Final Project Report

Full Unit – Final Report

Value at Risk Implementation in Python

Raphael Frach

A report submitted in part fulfilment of the degree of

BSc (Hons) in Computer Science

Supervisor: Dr Yuri Kalnishkan

Department of Computer Science
Royal Holloway, University of London

March 30, 2023

Declaration

This report has been prepared on the basis of my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count: 14728

Student Name: Raphael N. Frach

Date of Submission: 30/03/2023

Signature:

Table of Contents

Abstract

Value at Risk is a statistic that returns a single figure which estimates the possible worst case monetary losses from a portfolio or position over a specified time frame at a given confidence level. In this report I will describe the mathematical background associated with different methods for estimating Value at Risk and Conditional Value at Risk.

In my project I will compute VaR in three main ways: historical simulation, model-building (parametric) and Monte Carlo simulation. These approaches will take parameters from real world data, essentially: Historical simulation will plot previous real-world data to find VaR, Parametric will use the Gaussian or Student's t-distribution with parameters (volatility estimates based of portfolio returns) while Monte Carlo will generate "future" prices based on random numbers sampled from statistical distributions and assuming that the daily returns follow on similarly to Geometric Brownian Motion.

The Monte Carlo implementation will be extended to allow multiple stocks using Cholesky decomposition, this will help us as we will be simulating a system with multiple correlated variables. The covariance matrix will be decomposed, and we will take the Lower Triangular Matrix and use this with our samples from the normal distribution as well as the mean to determine our daily returns. Additionally, we will incorporate European put or call options into our portfolio by simulating the underlying asset and seeing how the option's price changes.

It is also important to look at different estimates for volatility. In contrast to using standard deviation(Equal Weighting) we will calculate with the Exponentially Weighted Moving Average (EWMA) approach. EWMA gives us more weight to more recent returns. The VaR results will be discussed in the report.

Finally, we would like to see how reliable our estimates are. Therefore, we will perform some qualitative assessment then followed by more rigorous backtesting and stress testing.

Project Specification

Adapted from Moodle.

Aims:

Write a program for computing (Conditional) Value at Risk.

Background:

It is a very difficult and important problem to estimate the riskiness of a portfolio of securities. Value at Risk (VaR) is a tool for measuring financial risk. The goal of this project is to implement some of the methodologies for VaR computation and test it on the real financial data. Other monetary measures of risk may also be studied.

Early Deliverables:

We have written a simple proof-of-concept program for computing Value at Risk for a portfolio consisting of 1 or 2 stocks using two different methods: model-building and historical simulation.

Program extended to use Monte Carlo Simulation based off similar formula for Geometric Brownian Motion.

Monte Carlo extended to allow multiple stocks for portfolio using Cholesky Decomposition.

All of the above methods extended to allow calculation of Expected Shortfall (Conditional Value at Risk).

Model-building method extended to use Exponentially Weighted Moving Average volatility.

The report will describe the program (software engineering, algorithms etc.,).

The report will also discuss different ways of estimating the variance of returns and the covariance between returns of two different securities (using the standard formula, exponentially weighted moving average).

The report discusses all the theory behind the algorithms.

The report discusses some basic experiments with numerical results.

Final Deliverables:

Different ways of computing Value at Risk will be backtested and the statistical significance of the results analysed.

As well as a stress test being performed.

The program is extended by allowing derivatives (such as options) in the portfolio.

The Student’s t-distribution has been implemented with the model-building approach.

We have a simple graphical user interface.

The final report describes: the theory behind the algorithms.

The final report describes: the implementation issues necessary to apply the theory.

The final report describes: the software engineering process involved in generating your software.

The final report describes: computational experiments with different data sets, methods, and parameters.

Chapter 1: **Introduction**

In industry, risk managers use VaR so that they can monitor and control the level of risk exposure of investments the company is willing to take and determine their cash reserves. This will lead to critical decisions being made that will be discussed below.

It is a widely adopted technique for estimating risk due to it being easy to understand the single figure that is returned and its applicability to many types of assets - shares, derivatives, currencies, etc. VaR owes most of its popularity to Dennis Weatherstone, former chairman of JP Morgan & Co, Inc. He realised that it would be a great benefit if market traders were able to easily quantify risk to their senior managers and directors. They would be able to inform them that they do not expect to lose more than X amount of money on more than one day in the next month. For large investment firms like JP Morgan, it is used to quantify company-wide risk exposure rather than investment specific risk. This allows them to predict the size of future losses/gains and to be compared against their available capital and cash reserves to guarantee the losses can be covered without putting the firm at risk of bankruptcy. They also use it to establish the collateral needed from a client, for example when applying for a margin loan when trading financial instruments. The clients borrow money from the broker provided that the securities they already own can cover the VaR determined collateral. If the client's investments take a turn for the worse, the broker's borrowed money can get lost; however, at whichever confidence level was used, they can claim their money back from the client's pre-owned securities. Leaving the broker with most likely no losses.

Additionally, bank regulators can ask banks to reserve a multiple of the VaR for the period it was calculated. If a risk is realised, the bank has this cash reserve. This gives banks the ability to respond to unexpected events and prevent exodus or even bankruptcy by using the reserve to make unexpected payments. It helps to keep a peace of mind to make more rational decisions. We have seen this just recently during the COVID recession when the market dropped in 2020, no one knew when it would rebound. With patience and an eased mind many financial institutions were able to use the reserves to keep them afloat waiting for the moment the market turned up again.

Chapter 2: **Background Research**

2.1 Literature Review

Since Value at Risk's inception in the 1980's by JPMorgan, there have been a plethora of research papers on the topic. As well as this, many popular quantitative finance textbooks contain sections specifically for it. Notable authors are John C. Hull and Paul Wilmott. The fundamental ideas of Value at Risk described in each piece of literature hasn't changed, due to the nature of the simple and intuitive statistical approach it takes. [18]This also led to VaR's popularity as a metric being able to estimate losses of a portfolio in monetary terms.

As the basics stayed the same, over the years we have seen many different refinements and additions. As an example, in 2004, Angelidis et al.[19] observed the majority of VaR models assumed that the underlying assets returns followed the Gaussian distribution, although excess kurtosis and more skewed distributions are common amongst real stock's returns[19]. Therefore, the use of leptokurtic distributions, like the Student's t-dist., began to see use as they had fatter tails to compensate for the similarity amongst stock's returns.

Another focal point of research in VaR models has been volatility estimates, volatility being one parameter incorporated in all models. As opposed to simply using the standard deviation of returns to come to a volatility estimate, other techniques like the Equally Weighted Moving Average (EWMA) have been investigated. And more recently we have seen developments like [20]GARCH models, both giving more weight to the latest returns in volatility calculations, with the assumption that recent returns have a bigger influence on future volatility compared to the far past.

There are many different ways to create a VaR model, we can combine volatility estimates with various distributions etc. This has led to the Basel[21] regulatory framework which provides guidelines for managing risk, for the use in large financial institutions to help mitigate market risk. These standardised methods are important as different models can give different estimates over the same time period.

Finally, to ensure that a model is accurate it is important to backtest, running VaR calculations over historical market data and comparing against the actual losses sustained.

2.2 Value at Risk Definition

Value at Risk is defined as the maximum loss from a portfolio, in a given holding period, to a certain confidence level. Risk managers want to say, we are interested in all the periods where the loss surpasses a value to the percentile in which they are interested. This allows them to say with c% certainty that our portfolio losses will not exceed the Value at Risk.

In a real-world scenario, to illustrate the above saying, there will be a 95% chance that the profit is going to be above our 2000$ (VaR point) for the next 10 trading days. This give risk managers an idea of how much cash reserve needs to be held to overcome these unexpected losses. We can then recalculate daily and consider the stock investments in our portfolio.

Given a profit-loss distribution X of a portfolio, it is negative for loss and conversely positive for profit. To simplify, we can say that if X is a random variable for the profit-loss distribution then:

$Y \coloneqq -X$ defines the loss distribution. Where all values for loss are positive and vice versa for profit.

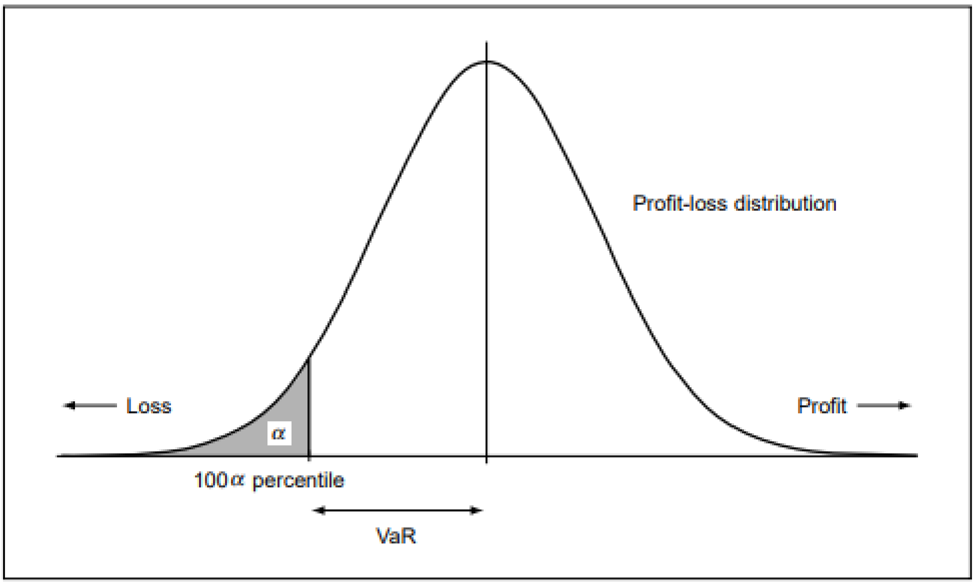


Figure 1 - profit-loss distribution[2]

Mathematically we can define it like this, where $F_X(x)$ is the cumulative distribution function of the profit-loss distribution and $F_Y^{-1}(y)$ is the inverse cumulative distribution function of the profit-loss distribution.

$$VaR_{\alpha}(X) = -\inf\{x \in R \mid F_X(x) > \alpha\} = F_Y^{-1}(1-\alpha)$$

Equation 1

So, in that sense, we use the formula above. inf{} takes the infimum of the set, which is the lowest bound of the set of values where $F_X(x) \geq \alpha$ and gives the lower bound point where $F_X(x) > \alpha$. Essentially, this gives us the loss limit under VaR where our confidence level is 100 - α.

It is all well and good to define the lower bound point, however this value might be hiding more information that we have available in the distribution. We still have a set of values past our lower bound that have not been taken into consideration, bringing us to our next sub-chapter Conditional Value at Risk Definition.

2.3 Conditional Value at Risk Definition

In this report we will call Conditional Value at Risk, Expected Shortfall for simplicity.

Following on, Expected Shortfall describes the expectation of the set of values that are past our VaR, lower bound. This lets risk managers propose the question, on the days where our profit-loss distribution is in that small percentile of loss, what is my expected loss on those days? They can then quantify the whole lower tail risk.

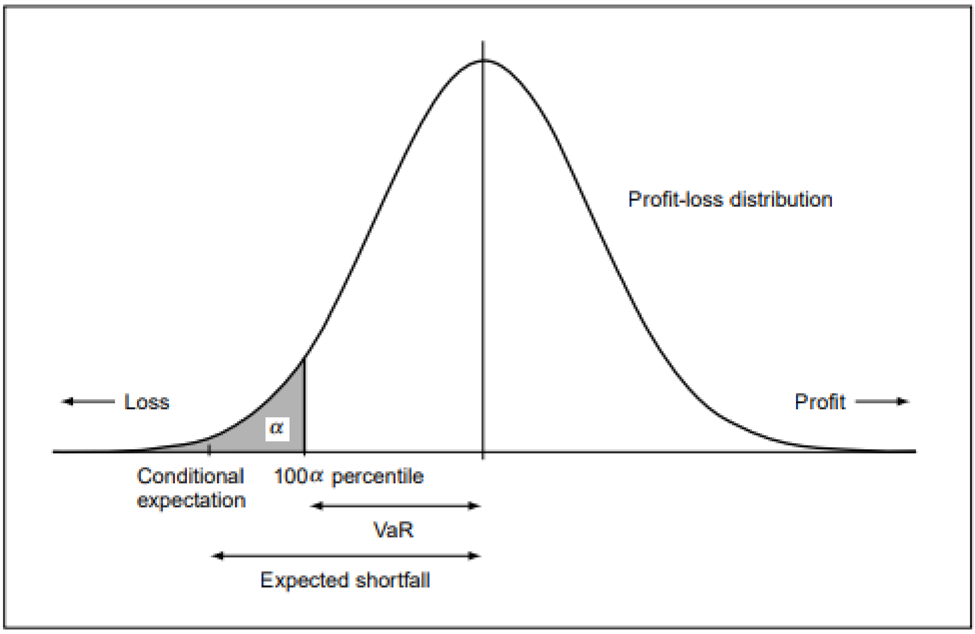


Figure 2 - profit-loss dist. with Expected Shortfall[2]

$$CVaR_{\alpha}(X) = E[-X \,|\, -X > VaR_{\alpha}(X)]$$

Equation 2

Formally, Expected Shortfall is defined above. It is the expected value of -X strictly exceeding VaR, it is just the mean loss of the lower tail set of values. From Figure 2, we see the Conditional expectation is our Expected Shortfall.

Our calculation of CVaR is dependent on our VaR assumptions such as: lower-bound point, volatility, and the shape of the profit-loss distribution.

The reason we are interested in Expected Shortfall is when our portfolio has not shown stability over time; VaR can be a good enough estimate for risk in the case of stable portfolio as it will be more indifferent beyond the threshold.

As an example, to illustrate a leading benefit of Expected Shortfall, we could have a historical profit-loss distribution with a lower tail that contains a higher number of scenarios where we experience greater losses than we would expect (for example from Gaussian distribution, it is not expected). Here our CVaR estimate will be a superior prediction as it considers all of those greater losses that our VaR would have missed. VaR on its own would lead to unanticipated risk.

A good illustration of the above paragraph can be seen in Chapter 16.1 from Options, Futures, and Other Derivatives, Hull[3].

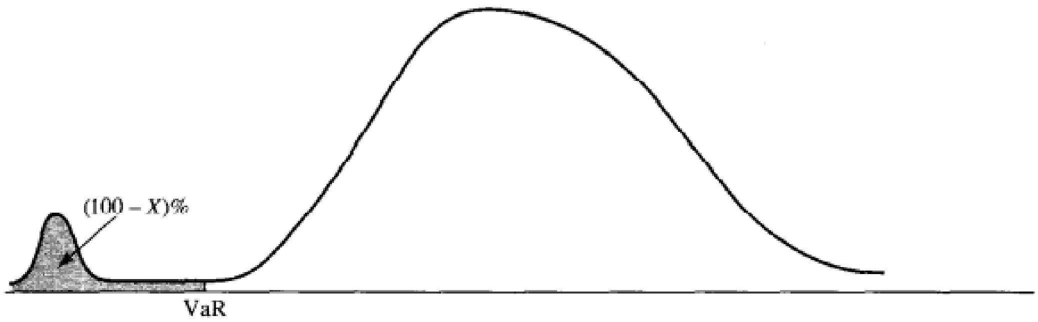


Figure 3- profit-loss dist. showing large tail of Expected Shortfall [3]

In the above figure we can see the lower tail has some unexpected losses we would not see if we just looked at the VaR estimate.

Now to talk about the different methods for computing Value at Risk and Expected Shortfall.

2.4 Historical Simulation

The Historical approach to calculating the VaR statistic is a simple non-parametric approach, it is widely adopted and uses historical market data to estimate future risk. We follow on from a similar approach Hull describes in Chapter 16.2[3]. However, instead of collecting data from multiple market variables like exchange rates or equity prices only daily stock prices are collected. Subsequently, we compute the daily price change/returns allowing us to define a probability distribution like we have described in the sub-sections above. Our belief is that probability distribution of the past is similar enough to that of the future, therefore we will sample tomorrow's daily return from our historical distribution.

Let us start by defining our historical data. Suppose we have a series of stock prices *S* where each element S_i is a reading taken from the closing price of that day over some τ time period of days.

Now to compute daily price changes, we will do this in the form of percentage changes (daily returns) with the following equation.

$$u_i = \frac{S_i - S_{i-1}}{S_{i-1}}$$

Equation 3

The percentage change u_i is that of the closing price from day $i - 1$ to i.

To create our distribution, for each stock in our portfolio calculate the series of daily returns *u*. Now for each day *i* we can also realise our portfolio returns and sort our portfolio series from highest percentage loss to largest percentage gain.

We can now estimate VaR percentile point, for any time-horizon, by finding the α quantile of the distribution of portfolio returns *r* we have defined, where our confidence level is (1-α). To realise the gain in terms of actual money, multiply the VaR percentile point by the initial investment and square root of the time-horizon to calculate VaR over.

For our Expected Shortfall estimate, we simply take the mean from our α lower quantile set of values and perform the realisation calculation explained above.

If our data consists of 501 days, we are able to calculate 500 days of daily returns and our initial investment is $10,000. After creating our distribution, we take a confidence level let us say c = 95%, find the point of the 5th percentile from the distribution. Our VaR estimate occurs at this point, which is the 475th biggest loss.

2.5 Model-building Approach

We move onto a parametric approach model-building, sometimes called variance-covariance. I will use these names interchangeably in the report. This method of computing a VaR estimate is a popular alternative to historical simulation. As opposed to using historical stock returns to build our profit-loss distribution, which is a non-parametric approach, we will make certain assumptions.

2.5.1 Gaussian Distribution

Assumptions

A main assumption is the volatility of a stock, for our definition now; we will consider the volatility derived from standard deviation Chapter 2.5.1 but later on we will define other volatility estimates that might prove to be more accurate for coming to a VaR result. These volatility estimates can be used in the variance-covariance approach in the same way we utilise standard deviation's volatility.

More importantly for applying our volatility, we will assume our returns follow on from a mathematical distribution like the Gaussian bell curve. In essence, the returns of the portfolio are normally distributed. We are also able to modify our approach and model our returns based on other distributions, seen in the next subsection.

Volatility calculations will also be discussed later.

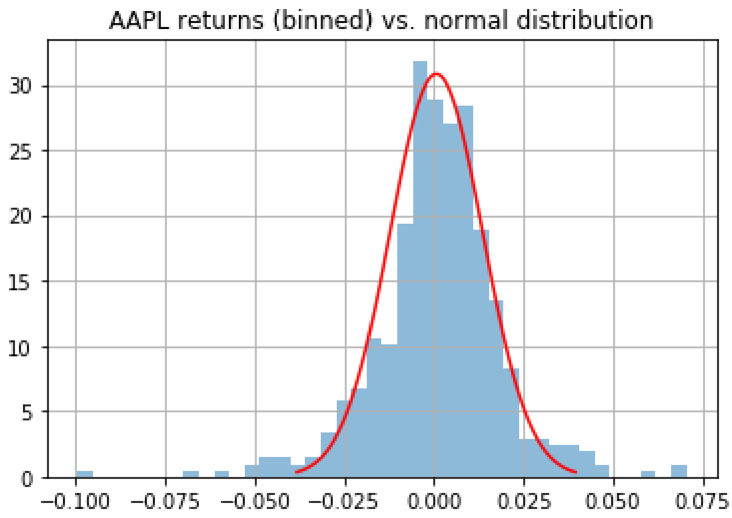


Figure 4 - profit-loss dist. vs normal dis. [4]

Using the Gaussian distribution to model profit-loss is quite applicable as we can see in figure above, but in spite of that during stressed market conditions (as one illustration where it does not hold) the normal distribution could fall short as the actual AAPL distribution might look quite different. Leading us to an inadequate VaR estimate.

The variance-covariance method will be looking at the historical price movements, however we compute mean returns and the volatility estimate of stocks over a specified lookback period as our parameters. We then use probability theory to calculate maximum loss for our specified confidence level, using the probability point function (inverse of the cumulative distribution function, similar to what we used in the historical approach).

For calculating VaR on a portfolio of multiple stocks we need to be cautious in the way we handle volatility. We need to measure how the different stocks vary with each other based on their volatilities. We do this by calculating their covariance Chapter 2.6.3

The mathematical definition of VaR under the normal distribution is simply the portfolio standard deviation multiplied by the percentile point function of the confidence interval and then subtracting the portfolio returns. However, to calculate our Expected Shortfall in the normal linear VaR model, we have to use the following equation which will automatically give us the expectation.

$$CVaR_{t,\alpha}(X) = \alpha^{-1}\varphi\left(\boldsymbol{\Phi^{-1}(\alpha)}\right)\boldsymbol{\sigma_t} - \boldsymbol{\mu_t}$$

Equation 4

Where $\varphi(Z)$ is the standard normal density function and $\boldsymbol{\Phi^{-1}(\alpha)}$ gives us the alpha quantile of the standard normal distribution.

Pawel Lachowicz has found this formula [1], where he integrated:

$$\alpha^{-1}\int_{-\infty}^{x_{h,a}} xf(x)dx$$

Equation 5

Where the upper limit $x_{h,a}$ is our VaR estimate and $f(x)$ is the continuous probability density function. α^{-1} is the inverse of our confidence level.

2.5.2 Student's t-distribution

Let us now change one of our major assumptions, the distribution that we model our returns on.

We will have a look at the Student's t-distribution. This distribution is said to be leptokurtic, having excess kurtosis than the Gaussian distribution. Having excess kurtosis means the distribution has a fatter tail compared to the Gaussian. The reason we would like to choose this to model our returns, can be seen previously in Figure 4, the majority of stock's daily return distributions display fatter tails. We can see that AAPL's returns peak much higher at both tails than the normal distribution, and these tails are our principal in finding VaR.

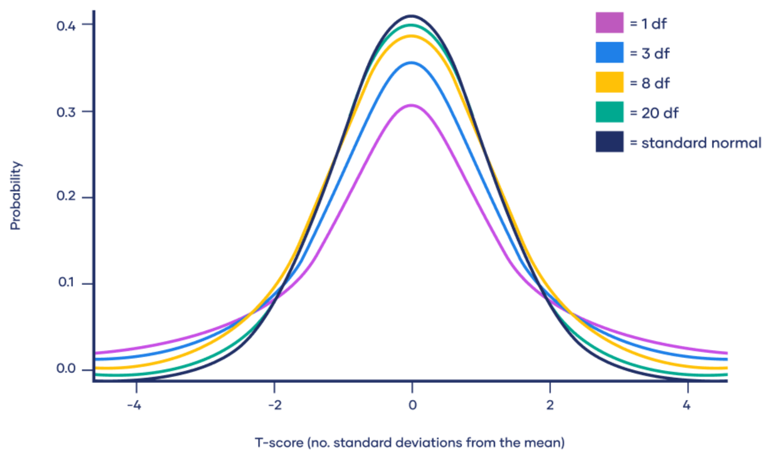


Figure 5 – student's t-dist. vs normal dist. [15]

Above we can see the fatter tails with Student's t-distribution compared to the normal distribution we used in the previous subsection. As well as this, the degrees of freedom determine the shape of the distribution and thickness of the tails. When $dof \to \infty$ the distribution tends to the standard normal.

The Student's t-distribution is formally given by its probability density function [16]:

$$f_v(x) = \sqrt{v\pi}\Gamma\left(\frac{v}{2}\right)^{-1}\Gamma\left(\frac{v+1}{2}\right)(1+v^{-1}x^2)^{-(v+1)/2}$$

Equation 32

Where v are the degrees of freedom, and Γ is the gamma function that is used to normalise the distribution.

We can now find the alpha quantile by integrating the probability density function, shown by Pawel Lachowicz[16], giving us:

$$\sqrt{v^{-1}(v-2)}t_v^{-1}(\alpha)$$

Equation 33

Where t_v^{-1} is the percentile point function.

When modelling returns, we deal with them as random variables $X = \mu + \sigma t$, suggesting we need to parametrise by mean and standard deviation. So, from this Pawel again shows us a formal definition of the 1–α, n-day VaR is:

$$t\ VaR_{v,\alpha,n} = \sqrt{v^{-1}(v-2)n}t_v^{-1}(1-\alpha)\sigma - n\mu$$

Equation 34

And William T. Shaw[17] shows us to find the CVaR which is:

$$t\ CVaR_{v,\alpha,n} = -\alpha^{-1}(1-v)^{-1}\left(v-2+x_{a,v}^2\right)\sigma - \mu$$

Equation 35

With both of these equations, we are now able to use them similarly to the Gaussian we have seen before.

2.6 Monte Carlo Simulation

The Monte Carlo method is generally described as a computational simulation that relies on multiple random sampling to obtain numerical results. These random variables are taken from a probability distribution. Random variables are important in transforming our returns in a Gaussian distribution.

Parts of our Monte Carlo approach follows similarly to our Historical method for computing (C)VaR, we take the bottom percentile in the same way. Nevertheless, in Monte Carlo Simulation we generate tomorrow's stock prices as opposed to sampling the historical returns.

Now to show how we will compute our future daily returns, based on historical data. This is the first step, to create a set of daily return values to then find the lower quantile point as the VaR.

We will assume that our daily returns are distributed by a Multivariate Normal Distribution:

$$R_t \sim MVN(\mu, \sum)$$

Equation 6

Where $\mu, \sum$ are the mean return from a series and the covariance matrix respectively and t is our time horizon.

Our model for daily returns R_t is going to follow similarly to Geometric Brownian Motion.[13]

$$R_t = \mu + LZ_t$$

Equation 7

Where:

$$Z_t \sim N(0, I)$$

Equation 8

Z_t are samples from the normal distribution with I representing an identity matrix.

And L represents the Lower Triangular Matrix from the Cholesky decomposition of the Covariance Matrix.

$$L \in LL' = \sum$$

Equation 9

This is used so that we can model an entire portfolio rather than one stock with covariances, shown in Chapter 2.6.

Now to perform the actual simulation we will need to calculate the parameters required above, define weights $w = [w_1 \dots w_n]$ and an initial portfolio value (how much we invested in total). We will then loop through for the number of simulations we want, performing the daily return calculation, equation 7, each time and then computing the dot/inner product between weights and daily returns. Storing this information, we will then have an array of portfolio simulations on one axis and time horizon on the other.

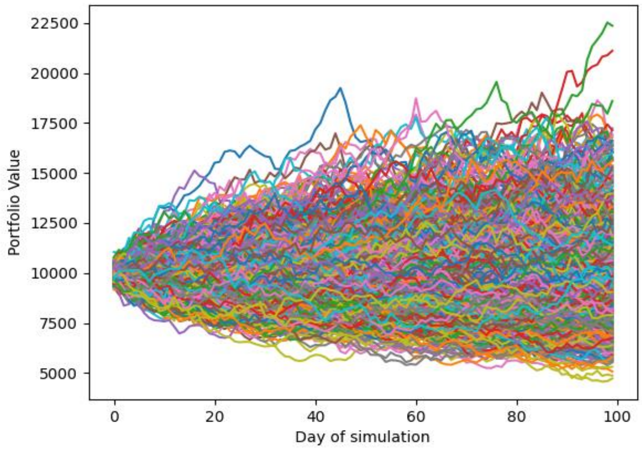


Figure 6- Monte Carlo Simulation from FYP program

Plotting this array on a graph will look like figure 6. Here we see a time horizon of 100 days starting with initial portfolio value of $10000.

Now that we have built up this distribution of potential daily returns, we can calculate the VaR and CVaR in the same way as the Historical Simulation, taking the bottom percentile of returns. A useful example is in the above figure 6 we can see our 5^{th} percentile VaR will be around $7500 indicating a VaR estimate of approximately $2500 for the 100 simulated days.

2.7 Variance and Volatility estimates

In this chapter we will explain different methods that give us estimates of variance for future and current time periods. We will also show how to calculate correlations and covariance between multiple assets so that we can model VaR for a portfolio. These calculations will primarily be used for parametric approaches to VaR such as model-building Chapter 2.4. We will start by looking at standard deviation which gives us an estimate to variance, but we will also look at other methods of estimating variance.

A notable example of these methods is Exponentially Weighted Moving Average (EWMA).

Volatility does not have an instantaneous existence; it is rather a measure that demands us to have a time series of other assumptions. We estimate the volatility on day i, and going forward to $i+1$, $i+2$ to the current day of $i+n$.

2.7.1 Standard deviation

For our standard deviation estimate we will have historical returns weighted equally, thus each day the variance will be just as important as one another when calculating our final variance estimate.

Leading on from Hull Chapter 17.1 [3] we define σ_n as the volatility on day n, which is calculated from day *n-1*. The square of volatility σ_n^2 on day n is the variance, that we are interested in for VaR.

As we have defined in historical simulation, we have u_i as the percentage change in the market price variable between the closing price of $i - 1$ to i.

$$u_i = \frac{S_i - S_{i-1}}{S_{i-1}}$$

Equation 10

Now to formally define variance from standard deviation for VaR:

$$\sigma_n^2 = \frac{1}{m}\sum_{i=1}^{m}(u_{n-1} - \bar{u})^2$$

Equation 11

Where each variance estimate has a weight of $\frac{1}{m}$, m is the number of observations. $\bar{u}$ is the mean of the series of returns and we can get u_{n-1} from equation above.

2.7.2 Equally Weighted Moving Average (EWMA)

As opposed to using an equal weighting scheme, by multiplying each day's variance by $\frac{1}{m}$ in equation 10, we will have a decay factor λ which we use to give more weight to more recent returns. The reasoning behind this approach is that we believe giving more weight to more recent returns will give us a closer estimate to the true variance. This solves the problem that Yesterday's, very recent, returns has no more influence than the returns from months ago.

Let us follow on from Hull Chapter 17.2. He introduces the formula for EWMA volatility as:

$$\sigma_n^2 = \lambda\sigma_{n-1}^2 + (1-\lambda)u_{n-1}^2$$

Equation 12

Where u_{n-1}^2 are the returns squared from day $n-1$.

Hull then shows us mathematically why the weights decrease exponentially, we substitute σ_{n-1}^2 with the formula above but for day $n-2$.

$$\sigma_n^2 = \lambda(\lambda\sigma_{n-2}^2 + (1-\lambda)u_{n-2}^2) + (1-\lambda)u_{n-1}^2$$

Equation 13

Or more simply by factorising,

$$\sigma_n^2 = (1-\lambda)(u_{n-1}^2 + u_{n-2}^2) + \lambda\sigma_{n-2}^2$$

Equation 14

If we then do this again for σ_{n-3}^2 we see

$$\sigma_n^2 = (1-\lambda)(u_{n-1}^2 + u_{n-2}^2 + u_{n-3}^2) + \lambda\sigma_{n-3}^2$$

Equation 15

And now to generalise for m days.

$$\sigma_n^2 = (1-\lambda)\sum_{i=1}^{m} \lambda^{i-1} u_{n-i}^2 + \lambda^m \sigma_{n-m}^2$$

Equation 16

For large m: Our smoothing factor λ is a decimal between 0 and 1, so it will begin to decrease to its limit of zero allowing us to ignore it. Leading us to the equation for estimating volatility with EWMA:

$$\sigma_n^2 = \sum_{i=1}^{m} (1-\lambda)\lambda^{i-1}\, u_{n-1}^2$$

Equation 17

Although we have seen something remarkably similar before, the standard deviation estimate equation 10. In the case of VaR it is common to see standard deviation not considering the mean, equation 18, as the variance rate is quite similar either way shown below.

$$\sigma_n^2 = \frac{1}{m}\sum_{i=1}^{m} (u_{n-1})^2$$

Equation 18

So apart from that fact, EWMA is just like doing a standard deviation calculation with the weighting scheme of $(1-\lambda)\lambda^{i-1}$ for each day i.

Now to talk about our parameter λ. In this report we will use a lambda value of 0.94 taken from RiskMetrics [5], a financial risk management company. Having a higher value of 0.94 (94%), this means our decay of the series of returns will be slower. Using a lower lambda would indicate a higher decay rate and more of our values would not be taken into account towards older returns.

2.7.3 Covariance

Covariance is a measure of how variables vary with each other. Let us say we have two random variables, if they move together, we get a positive covariance. Conversely, if they drift apart then the covariance will be negative.

This is a requirement for calculating VaR using a parametric method like model-building if we have a portfolio.

For a portfolio of multiple assets, we need to take a measure of how the variance of each asset moves with each other. The volatility is calculated using a matrix. This covariance matrix is used as a measure of the joint variability of the random variances.

To calculate covariance for n different stocks, start by calculating the average price of a stock. We can have a dataframe, of *n* series of stocks made up by their price values v:

$$S = \begin{bmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{x1} & \cdots & v_{xn} \end{bmatrix}$$

Equation 19

S is a dataframe made up of x stock prices where each column is the stock, so n stocks.

We then calculate the average price of each stock giving us:

$$M = [M_1\ M_2 \dots M_n\ ]$$

Equation 20

Now we need to establish an association that we will use to measure how each stock's change is related to the others. We can demean the prices, comparing how one stock's movement from its mean is dependent on the other movements from their means.

$$S - M = \begin{bmatrix} v_{11} - M_1 & \dots & v_{1n} - M_n \\ \dots & \dots & \dots \\ v_{1x} - M_n & \dots & v_{xn} - M_n \end{bmatrix}$$

Equation 21

Finally, to find the covariance matrix we multiply the transpose of the demeaned price series by itself and divide by the number of price data points n.

$$covariance\ matrix = \frac{(S - M)^T(S - M)}{n}$$

Equation 22

The resulting covariance matrix has the diagonals representing the variance of each stock while the rest of the values are symmetric representing covariance between stocks.

Portfolio Volatility

With the covariance matrix it allows us to find portfolio volatility which can be used in the model-building approach for multiple assets.

First, we need to define the weights of how many stocks we own in the portfolio, this vector will need to sum to 1 (e.g., 0.1 as 10%). $weights, w = \ [w_1 \dots w_n]$

$$portfolio\ volatility, \sigma_P^2 = w^T \sum w$$

Equation 23

Where $\sum$ is the covariance matrix.

With matrices the calculation looks like:

$$\sigma_P^2 = [w_1 \dots w_n] \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1n} \\ \vdots & \ddots & \vdots \\ \sigma_{n1} & \cdots & \sigma_{nn} \end{bmatrix} \begin{bmatrix} w_1 \\ \dots \\ w_n \end{bmatrix}$$

Equation 24

The portfolio volatility is the vector of the weights of the portfolio multiplied by the covariance matrix and then again multiplied by the transpose of the weights vector.

2.8 Incorporating Options

So far we have discussed how to compute a VaR estimate for stocks in a portfolio, but we can also incorporate more complex financial derivatives like European call and put options with the use of Historical or Monte Carlo simulation.

Focusing on the Monte Carlo simulation, we can modify it with the use of an option pricing method.

Firstly, we will incorporate options into our portfolio Π simply by:

$$\Pi \text{ value} = \Delta_{s_i} S_i + \Delta_{c_i} c_i$$

Where S_i is the stock price and Δ_{s_i} is number of the stocks we own. Similarly, c_i is the price of an option (from option pricing formula) and Δ_{c_i} is the number of options.

In our previous definition of Monte Carlo simulation, 2.5, we used the Cholesky decomposition to model multiple stocks moving together in a portfolio. For regular stocks we will stick with this method. However, we will additionally simulate each stock separately for its respective option. These stock prices will be used in our option pricing formulas.

Before stepping into the algorithm for simulation with options, let us define an option pricing method.

2.8.1 Black-Scholes Formulas

The Black-Scholes formula for pricing European call or put options:

$$c(S,t) = SN(d_1) - Xe^{-r(T-t)}N(d_2)$$

$$p(S,t) = Xe^{-r(T-t)}N(-d_2) - SN(-d_1)$$

Where S is the stock's price, X is the strike price, r is the interest rate, T-t is the time to maturity, and N() is the cumulative normal distribution function.

And:

$$d_1 = \frac{\ln\left(\frac{S}{X}\right) + (r + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}$$

$$d_2 = \frac{\ln\left(\frac{S}{X}\right) + (r - \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}$$

2.8.2 Monte Carlo Simulation with European Options

Now that we have already simulated our portfolio with stocks, let us call this $\Pi_S = \Delta_{s_i} S_i$. Bear in mind that Π_S will be changing as stock prices change throughout the simulation.

Algorithm: Monte Carlo VaR with Options

Let us simulate 10-day VaR.

1. Calculate option prices c_i, using pricing method, at present time.
2. Revalue portfolio: $initial\ \Pi_{total} = \Delta_{s_i} S_i + \Delta_{c_i} c_i$
3. For each stock an option is on:
   a. Simulate 10-days of prices, for M simulations.
4. For all simulations take 10th day simulated price and recalculate option prices c_i.
5. Revalue M portfolios: $simulated \Pi_{total} = \Delta_{s_i} S_i + \Delta_{c_i} c_i$
6. Compute VaR:
   a. Sort M portfolios.
   b. Chose alpha point from M portfolios.
   c. VaR is now difference between $initial \Pi_{total}$, and the alpha worst portfolio from $simulated\ \Pi_{total}$.

Options allow us to diversify our portfolio which can help to mitigate risk. This is dependent on the type, parameters, and quantity of options chosen for a specific portfolio. We will experiment with a put option to hedge our risk against a stock later.

2.9 Backtesting

Now that we have discussed multiple ways of computing VaR, we want to know how accurate and reliable our different models can be. Backtesting enables us to do this, by comparing our predictions to actual historical data.

To illustrate this, let's say we have a model that makes a 1-day VaR prediction using 252 days of data. If this was the Historical Simulation, we would use the 252 days of data to create our profit-loss distribution from which we take the α percentile to estimate VaR. Now to start our backtest, we will get 1000 days of market data. We will now go through each of these days using our model to make a prediction. Having the prediction for that day, we compare it against the actual 1-day fluctuation suffered in the market. If that day suffered a loss greater than our VaR estimate, we can count this as a violation.

If our VaR model is used with a confidence level of 95%, over 1000 days, we would expect to see 50 violations. Our definition of VaR is that with a 95% confidence level, we have a 5% chance of the actual losses exceeding the prediction over a given time period.

If we find that we have many more violations than 50, then our model has underestimated VaR. Similarly, if we have too few violations our VaR is overestimated.

However, we would also like to know the threshold for how many violations occur to still accept a certain model – the tests will evaluate the losses at a specified level of confidence.

We will make use of two backtesting methods that are under the category of coverage tests. Both having hypothesis tests where the null hypothesis H_0 is the expected number of violations is equal to the actual number of market violations.

Let us follow on from Holton's value-at-risk.net[14], where she provides a Recommended Standard Coverage Test and Kupiec's PF Coverage test.

2.9.1 Standard Coverage Test

First, we will start off by defining a univariate exceedance process $\boldsymbol{I}$ which will determine what constitutes a violation:

$$I^t = \begin{cases} 0, & if\ loss\ \Pi^{t-1} - \Pi^t\ less\ than\ or\ equal\ to\ VaR \\ 1, & if\ loss\ \Pi^{t-1} - \Pi^t\ greater\ than\ VaR \end{cases}$$

Equation 25

We assume that I^t are Independent and Identically Distributed random variables.

Considering a q quantile loss of the VaR measure we can also define the coverage q^* of the VaR measure – that is the frequency of exceedances where $i^t = 1$.

As stated earlier, coverage tests are hypothesis tests, with null hypothesis H_0 that q = q^*.

The number of violations over the backtest period is simply:

$$\mathrm{x} = \sum_{t=0}^{\alpha} I^t$$

Equation 26

x is the observed value in a particular scenario of X, where X takes the binomial distribution:

$$X \sim B(\alpha + 1, 1 - q)$$

Equation 27

Where α is the number of days of VaR estimates we use in our backtest.

We can have a significance level ε, that can control the precision of our estimation. To test H_0 at any ε, we must determine an interval [x_1, x_2] such that:

$$\Pr(X \notin [x1, x2]) \leq \varepsilon$$

Equation 28

This criterion can be met by multiple intervals, which makes us want to find an interval that is generally symmetric as follows:

$$\Pr(X < x_1) \approx \Pr(x_2 < X) \approx \frac{\varepsilon}{2}$$

Equation 29

With the above, we want to optimise n such that it maximises *Equation 28* with an interval as either:

$$[a + n, b] \ or \ [a, b - n]$$

Where a defined as the maximum integer such that $\Pr(X < a) \leq \varepsilon/2$ and b is the minimum integer such that $\Pr(b < X) \leq \varepsilon/2$ and n is a non-negative integer. Considering all intervals of the above form, set [x_1, x_2] = maximum($\Pr(X \notin [x1, x2])$).

Holton then states that the backtesting procedure just involves observing the performance of the VaR model over a historical period with $\alpha + 1$ periods and recording the number of exceedances. If X falls outside the interval *[x1, x2]* we reject the null hypothesis at the ε significance level. Thus, rejecting the VaR model.

2.9.2 Kupiec's PF Coverage Test

With a VaR approximation for $\alpha + 1$ periods, we have X exceedances occur.

Similarly, to the Standard Coverage Test, we use the same null hypothesis H_0 that q = q^*. But now instead of calculating the probabilities from $B(\alpha + 1, 1 - q)$ distribution of X under H_0, we will create a likelihood-ratio using it.

$$\Lambda = \frac{q^{\alpha+1-X}(1-q)^X}{\left(\frac{\alpha+1-X}{\alpha+1}\right)^{\alpha+1-X}\left(\frac{X}{\alpha+1}\right)^X}$$

Equation 30

It can be difficult to identify a non-rejection area with a desired confidence level because finding probabilities from Λ is challenging. A common method is just to consider -2log(L).

$$logL(n) = 2\log\left(\left(\frac{\alpha+1-X}{q(\alpha+1)}\right)^{\alpha+1-X}\left(\frac{X}{(1-q)(\alpha+1)}\right)^X\right)$$

Equation 31

Where X is a non-negative integer and q is our confidence level.

Like before, we use ε to determine the width of the non-rejection interval $[x_1, x_2]$ such that:

$$\Pr(X < x_1) \leq \frac{\varepsilon}{2}$$

And

$$\Pr(x_2 < X) \leq \frac{\varepsilon}{2}$$

Now, we can calculate the ε quantile of the χ^2 distribution and set it equal to $logL(n)$ giving us two solutions.

Kupiec specified to round the lower result down and the higher one up, giving us x_1, x_2. Using a continuous distribution to estimate a discrete one would make more sense to round the other way round, but we will stick to Kupiec's definition.

Note that, Equation 31, the likelihood ratio is asymptotically distributed as a χ^2 (chi-square) variable with 1 degree of freedom. If the statistic exceeds the value of the chi-square distribution, then the null hypothesis is rejected. This can lead us to use p-values instead of a non-rejection interval.

2.9.3 Stress Testing

Stress testing is a generic term where financial institutions take different approaches to suit their needs. It generally consists of running Value at Risk models during times of crisis in markets, or sometimes even creating hypothetical scenarios. For simplicity, we will just consider historical scenarios. For example, the 2007-2008 financial crisis or the chaos caused by the Covid-19 pandemic.

We will likely see that our Value at Risk estimates won't be accurate, such events don't occur often enough to be captured by our models which are empirically driven.

Chapter 3: **Software Engineering**

The implementation of the program this report introduces, is a numerically based program so it has been difficult to fit in the standard Software Engineering techniques that we might be so used to. However, it was still possible to use these techniques to our advantage.

3.1 Version Control System

Throughout the entirety of the program, it has been uploaded to a VCS (Version Control System) on a regular basis. The main reason for this is due to the fact that data storage devices are not built to last forever, being able to upload to a cloud service like GitLab has proved vital in keeping the integrity of the work produced. As it is on the cloud, it allows anyone concerned with the project to connect via the Internet to see the work that is being produced or even commit new work.

3.2 UnitTests

Due to the nature of this project, it has been difficult to use TDD as an advantage. However, where possible Python UnitTest has been used to help the flow of work. These tests primarily focus on the parts of the project that aren't concerned with the mathematical calculations but rather just as a check to make sure the algorithms don't break or that data has been collected properly. The following is an example of a two UnitTests to check that the data has been collected correctly and that the VaR value is smaller than the CVaR value respectively:

```
Def test_var_init():

  end_date = datet.datetime(2022,2,2)

  start_date = end_date - datet.timedelta(days=100)

  stock_list = ['AAPL', 'TSLA', 'GOOGL', 'RTX’]

  weights = np.array([0.20, 0.50, 0.20, 0.10])

  alpha = 5


  row_num = len(VaR(stock_list, start_date, end_date, weights, alpha).get_daily_return())

  assert row_num == 70
```

The above test runs the initialisation method and uses a getter to confirm that the right amount of data has been collected, 70 rows which I manually checked.

```
def test_historical_cvar():
```

```
  end_date = datet.datetime(2022,2,2)

  start_date = end_date - datet.timedelta(days=100)

  stock_list = ['AAPL', 'TSLA', 'GOOGL', 'RTX’]

  weights = np.array([0.20, 0.50, 0.20, 0.10])

  alpha = 5


  t = 20


  hvar = VaR(stock_list, start_date, end_date, weights, alpha).historical_var()*np.sqrt(t)

  hcvar = VaR(stock_list, start_date, end_date, weights, alpha).historical_cvar()*np.sqrt(t)


  assert (hcvar < hvar)
```

Now testing that the Expected Shortfall value is greater than the Value at Risk. This is always true so it is a reasonable way to check that the values returned are around the right range.

3.3 Documentation

Software is only complete with documentation; it helps outsiders understand the code and gives the programmer a better idea of what his code is doing too. For this project it has been especially vital as the documentation helps explain the difficult mathematical concepts. At the beginning of each function, there is a comment block that (if necessary) shows which mathematical formula is being used. The following is an example of this:

```
def monte_carlo_portfolio( …… ):

  """

  Simulating multiple runs of the portfolio using Monte Carlo simulation with equation similar to Geometric Brownian Motion.

  We will be following the formula to calculate future returns:

    Daily Returns t = MeanReturn + LZt


    where Zt are samples from normal distribution, L is a Cholesky decomposition of covariance matrix.
```

```
  :param mc_simulations: number of monte carlo simulations to be produced

  :param time: timeframe in days to simulate for.

  :return: portfolio_simulations: the results of our simulations, test_portfolio_val: Initial investment.

  """
```

3.4 Waterfall Approach

This approach of Software Engineering has been very easy to implement due to the nature of this project. This is because there were multiple different approaches, to compute VaR, so it only felt natural to complete the project in multiple sequential phases of the same type. Most of these phases even depended on the previous for them to work.

3.5 Yahoo Finance API

The first step we took to set up our (C)VaR program was to collect data in some meaningful way. One option was to gather data from the Internet and APIs do just the right job. They provide reliable, live data. This was a simple choice as Python has many libraries that are built for this purpose. I found a Python library, pandas-datareader[6], that pulls data from *Yahoo Finance API* and has multiple inbuilt functions that allow for cleaning irrelevant items. A big reason for choosing this way of getting data is that the library works under Python pandas' library, it is easy and simple to use, and I have worked with it before.

3.5.1 Historical Stock Data

My implementation consists of a Python file with a VaR Class, in this class we have an __init__() method that will initialise a VaR object anytime one is created. The purpose of this method is to firstly call *Yahoo Finance API* and then clean through the data selecting relevant information we will need for our VaR calculations.

```
def __init__(self, s, start, end, weights, alpha):

  yahoo_data = pandasdr.get_data_yahoo(s, start=start, end=end)['close']

  self.daily_returns = yahoo_data.pct_change()

  self.mean_returns = self.daily_returns.mean()

  self.covariance_matrix = self.daily_returns.cov()

  self.weights = weights

  self.alpha = alpha

  self.daily_returns.dropna()
```

```
self.daily_returns['portfolio']=self.daily_returns.dot(self.weight
s)
```

From the above code we can see that we call the API, requesting to get a pandas dataframe made up of multiple pandas' Series of stocks from the array `s`. We choose our start and end dates and also specify that we want `['Close']` closing prices which are the prices of the stock at the end of day i. The `self` keyword is used to assign attributes to each object that we will need for VaR calculations.

Our daily returns are calculated from the .pct_change() function, from pandas-datareader, like we have seen in equation 3. The mean of our returns is simply using Python's .mean() on our dataframe of daily_returns. For now, we use Python's .cov() to compute the covariance matrix of our daily returns however, I plan to implement the calculation myself later.

Clean the data by dropping not applicable values with dropna().

Finally, we create a new series of values in our `daily_returns` called `'portfolio'` which will consider the weights we have assigned with them by simply taking the dot product across each series.

With all of these values we are able to perform simple VaR calculations using a historical simulation approach. In spite of that, to perform other estimates (model-building as an example) we need to get two additional attributes that fall out of the scope of an initialisation method. This will spare some calculating time.

Chapter 4: **Technical Decision Making**

4.1 API Failure

The Python Library used for getting market data, pandas-datareader, stopped working in Christmas of 2022. The reason for this is due to Yahoo! encoding some data on their webpage under AES.

This was a concerning issue as my program was not able to get data and compute VaR. I began investigating the issue and noticed from similar libraries that the issue was the AES encoding in a new Yahoo! update. After a week of no success, I found a fix that used a cryptographic library to decode the data. I implemented this into pandas-datareader, and the library began to work again: https://github.com/pydata/pandas-datareader/pull/953. The PR didn't get accepted as the maintainer, Dr Kevin Sheppard, of the library has seemed to be inactive. Nevertheless, I used the fixed version locally.

Additionally, this prompted me to become wary of any similar incidents occurring again. I chose to download a few stock's pandas DataFrames in CSV format and storing them in a separate directory with an associated Python file that is able to download the data. This would enable me to carry on using my program without the library.

Inevitably, a couple months later pandas-datareader broke again. I used the CSV data for the time being, until I moved onto another Python Library called yfinance. This was due to the fact that it downloaded the data in the same format, so my original code didn't need much adaption. And importantly, I was able to use live market data again with ease.

The CSV files will always be kept in case, so that we can always work out a VaR for any period of historical data that is downloaded.

```
import yfinance as yf

from pandas_datareader import data as pandasdr



yf.override()

pandasdr.get_data_yahoo(symbols,end=end,start=start)['Close'].to_c
sv(i+'.csv')
```

The above snippet shows the use of the new yfinance library, as well as the function to download the data in CSV format.

4.2 Simulating Monte Carlo with Options

For the Monte Carlo Simulation to compute VaR on a simple stock portfolio we used Equation 7, which included Cholesky decomposition that allowed the simulation of multiple stocks together into one daily return value. Thus, only 1 simulation was needed for an entire portfolio, saving the need to simulate X amount of times for X stocks.

However, when it came to incorporating options into the portfolio, this became an issue. It was necessary to simulate prices of each underlying asset separately so that these stock prices could be respectively used in option pricing formulas.

```
for i in range(0, mc_simulations):

      Z = np.random.normal(size=(time, length_returns_columns - 1))

      daily_returns = mean_matrix.T + np.inner(cholesky_lower_l, Z)

      weights_returns_dotproduct=np.inner(self.weights,

      daily_returns.T)

      portfolio_simulations[:,i]=test_portfolio_val*np.cumprod(weig
      hts_returns_dotproduct + 1)
```

The above snippet is from the monte_carlo_portfolio function. This part simulates the daily returns of multiple stocks together with the Cholesky decomposition described earlier. But we also want to see the individual stock’s returns/prices.

We chose a crude approach to fix this to save development time as it was easy to implement without having to refactor much of the previous code.

Continuing from inside the same for loop as above:

```
      if option_on:

            Z_option = Z[:, index]

            stock_prices=np.zeros(time)
            stock_prices[0]=last_stock_price



            for j in range(1, time):



                  stock_prices[j]    =    stock_prices[j    -    1]    *
np.exp(mean_index - (vol ** 2) / 2 + mean_index*Z_option[j])



            stock_simulations[i, :] = stock_prices
```

We simply take the row of Z values from the previous snippet of code, which are our samples from the Normal Distribution. If our option is on stock XYZ, then we just take the row of Z values with the index of that particular stock from the array Z. Then we simulate that stock's price for M simulations alongside the previous combined simulation.

This method requires us to perform an extra simulation of the stock that an option is on, however the VaR estimate is still produced in a reasonable amount of time. In addition to this, a recent realisation was that the first snippet of code is vectorised. We have multiple stocks combined into one simulation as opposed to X amount of stocks requiring X amount of simulations. This is faster than simulating X times and combining after. So even though the option code appears to be crudely attached on, it is most likely more efficient.

Chapter 5: **Experimentation And Results**

Models in use:

Historical: This is the method as described in Section 2.4, it simply assumes that the past return distribution is that of the future.

Parametric Normal: The model-building approach as described in Section 2.5

Return distribution: Gaussian.

Volatility: Equal Weighting.

Parametric t-dist.: The model-building approach as described in Section 2.5

Return distribution: Student's t-distribution.

Volatility: Equal Weighting.

Parametric EWMA: The model-building approach as described in Section 2.5

Return distribution: Gaussian.

Volatility: Equal Weighting Moving Average (EWMA).

Monte Carlo: The Monte Carlo simulation as described in Section 2.6

Z sampled from Gaussian

Volatility: Equal Weighting

5.1 Qualitative Assessments

We will begin by performing some qualitative assessments of the VaR estimates, by subjectively judging our values. We will later go into more rigorous methods with backtesting.

5.1.1 Initial Observations

Using market data from 15/03/21 to 15/11/21.

Let us start on the following stock(s): [HD] (Home Depot).

Weights = HD:100

VaR confidence level: 95%.

Initial portfolio investment is $10,000.

Calculating for 1-day time horizon.

Method	**(Conditional) Value at Risk - ($)**
Historical VaR	982.23
Historical CVaR	1285.85
Parametric Normal VaR	1083.17
Parametric Normal CVaR	1372.23
Parametric Student t VaR	1302.57
Parametric Student t CVaR	1680.99
Parametric EWMA VaR	1190.17
Monte Carlo VaR	1023.45
Monte Carlo CVaR	1343.98

Our first observation is that our VaR and CVaR values are respectively all quite close together, indicating that we are most likely on the right path in our estimation. Every CVaR estimate is larger than it’s VaR value which follows the definition of Conditional Value at Risk. These estimates must be larger as they are the expectation of the bottom quantile, while VaR is the boundary closest to the mean of the distribution.

For the Student’s t-distribution, we see a larger VaR and CVaR compared to the others. This is due to the distribution having excess kurtosis (fatter tails). So, it is promising to see larger estimates, especially compared to the Parametric Normal and Monte Carlo as they both employ the standard normal distribution.

This leads us onto another observation, that the Parametric Normal and Monte Carlo methods provide almost the exact same estimates. This is expected, even though they use two different techniques to create a profit loss distribution, both utilise the Gaussian distribution for this.

5.1.2 More risk over longer holding period.

Now we will estimate with the same parameters for a 10-day time horizon as opposed to 1-day.

Method	**(Conditional) Value at Risk - ($)**
Historical VaR	2840.44
Historical CVaR	4066.21
Parametric Normal VaR	3051.46
Parametric Normal CVaR	3965.55
Parametric Student t VaR	3500.45
Parametric Student t CVaR	3929.12
Parametric EWMA VaR	3763.64
Monte Carlo VaR	3098.12
Monte Carlo CVaR	4001.85

Comparing these results to 1-day time horizon, they are a lot larger. This is due to the fact that over a larger time horizon, we will expect to have more days where we have losses. There is more risk in holding the stock for a longer period of time.

5.1.3 Stock price movements with Monte Carlo

We will visualise how our Monte Carlo simulation changes based on different market conditions. Our VaR time-frame is going to be 100-day which is generally too large to be used as an accurate estimate, however we just want to nicely picture some observations.

Portfolio : [AAPL, TSLA, GOOGL]

Weights: [1.00]

Time frame: 100-day

Confidence level: 95%

Initial Investment: $10,000

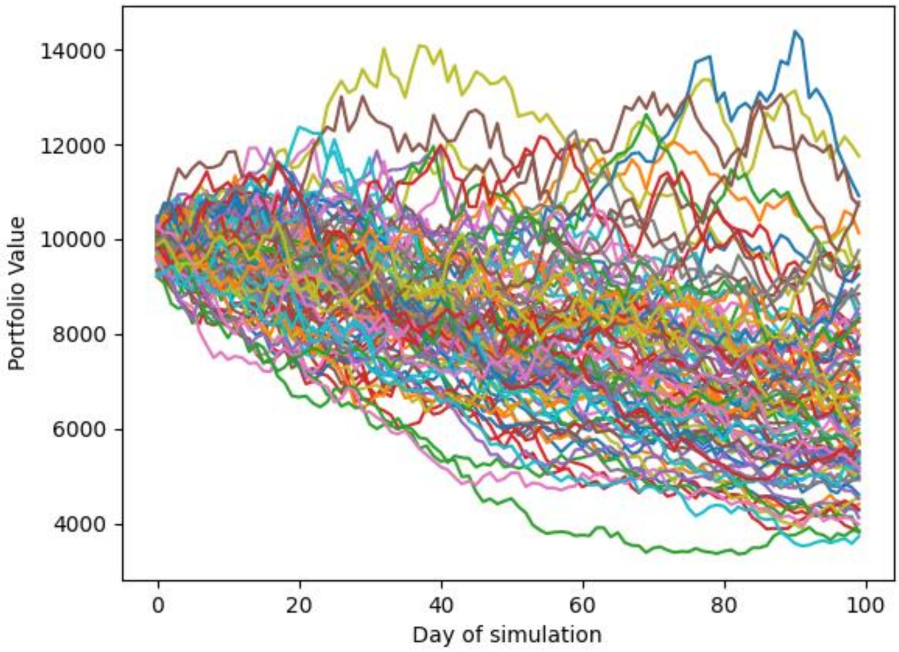


Figure 7 – Monte Carlo simulation of future prices

Method	**(Conditional) Value at Risk - ($)**
Monte Carlo VaR	5934.81
Monte Carlo CVaR	6561.33

Running our simulation, we can see that these stocks have a large risk and that the average stock price after 100 days is below the initial. This is because the Monte Carlo simulation will base its parameters from historical returns and these three stocks have been decreasing over the last few months as we can see below, on the right side of the stock graphs.

Figure 8 - Tesla Stock Price, last 6 months, Google Finance

Figure 9- Google/Alphabet Stock Price, last 6 months, Google Finance

Figure 10 - Apple Stock Price, last 6 months, Google Finance

If we instead calculate our historical volatility parameters over the time frame 15 June to 17 august, the left side of the above stock graphs. We will also modify our portfolio weights, so that we own more Apple stocks as it experienced more positive returns.

The more positive market period should forecast less risk.

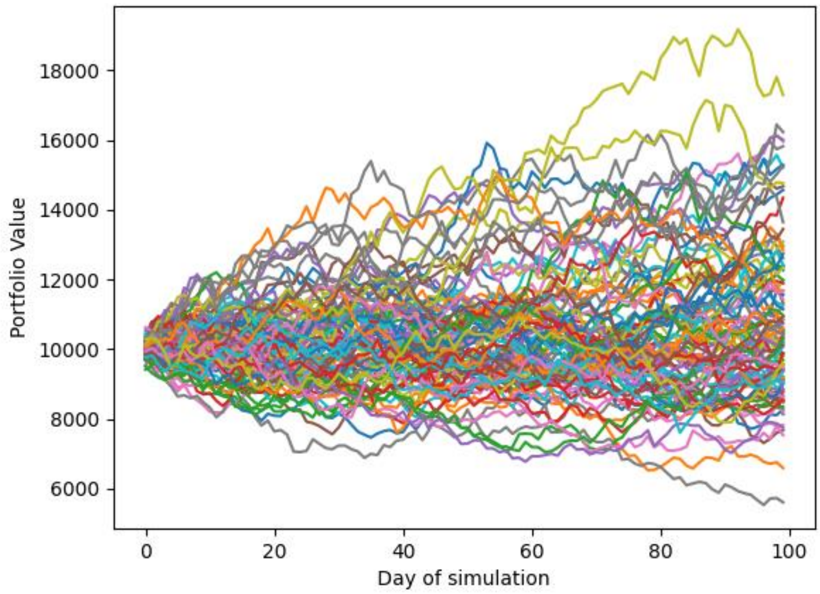


Figure 11 - Portfolio simulation based on above parameters.

Method	**(Conditional) Value at Risk - ($)**
Monte Carlo VaR	2570.07
Monte Carlo CVaR	2773.45

Now looking at the (C)VaR values we see over the same number of simulated days and calculating historical parameters in the same way but just over a different time period, a much greater decrease in risk. From $5934 to $2570. Additionally, our portfolio simulation looks much more positive.

5.2 Diversification

We will be looking at ways to mitigate risk in a portfolio. One method by diversifying with more stocks, and the second method by adding European put options on stock X to the portfolio of a stock X.

5.2.1 Stock only portfolio

Let us start off with testing the effect of increasing the number of stocks in a portfolio on our (C)VaR estimate. Our portfolio will consist of the stocks split up by weightings with an initial investment of $100,000.

Experiment 1

We will begin by estimating (C)VaR on a single stock.

Portfolio : [AAPL]

Weights: [1.00]

Time frame: 1-day

Confidence level: 95%

Method	**(Conditional) Value at Risk - ($)**
Historical VaR	2444.77
Historical CVaR	3222.52
Parametric Normal VaR	2421.07
Parametric Normal CVaR	3072.74
Parametric Student t VaR	2330.22
Parametric Student t CVaR	3218.48
Parametric EWMA VaR	3088.64
Monte Carlo VaR	2401.81
Monte Carlo CVaR	3084.73

Experiment 2

Now moving onto two stocks in the portfolio.

Portfolio : [AAPL, JPM]

Weights: [0.50, 0.50]

Time frame: 1-day

Confidence level: 95%

Method	**(Conditional) Value at Risk - ($)**
Historical VaR	2215.43
Historical CVaR	2918.88
Parametric Normal VaR	2029.54
Parametric Normal CVaR	2557.06
Parametric Student t VaR	1955.99
Parametric Student t CVaR	2675.04
Parametric EWMA VaR	2308.11
Monte Carlo VaR	2008.92
Monte Carlo CVaR	2546.64

Experiment 3

And finally with three stocks.

Portfolio : [AAPL, JPM, BOE]

Weights: [0.33, 0.33, 0.34]

Time frame: 1-day

Confidence level: 95%

Method	**(Conditional) Value at Risk - ($)**
Historical VaR	1855.53
Historical CVaR	2636.1
Parametric Normal VaR	1753.46
Parametric Normal CVaR	2195.93
Parametric Student t VaR	1691.78
Parametric Student t CVaR	2294.88
Parametric EWMA VaR	2066.95
Monte Carlo VaR	1765.43
Monte Carlo CVaR	2196.95

Visualising results:

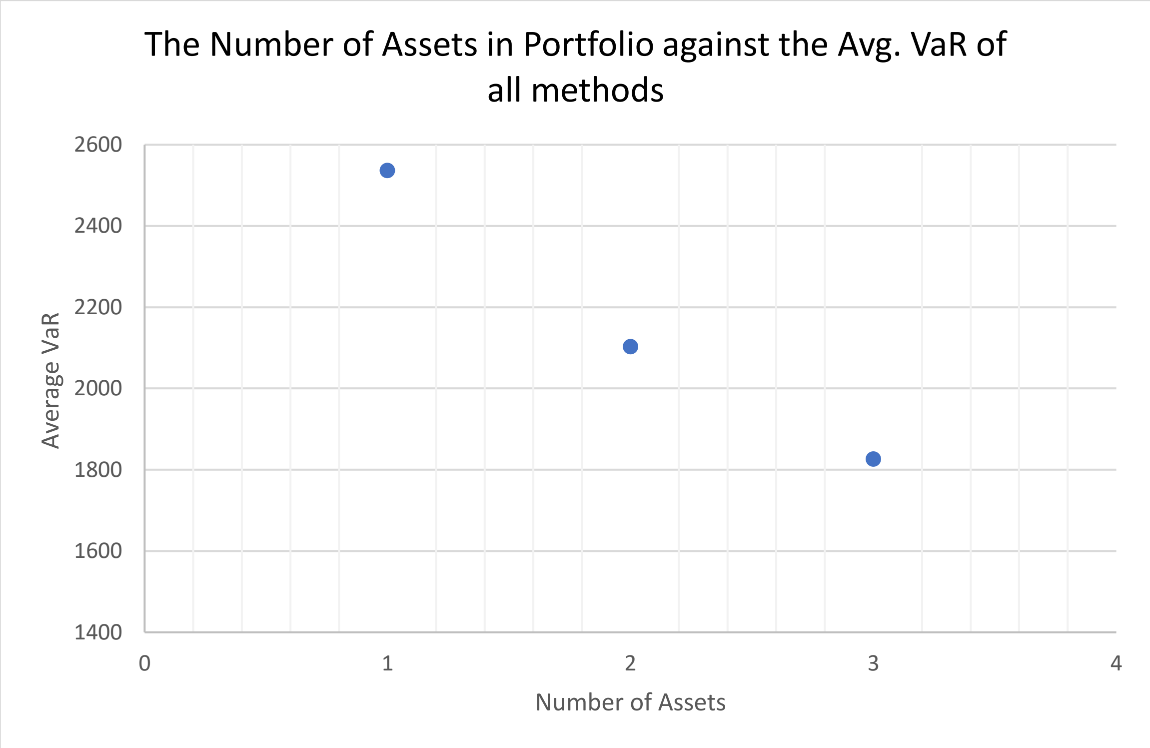


Figure 12 – Diversification visualised.

By taking the average VaR results of all methods from the above tables, we can see that as the number of assets increases, our VaR goes down. Due to our portfolio experiencing a decreased exposure to risk.

Our choice of assets (Apple, JP Morgan, Boeing) are in different sectors which further helps with diversification because their fluctuations are likely not to be linked together. The losses in one stock could be helped with an offset by gains in one of the other stocks.

5.2.2 Stock with European put options.

We will now use our modified Monte Carlo simulation to test the effects of adding a European Put option to a stock. Adding this type of option to a stock, that you own in your portfolio, is one technique for diversifying and mitigating risk.

Using a 10-day time frame to calculate VaR over will help us to see a change in the option price, and thus the portfolio, as it gets closer to maturity. We will use Monte Carlo to simulate stock prices for 10 days into the future.

Experiment 1

Portfolio : [TSLA] + No Options

Weights: [1.0]

Time frame: 10-day

Confidence level: 95%

Initial Investment = $100,000

Method	**Value at Risk - ($)**
Monte Carlo VaR	9861.58

Experiment 2

Adding 1000 European Put options:

With initial price $c_i = 8.02$ and,

Average final price $c_i = 6.89$ (from M simulations)

Portfolio : [TSLA] + 1000x European Put options

Time frame: 10-day

Confidence level: 95%

Initial Investment = $100,000 (approx. 600 TSLA shares)

Method	**Value at Risk - ($)**
Monte Carlo VaR (Black-Scholes pricing)	9376.27

Due to the option having a lower price on the last day of the simulation, the VaR has gone down accordingly as our portfolio with options is: $\Pi_{total} = \Delta_{s_i} S_i + \Delta_{c_i} c_i$

This shows that owning a stock with the stock's relative European put option is a hedging strategy to mitigate risk. This mainly depends on the strike price and other variables.

5.3 Backtesting

For our backtests, we will be focusing on the Kupiec POF test as it yields almost the same results as a Standard Coverage test. Let us first define the terms confidence level for VaR and backtesting as this can get slightly confusing, they are two separate definitions.

In VaR we have seen that the confidence level represents the degree of certainty that a given loss will not be exceeded within a specified time frame.

In the Kupiec tests, the confidence level is used as part of a statistical hypothesis test to determine the level of evidence required to reject the null hypothesis. Where the null hypothesis claims that the VaR model is accurate as described in Section 3.8. This confidence level is chosen before conducting the test.

In Section 3.8, we defined the Kupiec tests with non-rejection intervals. If our model produced a number of violations that are inside of this interval, the null hypothesis would not get rejected. The model would be deemed reliable. However, in this section we will work with p-values as they are easy to interpret and allow for pleasant charting.

Let us start off by defining our p-values. From Equation 31, we have seen the Kupiec likelihood ratio is asymptotically distributed as a χ^2(chi-square) variable with 1 degree of freedom. To find a p-value from that variable we just use the survival function, defined: (1 – cumulative distribution function). We take the χ^2 cdf with 1 degree of freedom, Figure 13: the line where k=1:

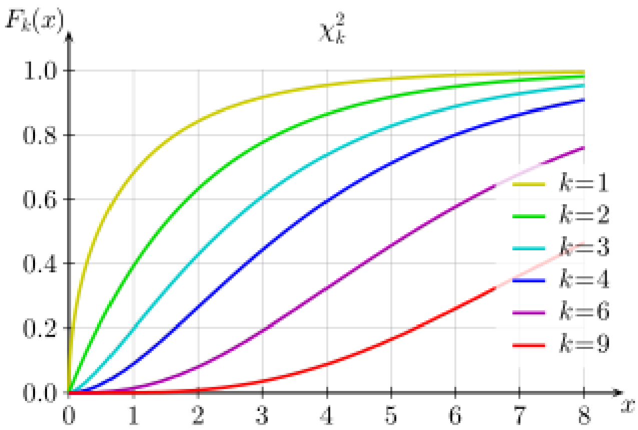


Figure 13 - χ^2 cumulative distribution function

Where k is the degrees of freedom.

Now let us give a toy example:

Backtest period: 1000 days

VaR confidence level: 5%

Kupiec confidence level: 5%

Method	Violations Count	Likelihood Ratio	p-value
Model 1	35	5.278	0.022
Model 2	40	2.253	0.133
Model 3	45	0.544	0.461
Model 4	47	0.193	0.660
Model 5	50	0.000	1.000

With 1000 days of backtesting at 5% VaR confidence level, we would expect to see $1000 * 0{,}05 = 50$ violations. Model 5 has exactly this many, resulting in a likelihood ratio of 0 and a perfect p-value of 100%. As the models violation counts get further away from 50, the likelihood ratio increases and subsequently the p-values get smaller. With a Kupiec confidence level of 5%, we reject any Model where the p-value is below 5%, in this case Model 1 has 2.2% which would be deemed inaccurate. 35 violations are quite far away from the expected 50. Model 2 has 40 violations with 13% p-value; this just about fits in to be deemed as a reliable model.

Now we will move onto performing backtests with our programmed models, all tests will be performed for a 1-day VaR time horizon.

Note: We will only be backtesting our models using the Value at Risk measure and not Expected Shortfall.

5.3.1 Student t-dist. Fat Tails

Idea: The Student t-distribution, with 3 degrees of freedom, should be better at predicting VaR at lower confidence levels due to it having fatter tails in the distribution, which is common across many stocks. The methods following the Gaussian approaches should perform weaker at the lower levels where they do not incorporate fatter tails.

Portfolio: [TSLA]

VaR conf. level = 1%:

Backtest period: 826 days

Kupiec confidence level: 5%

Method	Violations Count	Likelihood Ratio	p-value
Historical	12	1.501	0.221
Parametric Normal	23	17.894	0.0000023
Parametric t-dist.	12	1.500	0.221
Parametric EWMA	18	8.678	0.003
Monte Carlo	23	17.894	0.0000023

Visualising:

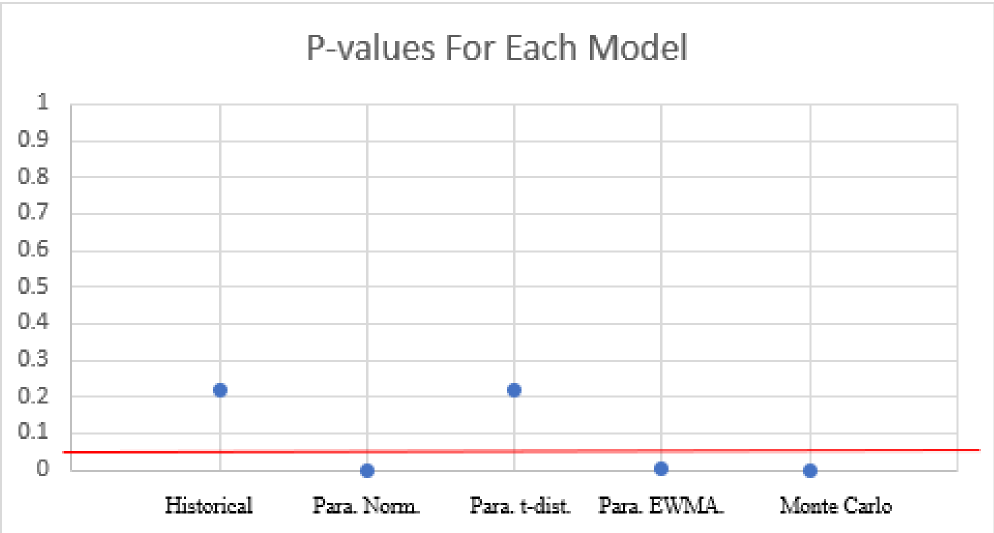


Figure 14 – P-values for 1%

The red line at p-value = 0.05 or 5%, shows anything below should be rejected as inaccurate.

Historical and Parametric t-distribution models are the only acceptable VaR estimates in this instance when VaR conf. level = 1%, the rest reject the null hypothesis, they are unreliable.

VaR conf. level = 5%:

Backtest period: 826 days

Kupiec confidence level: 5%

Method	**Violations Count**	**Likelihood Ratio**	**p-value**
Historical	44	0.182	0.669
Parametric Normal	49	2.238	0.232
Parametric t-dist.	65	12.281	0.00045
Parametric EWMA	44	0.128	0.669
Monte Carlo	49	1.429	0.232

Visualising:

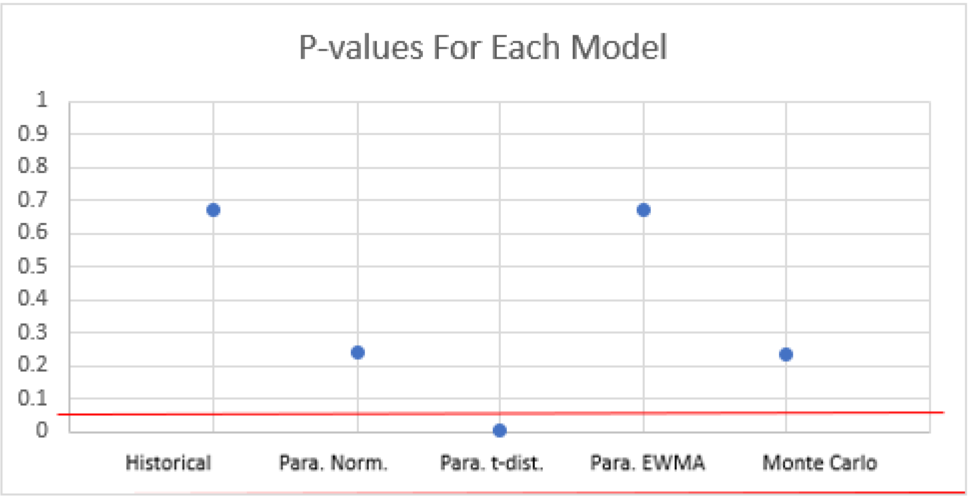


Figure 15 – P-values for 5%

All models are accepted as reliable VaR estimates, apart from Parametric Student's t-dist.

Now we can see that many of our models perform much better at a 5% significance level, however this is where the t-distribution starts to fail. It produces smaller VaR estimates at higher confidence levels leading to many more violations being counted, this is just the nature of the distribution.

Conclusion

For smaller VaR conf. levels, like 1%, the t-distribution is accepted. It is better at estimating for smaller conf. levels where the actual stock's profit-loss distribution has fatter tails, however it doesn't perform as well for VaR conf. level of 5% where we move further from the tail. It is also interesting to note that at 1% the Historical method worked just as well, as it doesn't base its return distribution on the Gaussian assumption. Additionally, at 5% it attained the highest p-value along with EWMA.

The results are compatible with the conjecture that the models which do not take account of excess kurtosis underestimate the Value at Risk at the 1% conf. level and don't perform well. The Historical approach seems to be more versatile as it worked well at both levels.

5.3.2 A bigger portfolio

Portfolio: [JPM, TSLA, ADBE, AAPL]

Weights: [0.30, 0.20, 0.20, 0.30]

VaR conf. level = 1%:

Backtest period: 826 days

Kupiec confidence level: 5%

Method	Violations Count	Likelihood Ratio	p-value
Historical	14	3.348	0.067
Parametric Normal	27	26.957	0.00000002
Parametric t-dist.	12	1.500	0.221
Parametric EWMA	15	4.491	0.0345
Monte Carlo	26	24.579	0.00000007

Visualising:

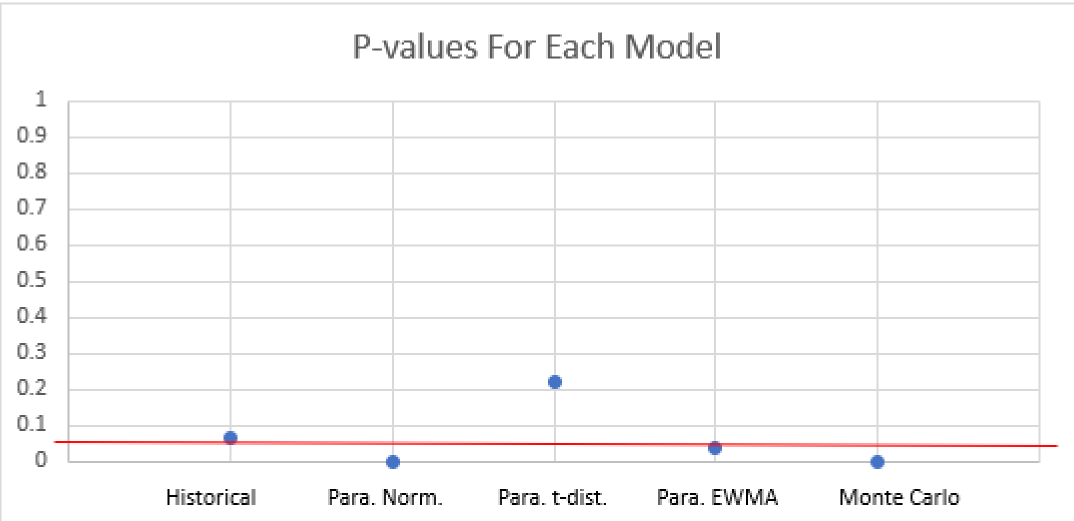


Figure 16

The Historical model is just about accepted as reliable, and the Student's t-dist. is accepted with a p-value of 22% at the 1% VaR conf. level. The rest are rejected as inaccurate.

VaR conf. level = 5%:

Backtest period: 826 days

Kupiec confidence level: 5%

Method	**Violations Count**	**Likelihood Ratio**	**p-value**
Historical	49	1.430	0.232
Parametric Normal	45	0.339	0.560
Parametric t-dist.	70	17.531	0.0000028
Parametric EWMA	39	0.137	0.711
Monte Carlo	45	0.339	0.560

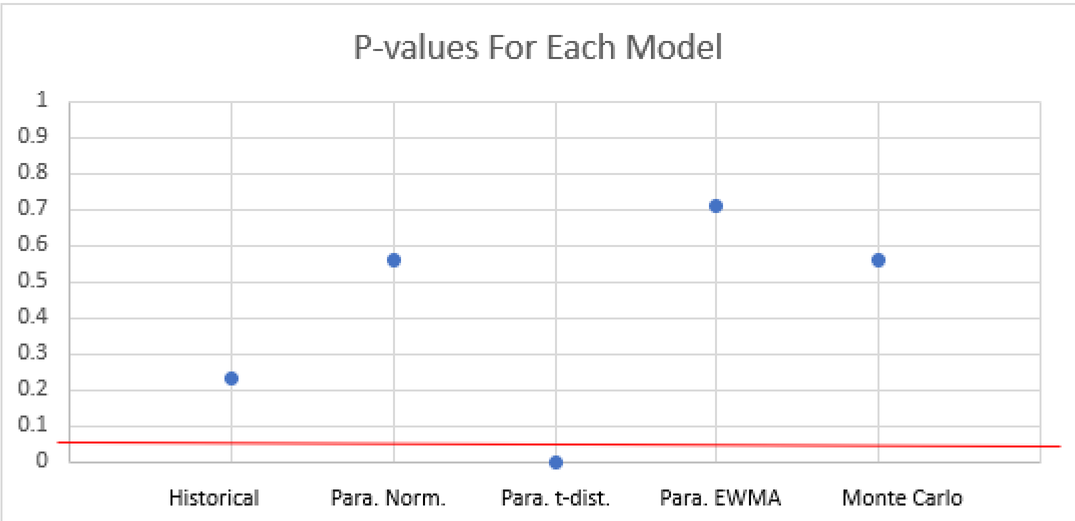


Figure 17

At 5% VaR conf. level, we again see much worse performance for the t-dist. while the other models perform better, all are accepted as reliable estimates apart from Parametric Student's t-dist.

VaR conf. level = 10%:

Backtest period: 826 days

Kupiec confidence level: 5%

Method	Violations Count	Likelihood Ratio	p-value
Historical	99	3.424	0.0642
Parametric Normal	82	0.005	0.944
Parametric t-dist.	103	5.233	0.022
Parametric EWMA	75	0.799	0.372
Monte Carlo	82	0.005	0.944

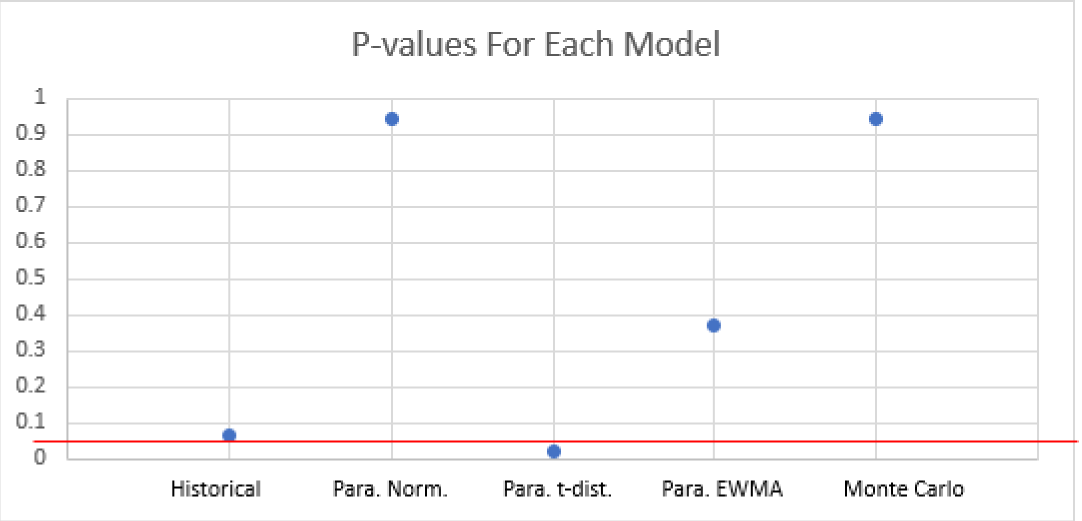


Figure 18.

Here we see that the models assuming Gaussian distribution work very well with p-values near perfect of 94%. The Historical, Para. Norm., Para. EWMA, and Monte Carlo models are accepted as reliable, however the Historical only just gets by. As expected, the t-dist. again performs badly as we are not focusing on the tails of the distribution with a VaR conf. level of 10%.

5.3.3 Another portfolio

Let us try a different portfolio.

Portfolio: [JPM, HD, AAPL]

Weights: [0.33, 0.33, 0.34]

VaR conf level = 1%

Backtest period: 826 days

Kupiec confidence level: 5%

Method	Violations Count	Likelihood Ratio	p-value
Historical	12	1.501	0.221
Parametric Normal	18	8.679	0.003
Parametric t-dist.	12	1.501	0.221
Parametric EWMA	20	12.061	0.0005
Monte Carlo	18	8.679	0.003

Visualising:

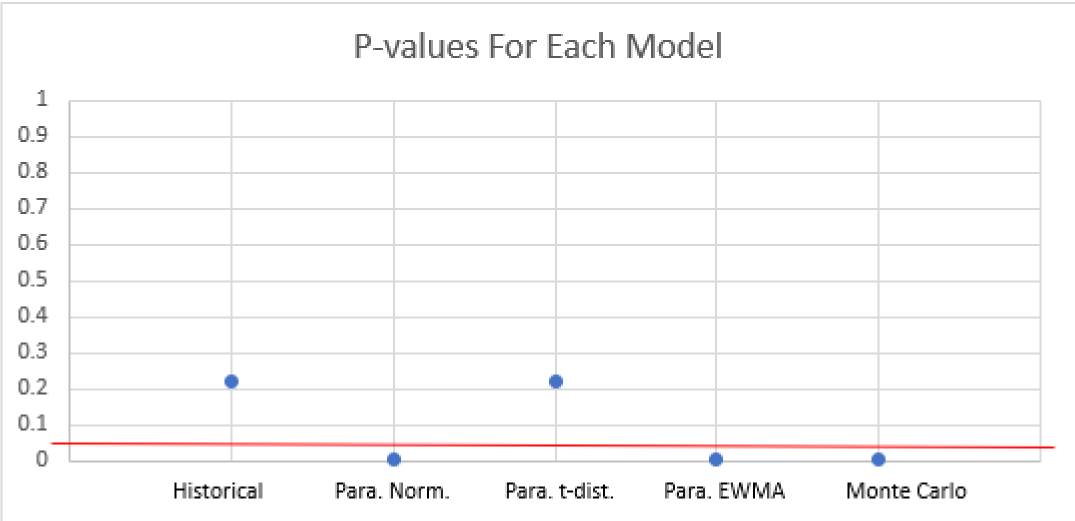


Figure 19

At the 1% VaR conf. level, Historical and Para. t-dist. work well again and are accepted as reliable models, while the others are rejected.

VaR Conf. level = 5%

Backtest period: 826 days

Kupiec confidence level: 5%

Method	Violations Count	Likelihood Ratio	p-value
Historical	39	0.137	0.711
Parametric Normal	45	0.339	0.560
Parametric t-dist.	54	3.764	0.052
Parametric EWMA	44	0.182	0.670
Monte Carlo	45	0.339	0.560

Visualising:

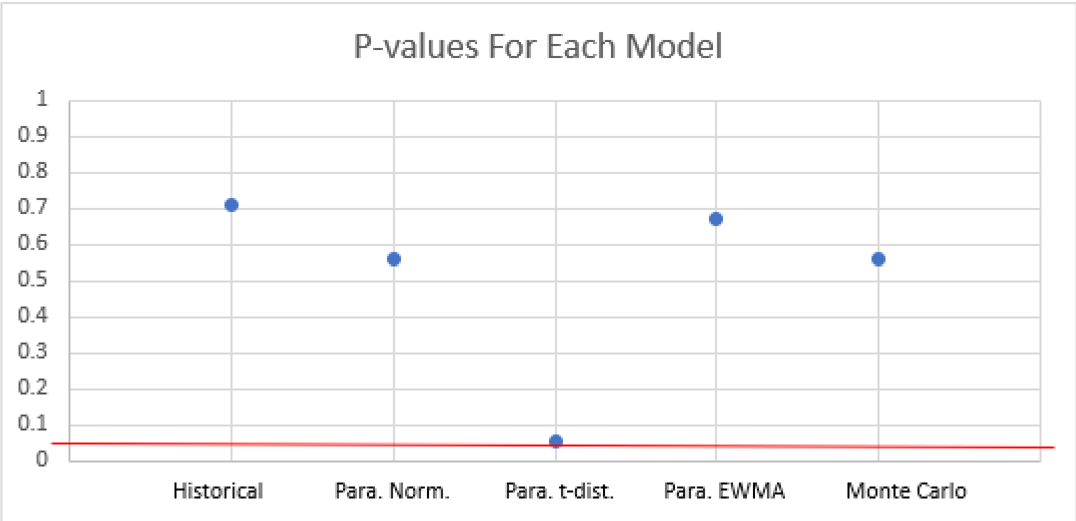


Figure 20

With a VaR conf. level of 5%, all our models are accepted. Noting that Para. t-dist. only just crosses into acceptance. However, the other models perform very well nearing the expected amount of violations and thus high p-values.

VaR conf. level = 10%

Backtest period: 826 days

Kupiec confidence level: 5%

Method	Violations Count	Likelihood Ratio	p-value
Historical	81	0.852	0.852
Parametric Normal	75	0.799	0.371
Parametric t-dist.	93	1.403	0.236
Parametric EWMA	72	1.573	0.209
Monte Carlo	75	0.799	0.371

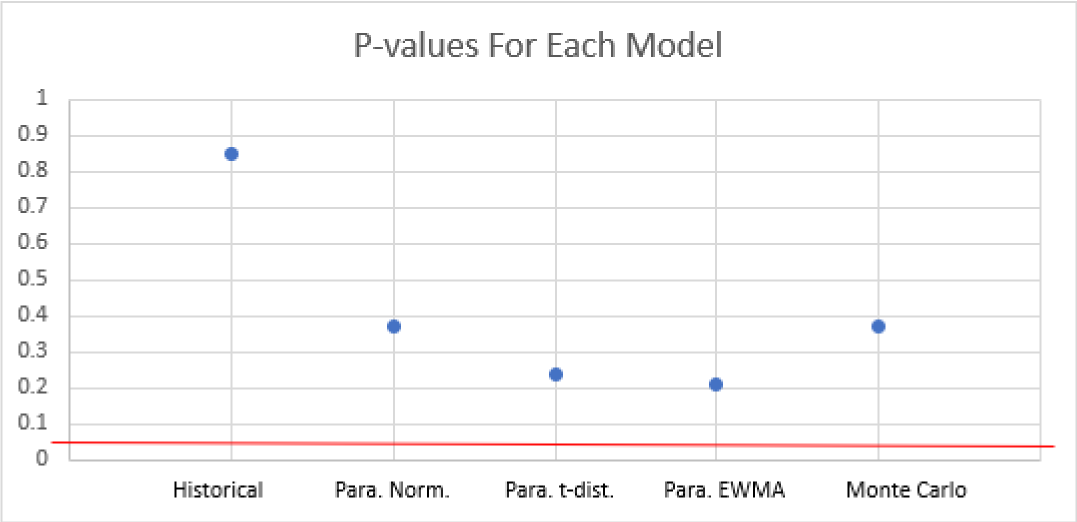


Figure 21

We experience much more pleasant results here compared to the 10% VaR test in Section 5.3.2 above.

Here all models are accepted as reliable while having promising p-values, Historical is almost perfect at 85%

Conclusion

Again, we notice that the non-Gaussian models (Historical, Parametric t-dist.) work best for the low VaR conf. level of 1%, and that the t-dist. breaks down on accuracy for the 5% conf. level, however surprisingly it proves to be accurate at 10%.

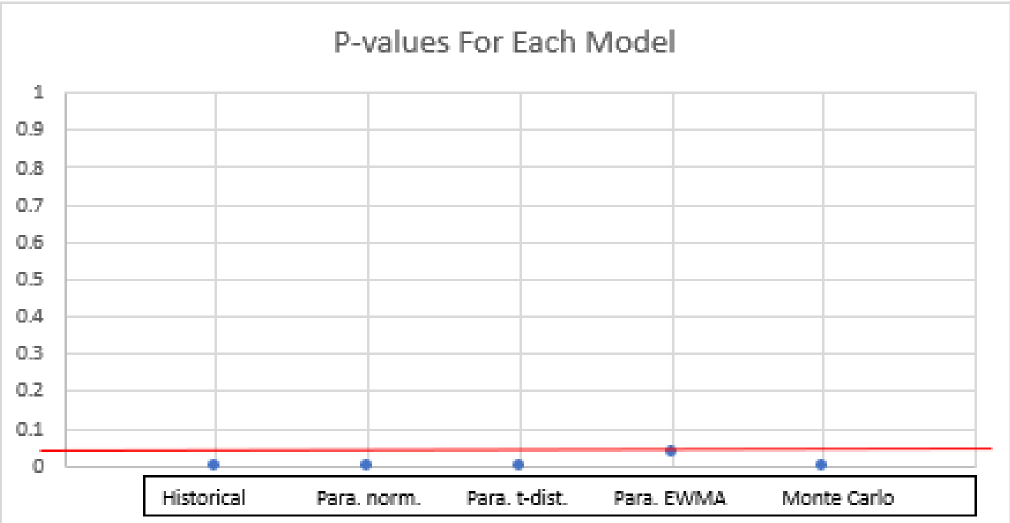
5.4 Stress testing

Let us try to run our backtests over a period where there has been heavy stress in the markets due to the Covid-19 Pandemic.

VaR conf. level = 1%

Kupiec conf. level = 5%

Method	Violations Count	Likelihood Ratio	p-value
Historical	6	16.256	0.0000006
Parametric Normal	8	26.097	0.00000003
Parametric t-dist.	7	21.015	0.0000005
Parametric EWMA	3	4.490	0.034
Monte Carlo	8	26.097	0.00000003



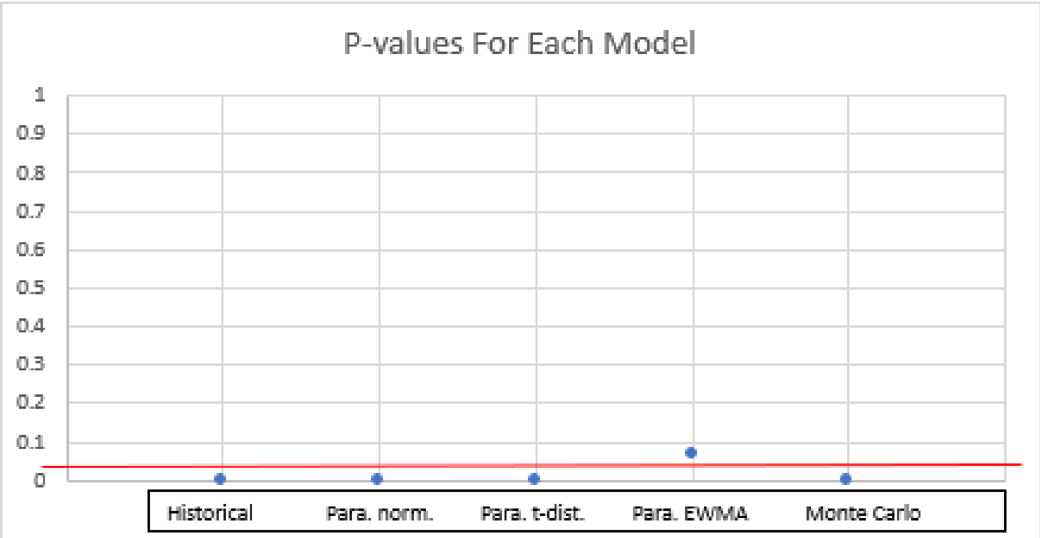
In the case of VaR conf. level = 1%, all models are rejected as inaccurate.

VaR conf. level = 5%

Kupiec conf. level = 5%

Method	Violations Count	Likelihood Ratio	p-value
Historical	14	21.035	0.0000005
Parametric Normal	15	24.003	0.00000008
Parametric t-dist.	14	21.035	0.0000005
Parametric EWMA	7	3.35	0.067
Monte Carlo	15	24.003	0.00000008

Visualising:



Generally, our models work better for VaR conf. level = 5%. But during this stressed period, the only model that is accepted as accurate is Parametric EWMA.

This could be due to the fact that the EWMA volatility estimate gives more weight to recent returns, so once the market becomes stressed, it gives more importance to it. A better method would be to use GARCH models as they are able to cluster volatilities which can happen quite often.

But to conclude, even the EWMA model performed poorly as the p-value is almost rejected. Our VaR models are not reliable during extreme market conditions.

The reason our models perform badly, is that a lot of violations occur. This is because the VaR estimate that we produce is too small and the stressed market periods have many more days with high losses. In other words, our VaR models are underestimating risk.

5.5 Conclusion

Let us summarise our results by firstly looking at how the models performed during more normal market conditions.

5.5.1 Which models work "best"?

To define which model works the best is quite difficult, there are many parameters and situations to consider with certain models being better for specific cases.

1% VaR Confidence Level:

We have noticed that most of our models do not work well here apart from the Historical and Parametric t-distribution. This is most likely due to them not following the Gaussian assumption of the other models, allowing to conform more to natural stock excess kurtosis.

Over the 3 backtests performed, our Parametric t-dist. and Historical models were the only ones to be accepted every time with average p-values of:

Method	**Avg. p-value**
Historical	17.0%
Parametric t-dist.	22.1%

While all other methods were consistently rejected as unreliable estimates, they frequented too many violations leading to the conclusion that they consistently underestimate VaR.

So, if we were to estimate VaR with 1% confidence level, the Parametric t-dist. model would be our most promising choice to use while we could also consider the Historical approach.

5% VaR Confidence Level:

At this level of confidence, we saw our models work best apart from the t-distribution assumption. Our Monte Carlo and Parametric normal models both produced very similar, if not the same, p-values as they both make the same probabilistic assumptions, which was expected. The Parametric EWMA model on average performed better at this level, with a p-value higher by 22.9% compared to Equal Weighted (EW) on average; EWMA is a good addition to our parametric model.

The Historical method sometimes performs just as good as the Parametric or Monte Carlo models, but it also had a case where it had a significantly lower p-value as can be seen in Figure 17. However, it has good performance over the 1% and 5% levels. It is surprising to see Historical working so well as it was the simplest to implement, but this is likely due to it building returns from history allowing adaptation to different environments and confidence levels.

Our Parametric EWMA model would be a great choice for estimating risk at this VaR significance level.

10% VaR Confidence Level:

At this significance, our results were fairly inconclusive. Sometimes models like Monte Carlo and Parametric Normal performed very well (Figure 18) and surprisingly the t-dist. model being accepted as reliable while other times the normal distribution didn't perform as well with the t-dist. being rejected.

Even the Historical model was nearly rejected in Figure 18, but ascertained a high p-value in Figure 21.

Overall, the p-values of the 10% level were all over the place. This could call for further testing, however due to time constraints we will have to hold off for this.

5.5.2 Stressed market conditions.

During our stress test, we noticed that almost all of our models would get rejected. Only the EWMA model slightly passed the p-value test to be accepted at 5% VaR confidence level. However, on the whole it appears that during times of chaotic markets none of our models would prove to be reliable. Perhaps we could have created a new model with EWMA for volatility estimation and the Student's t as a return distribution.

Chapter 6: **Self-Assessment**

Project Plan

During the first term I followed my pre-set project plan on time, if not completing some tasks ahead of schedule. This plan was coupled along with my diary where I entered each task completed and milestones reached. The idea behind this was to clearly document my progress to make sure that any deadlines wouldn't creep up surprisingly. It was easy to follow the plan due to the Waterfall like nature of the project, where each objective was a step up of the previous. For instance, implementing the Historical VaR method was simple and the next step to the Model-building approach was fluid. However, during Christmas the API used for getting data had broken down. This was an unforeseen incident that took me a couple of weeks to fix and finally set up a mitigation strategy by downloading data.

Towards the end of Term 2, I struggled with the incorporation of options for a couple weeks as it was an unintuitive concept for me and the literature online wasn't hopeful, but with the help of my supervisor, Dr Yuri Kalnishkan, I managed to grasp the concept.

Eventually, I completed all project goals, even managing to complete some Suggested Extensions: computing Conditional Value at Risk, complement backtesting with stress testing, and replacing Gaussian model with more robust models.

Future Enhancements

If I had more time to spend on this project, my main goal would be to focus on building different kinds of VaR models. It would be interesting to see how the Student's t-distribution coupled with EWMA volatility estimates would perform for instance. This would prove to be time consuming as each variation would have to be backtested and the significance of the results analysed.

It would additionally be interesting to backtest against a common standard that financial institutions employ. This would give more accurate results instead of blindly choosing a portfolio and time frame as done in the Experiments section.

Chapter 7: Deployment and YouTube

I have shown on the README.txt the introduction on how to get started with the VaR in Python Implementation. Then they are instructed to find the following videos:

Intro to the program: https://youtu.be/OvzyJQk_NuU

Performing VaR estimates for one stock: https://youtu.be/CUBD5ABI-q0

Note: For one stock, refer to Student's t-dist. video, as API update causes error in above.

Performing VaR estimates for three stocks: https://youtu.be/9vOOPwgkV54

First Monte Carlo calculation: https://youtu.be/47pcFNj3lXs

Second Monte Carlo calculation: https://youtu.be/OgoMuFgsW2Y

New Additions

Student's t-dist.: https://youtu.be/VeIFRh5rMeE

Options: https://youtu.be/ZQcFeTc-Dgk

Backtesting: https://youtu.be/0--qf2RGCVg

GUI: https://youtu.be/Q48N7zrk5-I

Final Thank You: https://youtu.be/VDoVW5P5-TI

Chapter 8: **Professional Issues**

Data Quality:

Value at Risk models rely heavily on historical data to produce their estimates of risk. Therefore, the completeness and accuracy of this data is critical. There cannot be any bias or errors otherwise this will badly effect the validity of the models.

We have made use of Yahoo! Finance for our source of data, they are popular amongst enthusiasts and provide a plentiful amount of information such as financial statements, currencies exchanges, stock prices, etc. We make use of their wide range of market data allowing us to choose specific points in history.

However, we should be aware that this is a free service so their data may not be comprehensive or as up to date as other financial data providers. For our intents and purposes Yahoo! Finance shall suffice, as our models aren't being used to make critical decisions affecting anyone's livelihood.

Experimentation:

When validating VaR models with the use of backtesting, it can become very biased. Depending on the assets and choice of time periods, it is possible to force certain results. Let us take stress testing, it is possible to choose companies that performed well during the Covid-19 Pandemic[21]. For instance, Amazon didn't experience a deep hit in their stock price as many other companies did. If we performed a backtest with a portfolio of Amazon, it would appear that our models work just as well during periods of stress. These results wouldn't be accurate.

In order to evaluate a model fairly, it is important to follow a set of guidelines. The Basel framework provides just this, it is a standardisation for risk management that is adopted by financial institutions. Many countries even require regulatory compliance in order for them to operate without legal action. The framework helps reduce bias as it provides tools for stress testing which have been verified amongst many financial risk experts.

Due to time constraints and the lack of real-world damage biased results could cause in this project, we do not follow any regulatory frameworks. Nevertheless, we are transparent with the backtesting data and perform tests with small and large portfolios.

Ethics:

Since the dawn of civilisations on Earth, there has always been a divide between the rich and poor. In the 20th and 21st Century we have seen great development and an increased life expectancy across the globe, with the working and middle class better off than hundreds of years ago. Although, some economists are concerned that the gap is again increasing, [23] Marinez-Toledano explains how this is worse, amongst First World countries, where more extreme capitalist economic models like in the US are in effect. She suggests that "High levels of economic inequality can lead to economic and political instability. This is why action needs to be taken before societies become polarised.". This begs for the question whether developing financial tools like VaR aid in causing inequality.

While it is true that there are a select few individuals at the top of investment banks reaping the benefits of VaR by minimising their risk in investments, it is also used by risk managers in pension and mutual funds. These funds are usually used by everyday people. As well as this, VaR is used by regulators to help prevent large institutions collapsing[24]. The side effects of large collapses affect more than just the owners, jobs are lost and ripple effects on the economy can result in higher commodity prices etc.

Conversely, to the argument of VaR saving collapses, it has been criticised for not measuring risk accurately. Mayer et al. indicated that there are common concerns, as examples: “creating over reliance on a single model” and ”creating pro-cyclical positive feedback in financial markets leading to ‘VaR shocks’”[25]. Eventually, they come to the conclusion that there is necessity to follow some sort of regulatory frameworks incorporating a solid VaR model. They even suggested the EWMA method for stress testing which is the “simplest and best”. It is promising to see that our EWMA implementation also performed the best under stressed 5% VaR in Section 5.4, however it only just passed acceptability, this is also similar to their results. Just because it is the “best” from many poorly performing models, doesn’t strike confidence in its use at such high stakes.

It is quite impossible to tell whether a Universe with or without VaR models is better, in the end these models should be incorporated with other risk measures and monitoring systems to provide maximum benefit with transparency of their use.

Bibliography

[1] Pawel Lachowicz. *Conditional Value-at-Risk in the Normal and Student t Linear VaR Model*. December 8, 2016. Retrieved from: https://quantatrisk.com/2016/12/08/conditional-value-at-risk-normal-student-t-var-model-python/#:~:text=CVaR%20h%20%2C%20%CE%B1%20(%20X%20),of%20the%20standard%20normal%20distribution

[2] Yasuhiro Yamai & Toshinao Yoshiba. *On the Validity of Value-at-Risk: Comparative Analyses with Expected Shortfall*. January 2002. Retrieved from: https://www.researchgate.net/publication/5119911_On_the_Validity_of_Value-at-Risk_Comparative_Analyses_with_Expected_Shortfall

[3] John C. Hull. *OPTIONS, FUTURES, AND OTHER DERIVATIVES*. 2002.

[4] team@interviewqs.com. *Calculating Value at Risk (VaR) of a stock portfolio using Python.* February 2020. Retrieved from: https://www.interviewqs.com/blog/value-at-risk

[5] Jacques Longerstaey, Martin Spencer *RiskMetricsTM—Technical Document*. December 17, 1996. Retrieved from: https://www.msci.com/documents/10199/5915b101-4206-4ba0-aee2-3449d5c7e95a

[6] Kevin Sheppard, Many other contributors. *pandas-datareader*. July 13, 2021. Retrieved from: https://github.com/bashtage/pandas-datareader

[7] Li, W. *Teaching VaR*. April 26, 2016. Retrieved from: https://web.mst.edu/~huwen/teaching_VaR_Weiqian_Li.pdf

[8] Eric Geveda. *What is the Square Root Rule?*. April 2008. Retrieved from: https://financetrain.com/what-is-the-square-root-rule

[9] Emerick, J. *Learn Quantitative Finance*. 2022. Retrieved from: https://quantpy.com.au/

[10] Jayati Walia. *The variance-covariance method for VaR calculation*. 17 December, 2021. Retrieved from: https://www.simtrade.fr/blog_simtrade/variance-covariance-method-var-calculation/

[11] Nicholas Burgess. *Correlated Monte Carlo Simulation using Cholesky Decomposition*. May 4, 2022. Retrieved from: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4066115

[12] Manish Sahajwani & Sheeba Manish. *Analytical Approach to Calculating VaR (Variance-Covariance Method)*. n/a. Retrieved from: https://financetrain.com/analytical-approach-to-calculating-var-variance-covariance-method

[13] Anirudha Si. *Understanding and Creating Monte Carlo Simulation*. April 24, 2017. Retrieved from: https://www.linkedin.com/pulse/understanding-creating-monte-carlo-simulation-anirudha-si/

[14] Glyn A. Holton. *Value-at-Risk Theory and Practice 2nd Edition*. 2013. Retrieved from: https://www.value-at-risk.net/backtesting-coverage-tests/

[15] Rebecca Bevans. *T-Distribution | What It Is and How To Use It and Practice*. 2023. Retrieved from: https://www.scribbr.co.uk/stats/t-distribution-meaning/

[16] Pawel Lachowicz. *Student t Distributed Linear Value-at-Risk*. 2015. Retrieved from: https://quantatrisk.com/2015/12/02/student-t-distributed-linear-value-at-risk/#:~:text=The%20Student%20t%20distribution%20is,%E2%88%92%204%20)%20%E2%88%92%201%20for%20

[17] William T. Shaw. *Risk, VaR, CVaR and their associated Portfolio Optimizations when Asset Returns have a Multivariate Student T Distribution*. 2011. Retrieved from: https://arxiv.org/pdf/1102.5665.pdf

[18] Giot, P., & Laurent, S. *Market risk in commodity markets: a var approach*. 2003. Retrieved from: https://www.sciencedirect.com/science/article/pii/S0140988303000525

[19] Timotheos Angelidis, Alexandros Benos, Stavros Degiannakis., & Laurent, S. *The use of GARCH models in VaR estimation*, 2004. Retrieved from: https://www.sciencedirect.com/science/article/abs/pii/S1572312704000103

[20] Turhan Korkmaz, Kazım Aydın. USING EVMA AND GARCH METHODS IN VAR CALCULATIONS: APPLICATION ON ISE-30 INDEX,. Retrieved from: https://content.csbs.utah.edu/~ehrbar/erc2002/pdf/P161.pdf

[21] Meera Sharma. *Evaluation of Basel III revision of quantitative standards for implementation of internal models for market risk*, 2012. Retrieved from: https://www.sciencedirect.com/science/article/pii/S0970389612000778#:~:text=Value%2Dat%2DRisk%20or%20VaR,Basel%20Accord%20II%20and%20III.

[22] Annie Palmer. *How Amazon managed the coronavirus crisis and came out stronger*, 2020. Retrieved-from: https://www.sciencedirect.com/science/article/pii/S0970389612000778#:~:text=Value%2Dat%2DRisk%20or%20VaR,Basel%20Accord%20II%20and%20III.

[23] T. Blanchet, C. Martinez-Toledano. *Wealth inequality dynamics in europe and the united states: Understanding the determinants*, 2023. Retrieved from: https://www.sciencedirect.com/science/article/pii/S0304393222001416?via%3Dihub

[24] Sakshi Shekhawat. *Value at Risk (VaR)*, 2020. Retrieved from: https://www.knightfintech.com/knowledge-center/articles/learning/value-at-risk-(var)#:~:text=Value%20at%20Risk%20(VaR)%20a,potential%20loss%20on%20the%20advances.

Appendix: Diary

For Interim submission:

My overall pace with the program went well, majority of the time I kept above or just at the Timeline I created in my project plan. The only time that I did fall back was when I had a family emergency that forced me to go abroad for a week, however when I got back, I was still on time with the schedule, and I finished the majority of the implementation I stated in the project plan. There were some exceptions like calculating VaR for Options using Monte Carlo however, I stated that this could be completed next term as I will learn about it in the module Computational Finance.

I believe the main reason for me keeping up to date was that during the summer research period, I already practiced implementing code to compute VaR with the various methods so when it came to creating my program, I was very ready.

For Final Submission:

To begin the year, I had a lot of struggles with the API as mentioned previously. It had broken down and it became very time consuming to fix, taking my progress away from the project and towards fixing a python library (pandas-datareader). In the end I managed to fix it and create backups of data, while also switching to a more robust library called yfinance.

After the fix, I moved onto backtesting and other project goals. These all went smoothly until I got to incorporating options into the portfolio. At first, I struggled as it wasn't intuitive for me, but in the end I managed to grasp it and implement it as part of the Monte Carlo simulation. I also managed to complete some suggested extensions: stress testing, using Student's t-distribution instead of Gaussian, computing Expected Shortfall. I also didn't believe I would be able to create a GUI as I struggle with this aspect of computer science, however it was possibly the simplest part of the project.

The following is my diary:

9th-October-2022:

Created a project layout that I experimented with over the summer.

- main risk package will contain a calculate class where mathematical VaR will be computed

- var class will get data and store in a VaR class with appropriate paramters.

10th-October-2022:

Set up TDD test for getting data

Created a get_data function that will access Yahoo API and calculates appropriate paramters.

11th-October-2022:

I struggled with setting up an Object Oriented Design at the beginning of the week which took the majoirty

of my time.

But, now I have a VaR class that uses __init__ to read yahoo API and perform relevant paramter calculations.

Therefore there is no need for get_data().

12th-October-2022:

historical_var() has been created which computes historical VaR calculation for a given start and end date, at a

given confidence level over a specified VaR time period.

I should put in more though into the way of storing the dates, conf level, and VaR time period.

13th-October-2022:

historical_cvar() has been added, This computes the Expected Shortfall (conditional value at risk).

The CVaR returned shows that the expected loss is greater, which makes sense as the expected shortfall is

at a lower percentile than the VaR.

I had some troubles feeding the right data in historical_cvar() because I needed to call the

historical_var() functuon with self, and keep the same return values.

Testing :

New UnitTest was added to replace get_data as the function was replaced.

I need to add more tests to check VaR results / function.

I need to add documentation to all functions / classes.

14th-October-2022:

Feature :

portfolio_analysis() in VaR Class can compute mean*weighted return and standard deviation over a specified

time for a portfolio.

Testing :

Added a UnitTest to make sure the Conditional VaR is lower than VaR for the specific data that was

pre-determined.

15th-October-2022:

Feature :

Paramateric ValueAtRisk can now be calculated in addition to Parametric Condtiontional Value At Risk,

These functions have the attribute of period of time to calculate VaR for however with historical I store

t-Time outside Class scope so this could be something to standardise as part of a refactor

Testing :

I have added a Unit Test to check that the CVaR is smaller than VaR for the pre-determined time frame.

18th-October-2022:

Feature :

Now it is possible to calculate volatility using Exponentially Weighted Moving Average (EWMA) formula.

This works by calculating the volatility for each stock in returns.columns including 'portfolio' returns

however, I dont know how to combine the volatilties for a single VaR of portfolio but for now it can find

it for each stock series in returns DataFrame.

The volatility from EWMA calculation is quite close to Standard Deviation so this gives me a good idea that

my calculation is right

I can later further test this by using pandas.ewm possibly.

19th-October-2022:

Feature :

I have implemented a method parametric_ewma_var() which uses EWMA volatility calculation from yesterday.

I have not yet found out if the portfolio calculation is valid

I calculate VaR for each stock including the portfolio column and these results are similar to my

Gaussian calculation though.

23rd-October-2022:

Feature :

New function where I have implemented the formula:

Daily Returns t = MeanReturn + LZt

where Zt are samples from normal distribution, L is a decomp. of covariance matrix using Cholesky Decomp.

This allows me to simulate a stock portfolio value over time as a Monte Carlo simulation.

Next step:

Compute VaR of the Monte Carlo simulation by looking at the bottom percentile, a bit more reading for me

is required.

Mistake :

I have been working on the project thus far without branches as I just completely forgot to use them for new

features such as implementing historical, parametric VaR estimates. However I have now created a Monte Carlo

branch which I will use for that deliverable and the rest of my project.

2nd-November-2022:

Feature :

New function where I create a similar function to that of Historical VaR to compute the alpha percentile

of the distribution from the series of portfolio simulations. This computes the VaR for the Monte Carlo Sim method. The series of portfolio simulations is computed in 23rd oct update.

9th-November-2022:

Created layout for interim report.

12-November-2022:

Started writing about background theory, more specifically the definition of VaR.

13-November-2022:

Finished defintion of VaR and also CVaR.

15-November-2022:

Now interim report is up to 1500 words. Started describing the theory of historical simulation.

25-November-2022:

Interim report describes MonteCarlo sim theory as well as model-building approach.

Next step is to describe the differences in variance calculations and covariance matrix.

27-November-2022:

Background theory is basically completed.

28-November-2022:

Feature: Added a function monte_carlo_cvar() to compute expected shortfall of Monte Carlo simulation, I previosly

forgot to implement this but now the results look good.

15-December-2022

Python API broke down

16-December-2022

Investigating Issues, seems to be encoding of data in AES by Yahoo! Finance

Results : The Python Library used for getting market data, pandas-datareader, stopped working in Christmas of

2022, The reason for this is due to Yahoo! encoding some data on their webpage under AES.

22-December-2022

Working on fix, seems Unlikely

1-January-2023

Imported a fix from another library and made some workaround for pandas-datareader for it to work

15-January-2023

Downloaded data into CSV files so that there is redundancy :

I chose to download a few stock's pandas DataFrames in CSV format and storing them in a separate directory with an

associated Python file that is able to download the data.

20-Janyary-2023

Started working on backtesting :

Researching : Kupiec and Standard coverage tests

Update : violation_count method in Backtest class returns dict of violations and new method proportion() calculate

the proportion of these violations compared to the number of testing days.

25-January-2023

Feature :

Kupiec POF test implemented, works with p-values by computing chi squared with survival function.

3-February-2023

Feature :

Standard coverage test implemented, works similar to Kupiec test and gives very similar p values.

16-February-2023

Feature :

Started using PySimpleGUI to create a simple GUI for the var.py file

So far opens a window to take in user input

22-February-2023

Feature : GUI is fully functional for all VaR estimates and can take in any user input to create a portfolio of

stocks

28-February-2023

Update : Finding a solution to value options, Can compute prices of options with three methods :

Black - scholes formulas

binomial treess

Monte Carlo simulation

5-March-2023

Update : Now can simulate a portfolio with an option on it for a selected stock.

Refactor : Creating OptionPricing class and using it instead.

12-March-2023

Update : Adding student-t distribution for VaR using parametric method.

Works for VaR and CVaR