Value at Risk: Assignment 3

Jake Mc Leroth (24913693)

14/04/2025

Contents

Introduction

This assignment looks at the use of Monte Carlo simulation as a method of calculating value at risk (VaR), and based on chapter 4 of Alexander (2009). First, we illustrate how the time-varying volatility model A-GARCH (Asymmetric Generalized AutoRegressive Conditional Heteroskedasticity) can be used in a simulation context. Then, we simulate returns using a multivariate normal distribution as well as a mixture distribution. An advantage of Monte Carlo VaR is that it is not constrained by data limitations, since you can simulate as much data as you want. Unfortunately, this also introduces model risk, as you need to make assumptions about the behaviour of the risk factors that you want to simulate. Note that since all values are calculated using Monte Carlo simulation in this assignment, they will differ ever so slightly each time they are calculated due to random sampling. In all three sections, the value n denotes the number of simulations, and the value h denotes the time horizon in which we calculate VaR.

Question 1: Multi-Step Monte Carlo

In this section, we compare a multi-step Monte Carlo simulation using A-GARCH to model volatility with a simulation using a constant volatility.

The A-GARCH model is given in Alexander (2009) as:

$$\begin{aligned}\sigma_t^2 &= \omega + \alpha\left(r_{t-1} - \lambda\right)^2 + \beta\sigma_{t-1}^2 \\ r_t &= \varepsilon_t \sigma_t\end{aligned}$$

Here, r_t and σ_t are daily values. To simulate using this model, we assume that ε_t follows a standard normal distribution and can hence simulate returns this way. An algorithm of this process is as follows:

1. Initialise $\sigma_0 = \sqrt{\frac{\omega+\lambda^2\alpha}{1-(\alpha+\beta)}}$ (this is the long-term daily volatility)
2. Initialise r_0 as the most recent return
3. For t = 1,2,...,h:
   - Simulate the next return: $r_t = \varepsilon_t \sigma_t$ where $\varepsilon_t \sim N(0,1)$
   - Calculate the next variance: $\sigma_t^2 = \omega + \alpha\left(r_{t-1} - \lambda\right)^2 + \beta\sigma_{t-1}^2$
4. Sum the h returns to get an h-day return
5. Repeat steps 3 and 4 n times to get a distribution of h-day returns.

We use the parameters given in example IV.4.7 in Alexander (2009):

- $\omega = 4 \times 10^{-6}$
- $\alpha = 0.06$
- $\lambda = 0.01$
- $\beta = 0.9$

This corresponds to a long-term daily volatility of 1.581139%, or an annual volatility of 25%. We assume that the most recent return is 10% and then compare that to a starting value of -10%. To simulate returns when volatility is constant, we use $r_t = \varepsilon_t \sigma$, where ε_t is again from a standard normal distribution. We assume that the annual volatility is 25%. Calculations were done in R, shown in "q1.R". They were also checked in C++, in the file "q1.cpp". A value of $n = 1000000$ (5000000 simulated values) was used to estimate the 5-day VaR ($h = 5$). For an accurate comparison, the same set of simulations from the standard normal distribution is used for all the VaR calculations (the same is done in the next two sections).

The values for the different VaR calculations are shown:

Model	**0.1%**	**1%**	**5%**	**10%**
Constant Volatility VaR	0.1092	0.0823	0.0582	0.0454
A-GARCH (Positive Shock)	0.2037	0.1444	0.0974	0.0743
A-GARCH (Negative Shock)	0.2334	0.1656	0.1119	0.0855

Table 1: MC A-GARCH v.s. Constant volatility VaR

We observe that the A-GARCH VaR is much higher than the constant volatility VaR for all α values, especially in the tails. The fatter tails are from the A-GARCH accounting for volatility clustering and the leverage effect. The effect of the starting value is highlighted in the last two rows, whereby the A-GARCH with a starting return of -10% results in a significantly higher VaR than when using a starting value of 10%. To confirm the results, C++ was used to do all calculations again. They are as follows:

VaR Method	0.1%	1%	5%	10%
Constant Volatility	0.1094	0.0825	0.0582	0.0453
A-GARCH (Positive Shock)	0.2029	0.1446	0.0974	0.0741
A-GARCH (Negative Shock)	0.2324	0.1658	0.1119	0.0852

Table 2: VaR from C++ code

We see that the values are all very close, with the small differences being attributed to sampling error.

Question 2: Multivariate Normal Monte Carlo VaR

This section shows the process of simulating a multivariate normal distribution to calculate the VaR of a portfolio with five underlying risk factors. This is based on example IV.4.8 in Alexander (2009).

The portfolio information is as follows:

Factor:	**A**	**B**	**C**	**D**	**E**
Expected excess return (%)	6.0	4.0	3.0	2.0	1.0
Risk factor volatilities (%)	15.0	20.0	12.0	10.0	18.0
Risk factor sensitivities	0.80	0.40	0.30	0.10	-0.10
Correlation Matrix					
	A	B	C	D	E
A	1.00	0.30	0.20	0.10	0.15
B	0.30	1.00	0.25	0.20	0.10
C	0.20	0.25	1.00	0.30	0.25
D	0.10	0.20	0.30	1.00	0.35
E	0.15	0.10	0.25	0.35	1.00

Table 3: Updated Factor Information and Correlation Matrix

We calculate the covariance matrix:

$$\Sigma_{annual} = \begin{bmatrix} 0.15 & 0 & 0 & 0 & 0 \\ 0 & 0.20 & 0 & 0 & 0 \\ 0 & 0 & 0.12 & 0 & 0 \\ 0 & 0 & 0 & 0.10 & 0 \\ 0 & 0 & 0 & 0 & 0.18 \end{bmatrix} \begin{bmatrix} 1.00 & 0.30 & 0.20 & 0.10 & 0.15 \\ 0.30 & 1.00 & 0.25 & 0.20 & 0.10 \\ 0.20 & 0.25 & 1.00 & 0.30 & 0.25 \\ 0.10 & 0.20 & 0.30 & 1.00 & 0.35 \\ 0.15 & 0.10 & 0.25 & 0.35 & 1.00 \end{bmatrix} \begin{bmatrix} 0.15 & 0 & 0 & 0 & 0 \\ 0 & 0.20 & 0 & 0 & 0 \\ 0 & 0 & 0.12 & 0 & 0 \\ 0 & 0 & 0 & 0.10 & 0 \\ 0 & 0 & 0 & 0 & 0.18 \end{bmatrix}$$

$$= \begin{bmatrix} 0.02250 & 0.00900 & 0.00360 & 0.00150 & 0.00405 \\ 0.00900 & 0.04000 & 0.00600 & 0.00400 & 0.00360 \\ 0.00360 & 0.00600 & 0.01440 & 0.00360 & 0.00540 \\ 0.00150 & 0.00400 & 0.00360 & 0.01000 & 0.00630 \\ 0.00405 & 0.00360 & 0.00540 & 0.00630 & 0.03240 \end{bmatrix}$$

To simulate a multivariate normal distribution, we need to account for correlations between variables. According to Rizzo (2019), if $\underline{Z} \sim N_d(\underline{0}, I_d)$, then the transformation $Q\underline{Z} + \underline{\mu} \sim N_d(\underline{\mu}, QQ')$. Hence, we decompose the covariance matrix using the Cholesky decomposition, $\Sigma = QQ'$, and use this to simulate the correlated risk factor returns (Alexander, 2009). Here Q is a lower triangular matrix.

In the R script "q2.R", we calculate the 10 day Cholesky matrix and use this to simulate returns. Risk factor mapping is applied using the risk factor sensitivities to get a return distribution for the portfolio, from which the 1% 10-day VaR is calculated. Again, we do 1000000 simulations for each risk factor (n = 1000000). The 1% 10-day VaR is calculated as 7.866522%.

Question 3: Normal Mixture Monte Carlo VaR

In this section, we explore Monte Carlo VaR using a normal mixture distribution. This scenario is based on example IV.4.13 in Alexander (2009). We have the following values for an equally weighted portfolio:

	Crash market			Ordinary market		
	Stock 1	Stock 2	Stock 3	Stock 1	Stock 2	Stock 3
Expected returns	−40%	−55%	−65%	7%	12%	10%
Volatilities	50%	60%	55%	20%	25%	22%
Correlation matrix	1.00	0.85	0.80	1.00	0.40	0.60
	0.85	1.00	0.75	0.40	1.00	0.30
	0.80	0.75	1.00	0.60	0.30	1.00
Regime probability		0.05		0.95		

According to Alexander (2009), we can simulate h-day returns from a normal mixture as follows:

1. Calculate the h-day covariance matrices for both markets, Σ_{crash} and $\Sigma_{ordinary}$
2. Calculate the h-day lower triangular Cholesky decomposition of the covariance matrices, Q_{crash} and $Q_{ordinary}$
3. Simulate a vector $\underline{z} \sim N_d(\underline{0}, I_d)$ - a vector of d standard normals
4. Simulate a value from a Bernoulli distribution with probability 0.05
5. If the Bernoulli value equals 0:
   - Calculate return vector as $\underline{r} = Q_{ordinary}\underline{z} + \underline{\mu}$
   - This represents a return in ordinary market conditions

   If the Bernoulli value equals 1:
   - Calculate return vector as $\underline{r} = Q_{crash}\underline{z} + \underline{\mu}$
   - This represents a return in crash market conditions
6. Repeat steps 3 to 5 n times

This gives us a distribution of returns, simulated from a normal mixture, which we use to calculate the VaR. We compare this with a normal distribution, where the mean and covariance are calculated from the normal mixture values. This is all done in "q3.R". The results are as follows:

Confidence Level	0.1%	1%	5%
VaR (Normal)	0.1241	0.0929	0.0650
VaR (Mixture)	0.2317	0.1113	0.0600

Table 4: 10-day Value-at-Risk (VaR) under Ordinary and Mixture Distributions

As expected, the tails of the mixture VaR are larger than the normal VaR because of large negative returns during crashes. At less extreme values, the normal VaR starts to get larger than the mixture VaR. This tells us that a mixture distribution is likely better than a normal distribution for VaR, since it captures the leptokurtic effects of returns that are observed in practice.

Conclusion

This assignment has shown the methodology of different ways to use Monte Carlo simulation to calculate VaR. The value of Monte Carlo is that you can get lots of data and model returns through time. The best use case of Monte Carlo for VaR is thus to make use of a model that can simulate volatility through time, capturing volatility clustering and the leverage affect. This allows us to better account for tail risk and leads to better VaR estimates.

References

Alexander, C. (2009). *Market risk analysis, value at risk models.* John Wiley & Sons.

Rizzo, M.L. (2019). *Statistical computing with R.* Chapman and Hall/CRC.