t in range(1, simulated_days):
        price_list[t] = price_list[t-1]*ret_reshape[t]
```

Appendix 6: Cholesky decomposition in Python

```python
def get_vola(portfolio, volatility_file, stock_file):
    for a in range(2):
        for i in portfolio.keys():
            ticks = portfolio[i]['underlyings']
            today = "30-06-2022"
            today = pd.to_datetime(today)
            end = today
            #end = portfolio[i]['effective_date']
            vola = "VOLATILITY_30D"
            stock_vola = []

            if portfolio[i]["vol_type"] == "I": #calculation of implied volatility
                ids = portfolio[i]["underlyings"]
                for underlying in ids:
                    underlying_vola = volatility_file._get_value(vola, underlying)
                    stock_vola.append(underlying_vola)
                vol = (sum(stock_vola)/len(stock_vola))
                portfolio[i]["vol"] = vol
                #No Impl. vol.?, change vol_type to historical for second loop
                if  math.isnan(vol) == True:
                    portfolio[i]["vol_type"] = "H"
```

Appendix 7: Volatility calculation in Python (1/2)

```python
            else:
                portfolio[i]["vol_type"] = "H"
                def vola_data(tickers): #basically same funcion as used in cholesky
                    vol_data = pd.DataFrame() #create new data frame
                    for t in tickers: #loop through underlying tickers
                        vol_data[t] = stock_file[t].iloc[1:]
                    return(vol_data)

                data_ticks = vola_data(ticks)
                end_date = len(data_ticks.loc[:end]) #determine lenght of data frame for vola
                start_date = end_date - 30

                used_data = data_ticks.iloc[start_date:end_date]
                #print(used_data)

                def log_returns(data):
                    return (np.log(1+data.pct_change()))
                stdev = log_returns(used_data).std().values
                #monthly vola of the option = equal weighting vola of underlyings
                monthly_vol = sum(stdev)/len(stdev)
                vol = monthly_vol * sqrt(12) #annual vola
                portfolio[i]["vol"] = vol #append dictionary


get_vola(options_portfolio, file_vola, file_stocks)
```

Appendix 8: Volatility calculation in Python (2/2)

```python
def interpolation(rates, maturity_date):
    today = "30-06-2022"
    today = pd.to_datetime(today)
    today = today.to_pydatetime().date()

    maturity_day = maturity_date.day
    maturity_month = maturity_date.month
    maturity_year = maturity_date.year
    name = rates.columns[0]

    for a in range(len(rates)-1):
        rate_date = rates.index[a]
        prev_date = rates.index[a-1]
        next_date = rates.index[a+1]
        rate_day = rates.index[a].day
        rate_month = rates.index[a].month
        rate_year = rates.index[a].year

        if rate_date == maturity_date:
            r = rates._get_value(rate_date, name)
```

Appendix 9: Linear interpolation in Python to get discount rates for Option payoffs (1/2)

```python
        elif rate_month == maturity_month and rate_year == maturity_year:
            #structure: rate_day - maturity_date - next_rate
            if rate_day < maturity_day:
                #todays rate, shorter maturity
                r1 = rates._get_value(rate_date, name)
                #next rate, longer maturity
                r2 = rates._get_value(next_date, name)
                #day difference between rate_date and maturity_date
                t1 = abs(rate_date.to_pydatetime().date() - today)
                #day difference bewween next_date and maturity_date
                t2 = abs(next_date.to_pydatetime().date() - today)
                #day difference between rate_date and maturity_date
                tn = abs(today - maturity_date.to_pydatetime().date())



                r = r1 + (r2-r1)/((t2-t1).days)*((tn-t1).days)

            else:
                r1 = rates._get_value(prev_date, name)
                r2 = rates._get_value(rate_date, name)

                t1 = abs(prev_date.to_pydatetime().date() - today)
                t2 = abs(rate_date.to_pydatetime().date() - today)
                tn = abs(today - maturity_date.to_pydatetime().date())


                r = r1 + (r2-r1)/((t2-t1).days)*((tn-t1).days)
    return r
```

Appendix 10: Linear interpolation in Python to get discount rates for Option payoffs (2/2)

$$\Delta = \frac{S_t(\varepsilon) - S_t}{\varepsilon} \tag{18}$$

$$\Gamma = \frac{\Delta_t(\varepsilon) - \Delta_t}{\varepsilon} \tag{19}$$

Appendix 11: Equations used for Delta/Gamma calculation

```python
# Calculating per Option:
for i in options_portfolio.keys():
    print(i)
    # Determining the risk free rate by maturity matching the Swap Rate with the maturity of the option
    r = interpolation(swaps, options_portfolio[i]['maturity'])

    # Get infos
    ticks = options_portfolio[i]['underlyings']
    start = options_portfolio[i]['effective_date']
    S = options_portfolio[i]['spot']
    K = options_portfolio[i]['strike']

    today = "30-06-2022"
    today = datetime.strptime(today, '%d-%m-%Y').date()

    T = options_portfolio[i]['maturity'].to_pydatetime().date() - today
    T = T.days/365
    div = options_portfolio[i]['div']
    vol = options_portfolio[i]["vol"]

    price=0
    delta=0
    sym_delta = 0
    delta_2=0
    delta_3=0
    g=0
    var = 0
    z = 3.291
```

Appendix 12: Delta/ gamma & VaR Calculation for each option & overall portfolio in Python (1/15)

```python
# Call different classes for payoffs and delta
    if options_portfolio[i]["payoff_id"] == 2:

        altiplano = Altiplano(options_portfolio,i)
        payoffs = altiplano.payoff()
        # Discounting the payoff with the maturity matched risk free rate
        price = payoffs[0] * math.exp(-r*T)

        #Deltas
        pos_price = payoffs[1] * math.exp(-r*T)
        neg_price = payoffs[2] * math.exp(-r*T)
        delta = (pos_price - price) / (percentage_change)
        sym_delta = (pos_price - neg_price)/(2*percentage_change)

        #Second & Third Deltas
        pos_price2 = payoffs[3] * math.exp(-r*T)
        neg_price2 = payoffs[4] * math.exp(-r*T)
        delta_2 = (pos_price2 - pos_price) / (percentage_change)
        delta_3 = (neg_price2 - neg_price) / (percentage_change)

        #Gamma
        delta_dif = delta_2 - delta
        g = abs(delta_dif)/percentage_change
        #sym_g = ()

        #print(pos_price)
        #print(neg_price)

        # 1d VaR and 99,9% interval (Z-score=3.291), calculated on 1y
        var = (delta*3.291*np.sqrt(1/252)*vol - g/2*(3.291*np.sqrt(1/252)*vol)**2)*np.sqrt(252)
```
Appendix 13: Delta/ gamma & VaR Calculation for each option & overall portfolio in Python (2/15)

```python
    elif options_portfolio[i]["payoff_id"] == 4:
        auto = Autocall(options_portfolio, i)
        payoffs = auto.payoff()
        # Discounting the payoff with the maturity matched risk free rate
        price = payoffs[0] * math.exp(-r*T)

        #Deltas
        pos_price = payoffs[1] * math.exp(-r*T)
        neg_price = payoffs[2] * math.exp(-r*T)
        delta = (pos_price - price) / (percentage_change)
        sym_delta = (pos_price-neg_price)/(2*percentage_change)

        #Second & Third Deltas
        pos_price2 = payoffs[3] * math.exp(-r*T)
        neg_price2 = payoffs[4] * math.exp(-r*T)
        delta_2 = (pos_price2 - pos_price) / (percentage_change)
        delta_3 = (neg_price2 - neg_price2) / (percentage_change)

        #Gamma
        delta_dif = delta_2 - delta
        g = abs(delta_dif)/percentage_change
        #sym_g = ()

        #print(pos_price)
        #print(neg_price)

        # 1d VaR and 99,9% interval (Z-score=3.291), calculated on 1y
        var = (delta*3.291*np.sqrt(1/252)*vol - g/2*(3.291*np.sqrt(1/252)*vol)**2)*np.sqrt(252)
```
Appendix 14: Delta/ gamma & VaR Calculation for each option & overall portfolio in Python (3/15)

```
elif options_portfolio[i]["payoff_id"] == 6:
    digi = Call_Digital(options_portfolio, i)
    payoffs = digi.payoff()
    # Discounting the payoff with the maturity matched risk free rate
    price = payoffs[0] * math.exp(-r*T)

    #Deltas
    pos_price = payoffs[1] * math.exp(-r*T)
    neg_price = payoffs[2] * math.exp(-r*T)
    delta = (pos_price - price) / (percentage_change)
    sym_delta = (pos_price-neg_price)/(2*percentage_change)

    #Second & Third Deltas
    pos_price2 = payoffs[3] * math.exp(-r*T)
    neg_price2 = payoffs[4] * math.exp(-r*T)
    delta_2 = (pos_price2 - pos_price) / (percentage_change)
    delta_3 = (neg_price2 - neg_price2) / (percentage_change)

    #Gamma
    delta_dif = delta_2 - delta
    g = abs(delta_dif)/percentage_change
    #sym_g = ()

    #print(pos_price)
    #print(neg_price)

    # 1d VaR and 99,9% interval (Z-score=3.291), calculated on 1y
    var = (delta*3.291*np.sqrt(1/252)*vol - g/2*(3.291*np.sqrt(1/252)*vol)**2)*np.sqrt(252)
```

Appendix 15: Delta/ gamma & VaR Calculation for each option & overall portfolio in Python (4/15)

```
elif options_portfolio[i]["payoff_id"] == 11:
    indi = Indicap(options_portfolio,i)
    payoffs = indi.payoff()
    # Discounting the payoff with the maturity matched risk free rate
    price = payoffs[0] * math.exp(-r*T)

    #Deltas
    pos_price = payoffs[1] * math.exp(-r*T)
    neg_price = payoffs[2] * math.exp(-r*T)
    delta = (pos_price - price) / (percentage_change)
    sym_delta = (pos_price-neg_price)/(2*percentage_change)

    #Second & Third Deltas
    pos_price2 = payoffs[3] * math.exp(-r*T)
    neg_price2 = payoffs[4] * math.exp(-r*T)
    delta_2 = (pos_price2 - pos_price) / (percentage_change)
    delta_3 = (neg_price2 - neg_price2) / (percentage_change)

    #Gamma
    delta_dif = delta_2 - delta
    g = abs(delta_dif)/percentage_change
    #sym_g = ()

    #print(pos_price)
    #print(neg_price)

    # 1d VaR and 99,9% interval (Z-score=3.291), calculated on 1y
    var = (delta*3.291*np.sqrt(1/252)*vol - g/2*(3.291*np.sqrt(1/252)*vol)**2)*np.sqrt(252)
```

Appendix 16: Delta/ gamma & VaR Calculation for each option & overall portfolio in Python (5/15)

```python
else:
    capprotect = Capital_Protected(options_portfolio,i)
    payoffs = capprotect.payoff()
    # Discounting the payoff with the maturity matched risk free rate
    price = payoffs[0] * math.exp(-r*T)

    #Deltas
    pos_price = payoffs[1] * math.exp(-r*T)
    neg_price = payoffs[2] * math.exp(-r*T)
    delta = (pos_price - price) / (percentage_change)
    sym_delta = (pos_price-neg_price)/(2*percentage_change)

    #Second & Third Deltas
    pos_price2 = payoffs[3] * math.exp(-r*T)
    neg_price2 = payoffs[4] * math.exp(-r*T)
    delta_2 = (pos_price2 - pos_price) / (percentage_change)
    delta_3 = (neg_price2 - neg_price2) / (percentage_change)

    #Gamma
    delta_dif = delta_2 - delta
    g = abs(delta_dif)/percentage_change
    #sym_g = ()

    #print(pos_price)
    #print(neg_price)

    # 1d VaR and 99,9% interval (Z-score=3.291), calculated on 1y
    var = (delta*3.291*np.sqrt(1/252)*vol - g/2*(3.291*np.sqrt(1/252)*vol)**2)*np.sqrt(252)
```

Appendix 17: Delta/ gamma & VaR Calculation for each option & overall portfolio in Python (6/15)

```python
options_portfolio[i]['option_price'] = price
options_portfolio[i]['delta'] = delta
options_portfolio[i]["symmetric_delta"] = sym_delta
options_portfolio[i]['delta2'] = delta_2
options_portfolio[i]['delta3'] = delta_3
options_portfolio[i]['g'] = g
options_portfolio[i]["var"] = var
```

Appendix 18: Delta/ gamma & VaR Calculation for each option & overall portfolio in Python (7/15)

Per Option Type: Delta, Second Delta, Gamma and VaR

Weights

```
alti_notional = 0
auto_notional = 0
digi_notional = 0
indi_notional = 0
capprotect_notional = 0
sum_notional = 0
for key in options_portfolio.keys():

    if options_portfolio[key]["payoff_id"] == 4:
        notional = options_portfolio[key]["notional"]
        auto_notional += notional
        sum_notional += notional

    elif options_portfolio[key]["payoff_id"] == 4:
        notional = options_portfolio[key]["notional"]
        options_portfolio[key]["weight_type"] = notional/auto_notional
        options_portfolio[key]["weight_total"] = notional/sum_notional

        options_portfolio[key]["weighted_delta_type"] = options_portfolio[key]["weight_type"]*options_portfolio[key]["delta"]
        options_portfolio[key]["weighted_gamma_type"] = options_portfolio[key]["weight_type"]*options_portfolio[key]["g"]

        options_portfolio[key]["weighted_delta_total"] = options_portfolio[key]["weight_total"]*options_portfolio[key]["delta"]
        options_portfolio[key]["weighted_gamma_total"] = options_portfolio[key]["weight_total"]*options_portfolio[key]["g"]

#options_portfolio
options_type = {"Altiplano": {}, "Autocall": {}, "Call_Digital": {}, "Indicap": {}, "Capital_Protect": {}}
options_type["Altiplano"]["notional"] = alti_notional
options_type["Autocall"]["notional"] = auto_notional
options_type["Call_Digital"]["notional"] = digi_notional
options_type["Indicap"]["notional"] = indi_notional
options_type["Capital_Protect"]["notional"] = capprotect_notional
```

Appendix 19: Delta/ gamma & VaR Calculation for each option & overall portfolio in Python (8/15)

```
alti_delta = 0
auto_delta = 0
digi_delta = 0
indi_delta = 0
capprotect_delta = 0

alti_gamma = 0
auto_gamma = 0
digi_gamma = 0
indi_gamma = 0
capprotect_gamma = 0

for key in options_portfolio.keys():

    if options_portfolio[key]["payoff_id"] == 4:
        auto_delta += options_portfolio[key]["weighted_delta_type"]
        auto_gamma += options_portfolio[key]["weighted_gamma_type"]

options_type["Altiplano"]["delta"] = alti_delta
options_type["Autocall"]["delta"] = auto_delta
options_type["Call_Digital"]["delta"] = digi_delta
options_type["Indicap"]["delta"] = indi_delta
options_type["Capital_Protect"]["delta"] = capprotect_delta

options_type["Altiplano"]["gamma"] = alti_gamma
options_type["Autocall"]["gamma"] = auto_gamma
options_type["Call_Digital"]["gamma"] = digi_gamma
options_type["Indicap"]["gamma"] = indi_gamma
options_type["Capital_Protect"]["gamma"] = capprotect_gamma
```

Appendix 20: Delta/ gamma & VaR Calculation for each option & overall portfolio in Python (9/15)

```
alti_underlyings = []
auto_underlyings = []
digi_underlyings = []
indi_underlyings = []
capprotect_underlyings = []

alti_underlyings_weights = []
auto_underlyings_weights = []
digi_underlyings_weights = []
indi_underlyings_weights = []
capprotect_underlyings_weights = []

today = "30-06-2022"
for key in options_portfolio.keys():

    if options_portfolio[key]["payoff_id"] == 4:
        underlyings_l = options_portfolio[key]["underlyings"]
        for underlyings in underlyings_l:
            single_weight = (1/5) * options_portfolio[key]["weight_type"] #stock weight in option * total option-type weight
            auto_underlyings_weights.append(single_weight)
            auto_underlyings.append(underlyings)

alti_underlyings_weights = np.array(alti_underlyings_weights)
auto_underlyings_weights = np.array(auto_underlyings_weights)
digi_underlyings_weights = np.array(digi_underlyings_weights)
indi_underlyings_weights = np.array(indi_underlyings_weights)
capprotect_underlyings_weights = np.array(capprotect_underlyings_weights)

options_type["Autocall"]["underlyings"] = auto_underlyings
options_type["Autocall"]["weights"] = auto_underlyings_weights

#options_type
```

Appendix 21: Delta/ gamma & VaR Calculation for each option & overall portfolio in Python (10/15)

For the whole Portfolio: Delta, Second Delta, Gamma and VaR

```
total_delta = 0
total_gamma = 0

for key in options_portfolio.keys():

    total_delta += options_portfolio[key]["weighted_delta_total"]
    total_gamma += options_portfolio[key]["weighted_gamma_total"]

total_portfolio = {}
total_portfolio["notional"] = sum_notional
total_portfolio["delta"] = total_delta
total_portfolio["gamma"] = total_gamma
```

Appendix 22: Delta/ gamma & VaR Calculation for each option & overall portfolio in Python (11/15)

Weights of the underlyings

```
list_underlyings = [] #== tickers
weights = []
today = "30-06-2022"
for key in options_portfolio.keys():
    underlyings_l = options_portfolio[key]["underlyings"]
    for underlyings in underlyings_l:
        single_weight = (1/5) * options_portfolio[key]["weight_total"] #stock weight in option * total option weight
        weights.append(single_weight)
        list_underlyings.append(underlyings)

weights = np.array(weights)
total_portfolio["underlyings"] = list_underlyings
total_portfolio["weights"] = weights
```

Appendix 23: Delta/ gamma & VaR Calculation for each option & overall portfolio in Python (12/15)

```python
weights = np.array(weights)

def vola_data(tickers): #basically same funcion as used in cholesky
    vol_data = pd.DataFrame(columns=tickers) #create new data frame
    for t in tickers: #loop through underlying tickers
        vol_data[t] = file_stocks[t].iloc[1:]
    return(vol_data)

tickerrs = list_underlyings


data_ticks = vola_data(tickerrs)

end_date = len(data_ticks.loc[:today]) #determine lenght of data frame for vola --> 30d volatility since effective date or today?
start_date = end_date - 30

used_data = data_ticks.iloc[start_date:end_date]


def log_returns(data):
    return (np.log(1+data.pct_change()))

returns = log_returns(used_data)
covar = returns.cov() * 12 #annualize monthly covariance

vol = np.sqrt(np.dot(weights.T, np.dot(covar, weights)))

total_portfolio["vol"] = vol
```

Appendix 24: Delta/ gamma & VaR Calculation for each option & overall portfolio in Python (13/15)

```python
# 1d VaR and 99,9% interval (Z-score=3.291), calculated on 1y
var = (total_delta*3.291*np.sqrt(1/252)*vol - total_gamma/2*(3.291*np.sqrt(1/252)*vol)**2)*np.sqrt(252)
total_portfolio["var"] = var
```

Appendix 25: Delta/ gamma & VaR Calculation for each option & overall portfolio in Python (14/15)

```python
elif key == "Autocall":
    notional = options_type[key]["notional"]
    tickerrs = options_type[key]["underlyings"]
    weights = options_type[key]["weights"]
    data_ticks = vola_data(tickerrs)
    end_date = len(data_ticks.loc[:today]) #determine lenght of data frame for vola --> 30d volatility since effective date or today?
    start_date = end_date - 30
    used_data = data_ticks.iloc[start_date:end_date]

    returns = log_returns(used_data)
    covar = returns.cov() * 12 #annualize monthly covariance

    vol = np.sqrt(np.dot(weights.T, np.dot(covar, weights)))
    options_type[key]["vol"] = vol

    delta = options_type[key]["delta"]
    g = options_type[key]["gamma"]
    # 1d VaR and 99,9% interval (Z-score=3.291), calculated on 1y
    var = (delta*3.291*np.sqrt(1/252)*vol - g/2*(3.291*np.sqrt(1/252)*vol)**2)*np.sqrt(252)
    options_type[key]["var"] = var
```

Appendix 26: Delta/ gamma & VaR Calculation for each option & overall portfolio in Python (15/15)

```python
class Autocall():

    def __init__(self,portfolio,id):
        self.id = id
        self.portfolio = portfolio


    def cholesky(self):

        def import_stock_data(tickers):
            data = pd.DataFrame()
            for t in tickers:
                data[t] = file_stocks[t].iloc[1:] #momentan mit iloc[1:]
            return(data)

        def get_timeseries(data_frame):
            data = data_frame
            date = self.portfolio[self.id]["effective_date"]
            row_number = len(data.loc[:date])
            maturity_date = self.portfolio[self.id]["maturity"]

            date = date.strftime("%Y-%m-%d")
            maturity_date = maturity_date.strftime("%Y-%m-%d")

            delta = np.busday_count(date, maturity_date)

            #today = datetime.today()
            #today = today.strftime("%Y-%m-%d")
            today = '2022-06-12'

            #print(delta)
            #2*252 --> 2 Years, Business Days, "take is the life of the option at start (2 years)"
            if delta >= 504:
                calc = 504
```

Appendix 27: Autocall class in Python (1/13)

```python
            else:
                #"From there on you calculate it decreasing with every day"
                delta_today = np.busday_count(date, today)
                #print(delta_today)
                days_used = 504 - delta_today
                #"until you reach the pre-defined minimum of 30 days
                # from which you keep calculating it on a 30-day basis until the option expires"
                minimum = 30
                calc = max(minimum, days_used)

            d1 = row_number - calc
            data = data.iloc[d1:row_number]
            return data


        ticks = self.portfolio[self.id]["underlyings"]
        num_stocks = len(ticks)
        data = import_stock_data(ticks)
        #print(f"full data: {data}")
        modified_data = get_timeseries(data)
        #print(f"modified data: {modified_data}")
        #print()

        def log_returns(data):
            return (np.log(1+data.pct_change()))

        log_return = log_returns(data)
```

Appendix 28: Autocall class in Python (2/13)

```python
def drift_calc(data, return_type='log'): #adjust for dividends
    if return_type == 'log':
        lr = log_returns(data)
    elif reutrn_type == 'simple':
        lr = simple_returns(data)
    u = lr.mean()
    var = lr.var()
    drift = u-(0.5*var)
    try:
        return drift.values
    except:
        return drift

drift = drift_calc(modified_data) #QUESTION: use data or modified data for drift

div = portfolio[self.id]['div']
# Drift adjusting if dividend paying (for Brownsche Motion)
if div > 0:
    drift = drift - div
#print(drift)
#print()
```

Appendix 29: Autocall class in Python (3/13)

```python
log_ret = log_returns(modified_data)

stdev = log_returns(modified_data).std().values
covar = log_ret.cov() #covariance matrix.
#print(covar)
#print()

chol = np.linalg.cholesky(covar) #create cholesky matrix from covariance matrix
#print(chol)
#print()

uncorr_x = norm.ppf(np.random.rand(num_stocks, simulated_days)) #stocks, days
#print(uncorr_x)
#print()

corr_x = np.dot(chol, uncorr_x)
#print(corr_x)
#print()

#Return an array of zeros with the same shape and type as a given array.
corr_2 = np.zeros_like(corr_x)
for i in range(num_stocks):
    corr_2[i] = np.exp(drift[i] + corr_x[i])
corr_2[0]
```

Appendix 30: Autocall class in Python (4/13)

```python
stock0 = pd.DataFrame()  #create new data frame
for s in range(len(ticks)):
    ret_reshape = corr_2[s]
    #Gives a new shape to an array without changing its data
    ret_reshape = ret_reshape.reshape(simulated_days)
    price_list = np.zeros_like(ret_reshape)
    #iloc = Purely integer-location based indexing for selection by position
    price_list[0] = data.iloc[-1, s]
    for t in range(1, simulated_days):
        price_list[t] = price_list[t-1]*ret_reshape[t]


    x = pd.DataFrame(price_list).iloc[-1]
    #fig, ax = plt.subplots(1,2, figsize=(14,4))
    #sns.distplot(x, ax=ax[0])
    #sns.distplot(x, hist_kws={'cumulative':True},kde_kws={'cumulative':True},ax=ax[1])
    #plt.xlabel("Stock Price")
    #plt.show()
    x = pd.DataFrame(price_list)
    stock0[ticks[s]]=x.loc[:,0]


last_date = data.last_valid_index()
last_date = pd.Timestamp(last_date)
dates = [last_date]
a = 1
while len(stock0) != len(dates):
    next_date = last_date + timedelta(days = a)
    if next_date.weekday() < 5:
        dates.append(next_date)
        a += 1
    else:
        a+=1
```

Appendix 31: Autocall class in Python (5/13)

```python
stock0["Date"] = dates
stock0 = stock0.set_index("Date")

output_stocks_combined = data.append(stock0[1:]) #create one big data frame with complete data

return output_stocks_combined
```

Appendix 32: Autocall class in Python (6/13)

```python
def autocall(self, first_price, second_price, third_price, expiration_price, strikes):

    cap = portfolio[self.id]["cap_auto"]
    first_coupon = portfolio[self.id]["first_coupon_auto"]
    second_coupon = portfolio[self.id]["second_coupon_auto"]
    third_coupon = portfolio[self.id]["third_coupon_auto"]
    #notional = portfolio[self.id]["notional"]
    notional = 1

    w = 0 #zählen der Stocks über Strike am 1. Stichtag
    x = 0 #zählen der Stocks über Strike am 2. Stichtag
    y = 0 #zählen der Stocks über Strike am 3. Stichtag
    z = 0 #zählen der Stocks über Strike am Maturity Date

    for u in range(len(strikes)):

        hasAutoCalled = False
        if hasAutoCalled: break

        if first_price[u] >= strikes[u]: #1. Stichtag
            w += 1

        if second_price[u] >= strikes[u]: #2. Stichtag
            x += 1

        if third_price[u] >= strikes[u]: #3. Stichtag
            y += 1

        if expiration_price[u] >= strikes[u]: #4. Stichtag
            z += 1

    #print(f"w:{w},x: {x},y: {y},z: {z}")
```

Appendix 33: Autocall class in Python (7/13)

```python
    if w == 5:
        hasAutoCalled = True
        payoff = notional*(1 + first_coupon)
        return payoff

    elif x == 5:
        hasAutoCalled = True
        payoff = notional*(1 + second_coupon)
        return payoff


    elif y == 5:
        hasAutoCalled = True
        payoff = notional*(1 + third_coupon)
        return payoff

    elif z == 5:
        hasAutoCalled = True
        payoff = notional*(1 + cap)
        return payoff

    else:
        payoff = notional
        return payoff
```

Appendix 34: Autocall class in Python (8/13)

```python
    def payoff(self):
        ticks = self.portfolio[self.id]["underlyings"]
        maturity_date = portfolio[self.id]['maturity']
        payoffs = []

        payoffs_pos = []
        payoffs_pos2 = []

        payoffs_neg = []
        payoffs_neg2 = []

        for i in range(number_simulations):
            output = self.cholesky()



            def dates_to_list(data_frame):
                date_list = data_frame.index.tolist()
                return date_list

            def observation_dates(date_list):
                firstobservationDate = 126 #wahrscheinlich eleganter: T=252 (Handelstage pro Jahr) und dann T/2
                secondobservationDate = 252 #wahrscheinlich eleganter: T=252 (Handelstage pro Jahr) und dann T
                thirdobservationDate = 378 #wahrscheinlich eleganter: T=252 (Handelstage pro Jahr) und dann T * 1.5
                for i in range(len(date_list)):
                    if date_list[i] == portfolio[self.id]["effective_date"]:
                        firstobservationDate = dates_list[i+ firstobservationDate]
                        secondobservationDate = dates_list[i+ secondobservationDate]
                        thirdobservationDate = dates_list[i+ thirdobservationDate]
                return firstobservationDate, secondobservationDate, thirdobservationDate

            dates_list = dates_to_list(output)
            valuation_date = observation_dates(dates_list)
```

Appendix 35: Autocall class in Python (9/13)

```python
first_stock_price = []
second_stock_price = []
third_stock_price = []
expiration_price = []

for k in ticks:
    firstobservationDate = valuation_date[0]
    first_stock = output._get_value(firstobservationDate, k)
    first_stock_price.append(first_stock)

    secondobservationDate = valuation_date[1]
    second_stock = output._get_value(secondobservationDate, k)
    second_stock_price.append(second_stock)

    thirdobservationDate = valuation_date[2]
    third_stock = output._get_value(thirdobservationDate, k)
    third_stock_price.append(third_stock)

    maturity_value = output._get_value(maturity_date, k)
    expiration_price.append(maturity_value)

#print(f"dates: {valuation_date}, {maturity_date}")
#print(first_stock_price)
#print(second_stock_price)
#print(third_stock_price)
#print(expiration_price)
```

Appendix 36: Autocall class in Python (10/13)

```python
first_stock_price_pos = []
second_stock_price_pos = []
third_stock_price_pos = []
expiration_stock_price_pos = []

first_stock_price_pos2 = []
second_stock_price_pos2 = []
third_stock_price_pos2 = []
expiration_stock_price_pos2 = []

first_stock_price_neg = []
second_stock_price_neg = []
third_stock_price_neg = []
expiration_stock_price_neg = []

first_stock_price_neg2 = []
second_stock_price_neg2 = []
third_stock_price_neg2 = []
expiration_stock_price_neg2 = []
```

Appendix 37: Autocall class in Python (11/13)

```python
        for k in range(len(ticks)):
            first_pos = first_stock_price[k] * (1 + percentage_change)
            first_pos2 = first_pos * (1 + percentage_change)
            first_stock_price_pos.append(first_pos)
            first_stock_price_pos2.append(first_pos2)

            second_pos = second_stock_price[k] * (1 + percentage_change)
            second_pos2 = second_pos * (1 + percentage_change)
            second_stock_price_pos.append(second_pos)
            second_stock_price_pos2.append(second_pos2)

            third_pos = third_stock_price[k] * (1 + percentage_change)
            third_pos2 = third_pos * (1 + percentage_change)
            third_stock_price_pos.append(third_pos)
            third_stock_price_pos2.append(third_pos2)

            expiration_pos = expiration_price[k] * (1 + percentage_change)
            expiration_pos2 = expiration_pos * (1 + percentage_change)
            expiration_stock_price_pos.append(expiration_pos)
            expiration_stock_price_pos2.append(expiration_pos2)

            first_neg = first_stock_price[k] * (1 - percentage_change)
            first_neg2 = first_neg * (1 - percentage_change)
            first_stock_price_neg.append(first_neg)
            first_stock_price_neg2.append(first_neg2)

            second_neg = second_stock_price[k] * (1 - percentage_change)
            second_neg2 = second_neg * (1 - percentage_change)
            second_stock_price_neg.append(second_neg)
            second_stock_price_neg2.append(second_neg2)

            third_neg = third_stock_price[k] * (1 - percentage_change)
            third_neg2 = third_neg * (1 - percentage_change)
            third_stock_price_neg.append(third_neg)
            third_stock_price_neg2.append(third_neg2)
```

Appendix 38: Autocall class in Python (12/13)

```
                expiration_neg = expiration_price[k] * (1 - percentage_change)
                expiration_neg2 = expiration_neg * (1 - percentage_change)
                expiration_stock_price_neg.append(expiration_neg)
                expiration_stock_price_neg2.append(expiration_neg2)

            normal_payoff = self.autocall(first_stock_price, second_stock_price, third_stock_price,
                                          expiration_price, self.portfolio[self.id]["strikes"])
            payoffs.append(normal_payoff)

            pos_payoff = self.autocall(first_stock_price_pos, second_stock_price_pos, third_stock_price_pos,
                                       expiration_stock_price_pos, self.portfolio[self.id]["strikes"])
            payoffs_pos.append(pos_payoff)

            neg_payoff = self.autocall(first_stock_price_neg, second_stock_price_neg, third_stock_price_neg,
                                       expiration_stock_price_neg, self.portfolio[self.id]["strikes"])
            payoffs_neg.append(neg_payoff)


            #Gamma-Calculation-Payoffs
            pos_payoff2 = self.autocall(first_stock_price_pos2, second_stock_price_pos2, third_stock_price_pos2,
                                        expiration_stock_price_pos2, self.portfolio[self.id]["strikes"])
            payoffs_pos2.append(pos_payoff2)

            neg_payoff2 = self.autocall(first_stock_price_neg2, second_stock_price_neg2, third_stock_price_neg2,
                                        expiration_stock_price_neg2, self.portfolio[self.id]["strikes"])
            payoffs_neg2.append(neg_payoff2)


        average_payoff = sum(payoffs)/len(payoffs)
        average_pos = sum(payoffs_pos)/len(payoffs_pos)
        average_neg = sum(payoffs_neg)/len(payoffs_neg)
        average_pos2 = sum(payoffs_pos2)/len(payoffs_pos2)
        average_neg2 = sum(payoffs_neg2)/len(payoffs_neg2)

        return [average_payoff, average_pos, average_neg, average_pos2, average_neg2]
```

Appendix 39: Autocall class in Python (13/13)

Autocall Product ID 1015 -Invest Health & Tech Jun-20

Underlying	ID	Name Underlying	Bloomberg Ticker	Industry	Asset Class	Strike	Price as of 30/06/2022
1	103	GSK PLC	GSK LN Equity	Health Technology	Equity	1.636,60 €	1.786,06 €
2	185	Nestle	NESN SW Equity	Consumer Non-Durables	Equity	104,74 €	111,44 €
3	424	SAP SE O.N.	SAP GY Equity	Technology Services	Equity	124,32 €	86,93 €
4	750	ASML Holding	ASML NA Equity	Electronic Technology	Equity	326,90 €	455,85 €
5	1345	Novo Nordisk	NVO US Equity	Health Technology	Equity	65,48 €	111,43 €

trade date:	30/06/2020	103	185	424	750	1345
ID	Underlying	GSK LN Equity	NESN SW Equity	SAP GY Equity	ASML NA Equity	NVO US Equity
103	GSK LN Equity	1,00	0,56	0,39	0,40	0,43
185	NESN SW Equity	0,56	1,00	0,42	0,48	0,41
424	SAP GY Equity	0,39	0,42	1,00	0,64	0,39
750	ASML NA Equity	0,40	0,48	0,64	1,00	0,29
1345	NVO US Equity	0,43	0,41	0,39	0,29	1,00

Appendix 40: Overview example Autocall option (ID 1015)

Autocall Product ID 1033 -Invest Personal Care Ago-20

Underlying	ID	Name Underlying	Bloomberg Ticker	Industry	Asset Class	Strike	Price as of 30/06/2022
1	747	British American Tobacco	BATS LN Equity	Health Technology	Equity	2.515,00 €	3.519,50 €
2	797	Danone	BN FP Equity	Consumer Non-Durables	Equity	54,52 €	53,26 €
3	798	Unilever	UNA NA Equity	Consumer Non-Durables	Equity	48,69 €	43,32 €
4	1152	Procter & Gamble	PG US Equity	Consumer Non-Durables	Equity	138,18 €	143,79 €
5	1362	Rocket Companies	RKT LN Equity	Finance	Equity	7.440,00 €	6.170,00 €

trade date:	01/09/2020	747	797	798	1152	1362
ID	**Underlying**	BATS LN Equity	BN FP Equity	UNA NA Equity	PG US Equity	RKT LN Equity
747	BATS LN Equity	1,00	0,43	0,46	0,36	0,40
797	BN FP Equity	0,43	1,00	0,66	0,34	0,54
798	UNA NA Equity	0,46	0,66	1,00	0,33	0,63
1152	PG US Equity	0,36	0,34	0,33	1,00	0,43
1362	RKT LN Equity	0,40	0,54	0,63	0,43	1,00

Appendix 41: Overview example Autocall option (ID 1033)

Autocall Product ID 1041 -Invest Back to School Set-20

Underlying	ID	Name Underlying	Bloomberg Ticker	Industry	Asset Class	Strike	Price as of 30/06/2022
1	101	Adidas AG	ADS GY Equity	Consumer Non-Durables	Equity	276,10 €	168,76 €
2	185	Nestle	NESN SW Equity	Consumer Non-Durables	Equity	109,34 €	111,44 €
3	358	Alphabet Inc.	GOOGL US Equity	Technology Services	Equity	1.465,60 €	108,96 €
4	360	HP Inc.	HPQ US Equity	Electronic Technology	Equity	18,99 €	32,78 €
5	848	Nokia Corporation	NOKIA FH Equity	Electronic Technology	Equity	3,35 €	4,44 €

trade date:	30/09/2020	101	185	358	360	848
ID	**Underlying**	ADS GY Equity	NESN SW Equity	GOOGL US Equity	HPQ US Equity	NOKIA FH Equity
101	ADS GY Equity	1,00	0,44	0,35	0,36	0,48
185	NESN SW Equity	0,44	1,00	0,35	0,31	0,29
358	GOOGL US Equity	0,35	0,35	1,00	0,59	0,32
360	HPQ US Equity	0,36	0,31	0,59	1,00	0,38
848	NOKIA FH Equity	0,48	0,29	0,32	0,38	1,00

Appendix 42: Overview example Autocall option (ID 1041)

Autocall Product ID 1130 -Invest Digital 5G Fev-21

Underlying	ID	Name Underlying	Bloomberg Ticker	Industry	Asset Class	Strike	Price as of 30/06/2022
1	64	AT&T Inc.	T US Equity	Communications	Equity	21,05 €	20,96 €
2	183	Vodafone Group	VOD LN Equity	Communications	Equity	122,02 €	126,66 €
3	381	Broadcom Inc.	AVGO US Equity	Electronic Technology	Equity	469,87 €	485,81 €
4	1009	Intel Corporation	INTC US Equity	Electronic Technology	Equity	60,78 €	37,41 €
5	1347	Crown Castle Inc.	CCI US Equity	Finance	Equity	155,75 €	168,38 €

trade date:	26/02/2021	64	183	381	1009	1347
ID	**Underlying**	T US Equity	VOD LN Equity	AVGO US Equity	INTC US Equity	CCI US Equity
64	T US Equity	1,00	0,51	0,51	0,55	0,57
183	VOD LN Equity	0,51	1,00	0,43	0,38	0,28
381	AVGO US Equity	0,51	0,43	1,00	0,65	0,58
1009	INTC US Equity	0,55	0,38	0,65	1,00	0,48
1347	CCI US Equity	0,57	0,28	0,58	0,48	1,00

Appendix 43: Overview example Autocall option (ID 1130)

Autocall Product ID 1152 -Invest Natural Resources Abr-21

Underlying	ID	Name Underlying	Bloomberg Ticker	Industry	Asset Class	Strike	Price as of 30/06/2022
1	62	Royal Dutch Shell PLC	SHELL NA Equity	Energy Minerals	Equity	15,84 €	24,85 €
2	81	Totalenergies SE	TTE FP Equity	Energy Minerals	Equity	36,83 €	50,37 €
3	164	Vale S.A.	VALE US Equity	Non-Energy Minerals	Equity	20,12 €	14,63 €
4	1384	BHP Group Limited	BHP LN Equity	Non-Energy Minerals	Equity	1.975,80 €	2.297,00 €
5	1457	Weyerhaeuser Company	WY US Equity	Finance	Equity	38,77 €	33,12 €

trade date:	30/04/2021	62	81	164	1384	1457
ID	Underlying	SHELL NA Equity	TTE FP Equity	VALE US Equity	BHP LN Equity	WY US Equity
62	SHELL NA Equity	1,00	0,88	0,36	0,53	0,38
81	TTE FP Equity	0,88	1,00	0,35	0,52	0,35
164	VALE US Equity	0,36	0,35	1,00	0,51	0,42
1384	BHP LN Equity	0,53	0,52	0,51	1,00	0,33
1457	WY US Equity	0,38	0,35	0,42	0,33	1,00

Appendix 44: Overview example Autocall option (ID 1152)

Autocall Product ID 1190 -Invest Fintech Jul-21

Underlying	ID	Name Underlying	Bloomberg Ticker	Industry	Asset Class	Strike	Price as of 30/06/2022
1	359	Cisco Systems Inc.	CSCO US Equity	Electronic Technology	Equity	55,37 €	42,64 €
2	377	Lloyds Banking Group	LLOY LN Equity	Finance	Equity	45,64 €	42,31 €
3	424	SAP SE O.N.	SAP GY Equity	Technology Services	Equity	120,84 €	86,93 €
4	748	ING Groep N.V.	INGA NA Equity	Finance	Equity	10,85 €	9,43 €
5	1110	Block Inc.	SQ US Equity	Technology Services	Equity	247,26 €	61,46 €

trade date:	30/07/2021	359	377	424	748	1110
ID	Underlying	CSCO US Equity	LLOY LN Equity	SAP GY Equity	INGA NA Equity	SQ US Equity
359	CSCO US Equity	1,00	0,29	0,30	0,23	0,08
377	LLOY LN Equity	0,29	1,00	0,16	0,73	-0,13
424	SAP GY Equity	0,30	0,16	1,00	0,15	0,18
748	INGA NA Equity	0,23	0,73	0,15	1,00	-0,15
1110	SQ US Equity	0,08	-0,13	0,18	-0,15	1,00

Appendix 45: Overview example Autocall option (ID 1190)

Autocall Product ID 1203 -Invest Health Innovation Ago-21

Underlying	ID	Name Underlying	Bloomberg Ticker	Industry	Asset Class	Strike	Price as of 30/06/2022
1	82	Banco Santander S.A.	SAN FP Equity	Finance	Equity	87,15 €	96,34 €
2	83	Johson & Johnson	JNJ US Equity	Health Technology	Equity	173,13 €	177,51 €
3	104	Bayer AG	BAYN GY Equity	Health Technology	Equity	47,15 €	56,72 €
4	1272	Biogen Inc.	BIIB US Equity	Health Technology	Equity	338,91 €	203,94 €
5	1490	Medtronic PLC.	MDT US Equity	Health Technology	Equity	133,48 €	89,75 €

trade date:	31/08/2021	82	83	104	1272	1490
ID	Underlying	SAN FP Equity	JNJ US Equity	BAYN GY Equity	BIIB US Equity	MDT US Equity
82	SAN FP Equity	1,00	0,28	0,39	0,15	0,22
83	JNJ US Equity	0,28	1,00	0,27	0,01	0,48
104	BAYN GY Equity	0,39	0,27	1,00	-0,02	0,36
1272	BIIB US Equity	0,15	0,01	-0,02	1,00	-0,16
1490	MDT US Equity	0,22	0,48	0,36	-0,16	1,00

Appendix 46: Overview example Autocall option (ID 1203)

Autocall Product ID 1204 -Invest Back to School Set-21

Underlying	ID	Name Underlying	Bloomberg Ticker	Industry	Asset Class	Strike	Price as of 30/06/2022
1	101	Adidas AG	ADS GY Equity	Consumer Non-Durables	Equity	271,80 €	168,76 €
2	185	Nestle	NESN SW Equity	Consumer Non-Durables	Equity	112,70 €	111,44 €
3	358	Alphabet Inc.	GOOGL US Equity	Technology Services	Equity	2.673,52 €	108,96 €
4	360	HP Inc.	HPQ US Equity	Electronic Technology	Equity	27,36 €	32,78 €
5	848	Nokia Corporation	NOKIA FH Equity	Electronic Technology	Equity	4,76 €	4,44 €

trade date:	30/09/2021	101	185	358	360	848
ID	Underlying	ADS GY Equity	NESN SW Equity	GOOGL US Equity	HPQ US Equity	IOKIA FH Equity
101	ADS GY Equity	1,00	0,22	0,12	0,14	0,14
185	NESN SW Equity	0,22	1,00	0,19	-0,11	0,18
358	GOOGL US Equity	0,12	0,19	1,00	0,26	-0,03
360	HPQ US Equity	0,14	-0,11	0,26	1,00	0,02
848	NOKIA FH Equity	0,14	0,18	-0,03	0,02	1,00

Appendix 47: Overview example Autocall option (ID 1204)

Autocall Product ID 1229 -Invest Infraestruturas Globais Out-21

Underlying	ID	Name Underlying	Bloomberg Ticker	Industry	Asset Class	Strike	Price as of 30/06/2022
1	64	AT&T Inc.	T US Equity	Communications	Equity	19,07 €	20,96 €
2	178	Acciones Iberdrola	IBE SM Equity	Utilities	Equity	10,22 €	9,90 €
3	184	Siemens AG	SIE GY Equity	Health Technology	Equity	140,28 €	97,09 €
4	209	Enel	ENEL IM Equity	Utilities	Equity	7,24 €	5,22 €
5	1113	Dollar General Corporation	DG FP Equity	Retail Trade	Equity	92,37 €	84,96 €

trade date:	29/10/2021	64	178	184	209	1113
ID	Underlying	T US Equity	IBE SM Equity	SIE GY Equity	ENEL IM Equity	DG FP Equity
64	T US Equity	1,00	0,06	0,22	0,17	0,19
178	IBE SM Equity	0,06	1,00	0,26	0,72	0,28
184	SIE GY Equity	0,22	0,26	1,00	0,34	0,31
209	ENEL IM Equity	0,17	0,72	0,34	1,00	0,38
1113	DG FP Equity	0,19	0,28	0,31	0,38	1,00

Appendix 48: Overview example Autocall option (ID 1229)

Autocall Product ID 1244 -Invest Hydrogen Nov-21

Underlying	ID	Name Underlying	Bloomberg Ticker	Industry	Asset Class	Strike	Price as of 30/06/2022
1	178	Acciones Iberdrola	IBE SM Equity	Utilities	Equity	9,90 €	9,90 €
2	395	BASF SE	BAS GY Equity	Process Industries	Equity	57,88 €	41,53 €
3	1331	C3.AI Inc.	AI FP Equity	Technology Services	Equity	132,56 €	128,12 €
4	1435	Linde PLC	LIN GY Equity	Process Industries	Equity	281,75 €	273,95 €
5	1438	Air Products and Chemicals Inc.	APD US Equity	Process Industries	Equity	287,44 €	240,48 €

trade date:	30/11/2021	178	395	1331	1435	1438
ID	Underlying	IBE SM Equity	BAS GY Equity	AI FP Equity	LIN GY Equity	APD US Equity
178	IBE SM Equity	1,00	0,20	0,42	0,45	0,14
395	BAS GY Equity	0,20	1,00	0,32	0,50	0,21
1331	AI FP Equity	0,42	0,32	1,00	0,59	0,17
1435	LIN GY Equity	0,45	0,50	0,59	1,00	0,30
1438	APD US Equity	0,14	0,21	0,17	0,30	1,00

Appendix 49: Overview example Autocall option (ID 1244)

Autocall Product ID 1260 -Invest Communication & Media Dez-21

Underlying	ID	Name Underlying	Bloomberg Ticker	Industry	Asset Class	Strike	Price as of 30/06/2022
1	65	Telefonica S.A.	TEF SM Equity	Communications	Equity	3,85 €	4,86 €
2	114	Alphabet Inc.	GOOG US Equity	Technology Services	Equity	2.920,05 €	109,37 €
3	165	Meta Platforms Inc.	META US Equity	Technology Services	Equity	344,36 €	161,25 €
4	180	Orange	ORA FP Equity	Communications	Equity	9,43 €	11,22 €
5	1520	Paramount Global	PARA US Equity	Consumer Services	Equity	31,12 €	24,68 €

trade date:	30/12/2021	65	114	165	180	1520
ID	Underlying	TEF SM Equity	GOOG US Equity	META US Equity	ORA FP Equity	PARA US Equity
65	TEF SM Equity	1,00	0,05	0,02	0,56	0,09
114	GOOG US Equity	0,05	1,00	0,57	0,06	-0,02
165	META US Equity	0,02	0,57	1,00	0,06	0,06
180	ORA FP Equity	0,56	0,06	0,06	1,00	0,08
1520	PARA US Equity	0,09	-0,02	0,06	0,08	1,00

Appendix 50: Overview example Autocall option (ID 1260)

Autocall Product ID 1348 -Invest Metaverse Mar-22

Underlying	ID	Name Underlying	Bloomberg Ticker	Industry	Asset Class	Strike	Price as of 30/06/2022
1	115	Microsoft Corporation	MSFT US Equity	Technology Services	Equity	308,31 €	256,83 €
2	116	International Business Machines Corporation	IBM US Equity	Technology Services	Equity	130,02 €	141,19 €
3	165	Meta Platforms Inc.	META US Equity	Technology Services	Equity	222,36 €	161,25 €
4	750	ASML Holding N.V.	ASML NA Equity	Electronic Technology	Equity	610,00 €	455,85 €
5	1498	Tencent Holding Ltd.	TCEHY US Equity	Technology Services	Equity	46,42 €	45,39 €

trade date:	31/03/2022	115	116	165	750	1498
ID	Underlying	MSFT US Equity	IBM US Equity	META US Equity	ASML NA Equity	CEHY US Equity
115	MSFT US Equity	1,00	0,08	0,55	0,46	0,23
116	IBM US Equity	0,08	1,00	0,09	0,04	0,01
165	META US Equity	0,55	0,09	1,00	0,30	0,24
750	ASML NA Equity	0,46	0,04	0,30	1,00	0,30
1498	TCEHY US Equity	0,23	0,01	0,24	0,30	1,00

Appendix 51: Overview example Autocall option (ID 1348)

Autocall Product ID 1392 -Invest Basic Resources Abr-22

Underlying	ID	Name Underlying	Bloomberg Ticker	Industry	Asset Class	Strike	Price as of 30/06/2022
1	81	Totalenergies	TTE FP Equity	Energy Minerals	Equity	47,23 €	50,37 €
2	97	Veolia Environ	VIE FP Equity	Utilities	Equity	27,90 €	23,29 €
3	123	Rio Tinto PLC	RIO LN Equity	Non-Energy Minerals	Equity	5.706,00 €	4.916,50 €
4	164	Vale S.A.	VALE US Equity	Non-Energy Minerals	Equity	16,89 €	14,63 €
5	1457	Weyerhaeuser Company	WY US Equity	Finance	Equity	41,22 €	33,12 €

trade date:	29/04/2022	81	97	123	164	1457
ID	Underlying	TTE FP Equity	VIE FP Equity	RIO LN Equity	VALE US Equity	WY US Equity
81	TTE FP Equity	1,00	0,29	0,41	0,25	0,13
97	VIE FP Equity	0,29	1,00	0,05	-0,09	0,12
123	RIO LN Equity	0,41	0,05	1,00	0,58	0,12
164	VALE US Equity	0,25	-0,09	0,58	1,00	0,21
1457	WY US Equity	0,13	0,12	0,12	0,21	1,00

Appendix 52: Overview example Autocall option (ID 1392)

Autocall Product ID 1429 -Invest Technology Jun-22

Underlying	ID	Name Underlying	Bloomberg Ticker	Industry	Asset Class	Strike	Price as of 30/06/2022
1	159	Apple Inc.	AAPL US Equity	Electronic Technology	Equity	139,23 €	136,72 €
2	424	SAP SE O.N.	SAP GY Equity	Technology Services	Equity	90,17 €	86,93 €
3	755	Dassault Systemes	DSY FP Equity	Technology Services	Equity	36,09 €	35,12 €
4	1009	Intel Corporation	INTC US Equity	Electronic Technology	Equity	37,29 €	37,41 €
5	1110	Block Inc.	SQ US Equity	Technology Services	Equity	63,84 €	61,46 €

trade date:	30/06/2022	159	424	755	1009	1110
ID	Underlying	AAPL US Equity	SAP GY Equity	DSY FP Equity	INTC US Equity	SQ US Equity
159	AAPL US Equity	1,00	0,28	0,42	0,57	0,60
424	SAP GY Equity	0,28	1,00	0,60	0,27	0,31
755	DSY FP Equity	0,42	0,60	1,00	0,29	0,43
1009	INTC US Equity	0,57	0,27	0,29	1,00	0,50
1110	SQ US Equity	0,60	0,31	0,43	0,50	1,00

Appendix 53: Overview example Autocall option (ID 1429)

Autocall Product ID 1455 -Invest Blockchain Jul-22

Underlying	ID	Name Underlying	Bloomberg Ticker	Industry	Asset Class	Strike	Price as of 30/06/2022
1	59	Banco Bilbao Vizcaya Argentaria S.A.	BBVA SM Equity	Finance	Equity	5,01 €	4,33 €
2	115	Microsoft Corporation	MSFT US Equity	Technology Services	Equity	272,42 €	256,83 €
3	116	International Business Machines Corporation	IBM US Equity	Technology Services	Equity	139,43 €	141,19 €
4	149	BNP Paribas	BNP FP Equity	Finance	Equity	52,81 €	45,37 €
5	1144	Paypal Holdings Inc.	PYPL US Equity	Commercial Services	Equity	82,48 €	69,84 €

trade date:	29/07/2022	59	115	116	149	1144
ID	Underlying	BBVA SM Equity	MSFT US Equity	IBM US Equity	BNP FP Equity	PYPL US Equity
59	BBVA SM Equity	1,00	0,27	0,18	0,75	0,25
115	MSFT US Equity	0,27	1,00	0,21	0,18	0,52
116	IBM US Equity	0,18	0,21	1,00	0,22	0,13
149	BNP FP Equity	0,75	0,18	0,22	1,00	0,24
1144	PYPL US Equity	0,25	0,52	0,13	0,24	1,00

Appendix 54: Overview example Autocall option (ID 1455)