VaR Prediction for Multi-Asset Investment Portfolios Based on Copula, GARCH, and Traditional Models

Group O

Jiatong Chen	2034631
Zihao Li	2037161
Changqing Lin	2039153
Yun Rong	2034422
Han Zhang	1929916
Jialu Wang	2036191

GROUP MEMBER CONTRIBUTION FORM

FIN305 – Group Assignment

The purpose of this document is to provide each group with an opportunity to reward or punish an individual's contribution to the group. All members of the group should discuss this form, fill in this form as directed, sign this form, and submit it with the report as the coversheet.

We agree that all group members made a valuable contribution. Please adjust our grades based on the following percentage of contribution.

Individual Name (print):	**%Contribution to the group project**	**Authorship contribution statement1**
Han Zhang	**16.666667%**	
Changqing Lin	**16.666667%**	
Zihao Li	**16.666667%**	
Jiatong Chen	**16.666667%**	
Yun Rong	**16.666667%**	
Jialu Wang	**16.666667%**	
	100%	

Signatures:

1. 林长青 Changqing Li
2. 陈佳彤 Jiatong Chen
3. 戎贇 Yun Rong
4. 张翰 Han Zhang
5. 李梓豪 Zihao Li
6. 王嘉璐 Jialu Wang

1. Introduction

In recent years, major global incidents like the European debt crisis and the COVID-19 outbreak have significantly disrupted the world's financial markets. As Jones (2019) observed, European countries faced challenges in managing budget deficits, particularly in securing funds through usual methods like international bond sales. Similarly, Banerjee (2022) noted the widespread turmoil in financial markets triggered by the pandemic, leading to a state of near-collapse globally. Investopediaa (2016) emphasized that such negative market trends increase the uncertainty of financial losses. Mourik (2003) identified various market risks, including Equity and Interest rate risks, as inherent in these turbulent times. To counter these risks, investors have turned to diversifying their portfolios, a strategy recommended by Investopediab (2016) for managing equity risk. Wohlner (2013) compared this to avoiding putting all eggs in one basket, suggesting a mix of investment targets from varied categories and sizes to lower unsystematic risks, thereby minimizing potential financial losses.

The purpose of this project is to construct and analyze an equally weighted portfolio comprising diverse asset classes, specifically targeting the SPDR S&P 500 ETF Trust (SPY), iShares 20+ Year Treasury Bond ETF (TLT), and SPDR Gold ETF. The selection of these assets is grounded in both their inherent characteristics and their relevance to significant financial events. The SPY offers exposure to a broad range of U.S. equities, reflecting the general market dynamics and investor sentiment (Patel, Pereira & Delurgio, 2018). In contrast, the TLT, representing long-term U.S. Treasury bonds, typically exhibits an inverse relationship with the stock market, providing a hedge against equity volatility (Wang & Lee, 2019). Lastly, gold, as represented by the SPDR Gold ETF, is traditionally viewed as a safe haven during times of financial distress and inflationary pressures, adding another layer of diversification to the portfolio (Bhatia, Das & Kumar, 2020).

2. Methodology

2.1 GARCH

The Generalized Autoregressive Conditional Heteroskedasticity model, or simply the GARCH model, was collaboratively proposed by Robert Engle and Tim Bollerslev in 1986. Built upon the foundation of the ARCH (Autoregressive Conditional Heteroskedasticity) model, they further dissected lagged variances, thereby bringing forth the GARCH model. In recognition of Engle's significant contribution to the field through the ARCH model, he was rewarded with the prestigious Nobel Prize in Economics in 2003.

The general form of the GARCH model can be articulated as:

$$\sigma^2(t) = a_0 + a_1\varepsilon^2(t-1) + b_1\sigma^2(t-1)$$

Within this model, $\sigma^2(t)$ represents the variance of period t, $\varepsilon^2(t-1)$ signifies the squared residuals from the previous period, a_0 represents a constant, while a1 and b_1 serve as coefficients respectively for the ARCH and GARCH terms. The coefficient a1 visualizes the influence flowing from past residuals to the current variance, whereas b_1 illustrates the persistence of influence from past variance to the current variance.

In empirical analyses, we traditionally categorize the GARCH model as GARCH(p,q), with 'p' referencing the ARCH term and 'q' indicating the GARCH term. A GARCH model, like GARCH(1,1), constitutes of one ARCH term and one GARCH term, with the parameters a0, a1 and b1 being estimated through the methodology of maximum likelihood estimation. Therefore, we need to construct the mean equation and variance equation for the GARCH model. Finally, the criterion Akaike Information Criterion (AIC) is used to determine which model is best to fit the data series.

$$AIC = -2 * LogLik + 2 * NE$$

where $LogLik$ is the log-likelihood value and NE equals the number of estimates. The best fit GARCH model for each series is selected based on the smallest AIC value.

2.2 Gaussian/student-t Copula:

The Gaussian copula is a type of elliptical copula that uses the multivariate normal distribution to model the dependence structure between variables. It is defined by a correlation matrix $\sum$, which specifies the linear dependence between the variables. Mathematically, the Gaussian copula for a d-dimensional vector $u = (u_1, u_2, \dots, u_d)$ is given by:

$$C_{Gauss}(u) = \emptyset_{\Sigma}(\emptyset^{-1}(u_1), \dots, \emptyset^{-1}(u_d))$$

Where $\emptyset_{\Sigma}$ is the cumulative distribution function (CDF) of the multivariate normal distribution with mean vector zero and correlation matrix Σ. $\emptyset^{-1}$is the inverse CDF (quantile function) of a univariate standard normal distribution. $u_i$is the marginal CDF of the i-th variable transformed to a uniform distribution on [0,1]. The Gaussian Copula, however, has been subject to critique for its inability to model the dependence of extreme events accurately. To address these limitations, the use of a t-copula is often recommended, which can adequately model the lower and upper tail dependencies (Mittal, Pradhan, and Tiwari, 2021). The t-copula is described as:

$$C_{v,\Sigma}^{t}(u) = T_{v,\Sigma}(T_v^{-1}(u_1), T_v^{-1}(u_2), \dots, T_v^{-1}(u_d))$$

Where: $T_{v,\Sigma}$ is the CDF of the multivariate t-distribution with v degrees of freedom and scale matrix Σ. T_v^{-1} is the inverse CDF of the univariate t-distribution with v degrees of freedom. u_i is the marginal CDF of the i-th variable transformed to a uniform distribution on [0,1].

VaR Estimation:

Value at Risk (VaR) is a statistical measure that quantifies the potential loss in value of a risky asset or portfolio over a defined period for a given confidence interval. Mathematically, VaR can be expressed as:

$$VaR_{\partial}(X) = -inf\{x \in R: P(X \leq x) > \partial\}$$

Where $VaR_{\partial}(X)$ is the VaR at the confidence level ∂; X represents the loss distribution of the portfolio; P denotes the probability; inf stands for the infimum, which is the greatest value that is less than or equal to all values of x for the given probability condition.

Linkage with Gaussian Copula:

When using a Gaussian copula, the joint distribution of asset returns is assumed to be multivariate normal. This allows for the simulation of portfolio loss distributions under the assumption of normality and linear correlations. The Gaussian copula helps to estimate VaR by generating a large number of potential outcomes for asset returns, then determining the percentile that corresponds to the VaR confidence level (Springer, 2021).

Linkage with Student-t Copula:

The Student-t copula, which captures tail dependence, is particularly useful for VaR estimation during times of market stress when asset returns exhibit fat tails and are not well modeled by a Gaussian distribution. By simulating returns using a Student-t copula, one can generate a loss distribution that better accounts for extreme events. The VaR is then calculated from this distribution, providing a more conservative and potentially more accurate risk measure (Lourme & Maurer, 2017).

3. Monte Carlo Simulation:

In this study, Monte Carlo simulation is employed to forecast portfolio VaR using a GARCH model and the performance is assessed through back-testing which involves comparing daily losses with daily VaR estimates. We have read other paper for additional study. For instance, the research conducted by Angelidis and Benos (2005) applied this methodology in their study to assess the VaR of Greek stocks.

Further enhancing the accuracy of forecasts, some studies have adopted methods combining the GARCH model with various Copula models. Lu, Lai, and Liang (2014), for example, used a time-varying Copula-GARCH model to estimate the VaR of a portfolio in energy futures markets.

4. Traditional Methods

Historical simulation is a non-parametric method that uses historical price data to estimate the potential loss of an investment or portfolio. It assumes that future returns will follow the same distribution as past returns and calculates VaR based on the historical distribution of returns.

A parametric approach is one that assumes that the returns on an asset or portfolio follow a particular distribution (e.g. normal). It uses statistical techniques to estimate the parameters of the distribution and calculates VaR based on these parameters. To check the goodness of fit of the approach we use Backtesting methods.

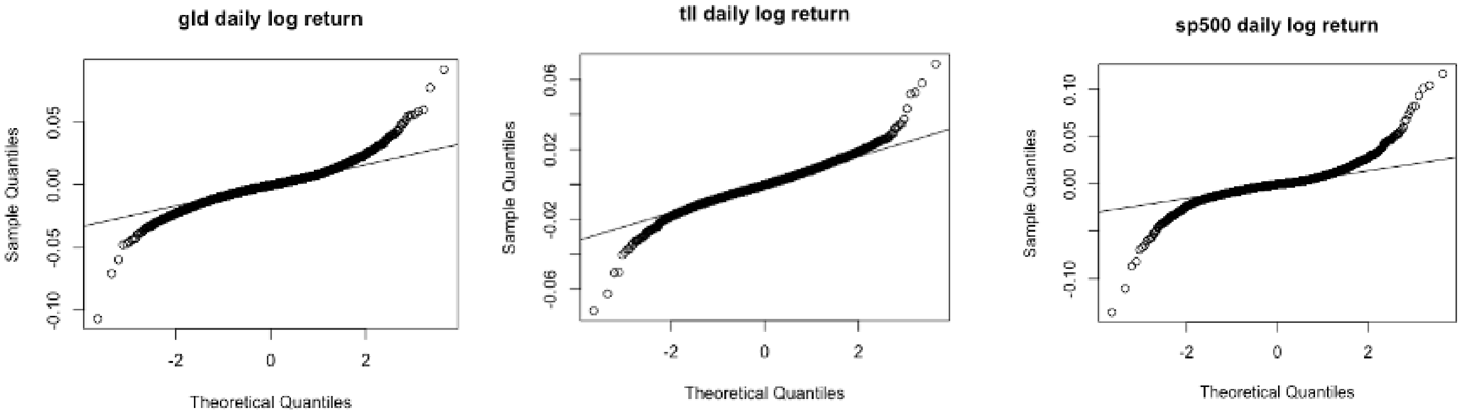
3. Data

Our data comes from Bloomberg Terminal. This portfolio consists of long-term U.S. Treasuries (TLT), gold and the S&P 500 ETF Trust-SPY. The S&P 500 ETF was chosen because this is a broad measure of U.S. stocks, and stocks of companies from a wide range of industries are included here, reflecting the general volatility of the market under different scenarios. Long-term U.S. Treasury EFTs were chosen because the performance of Treasuries is generally negatively correlated with equities, but unlike the trading logic of commodities (futures/gold) and other categories, the logic of bonds is interest rate oriented; and long-dated bonds are more sensitive to interest rates than shorter-dated bonds, which amplifies our findings and meets the time horizons required for this project. In addition, gold is a strong safe haven for developed markets such as the US and European countries. Gold moves inversely to stocks during market stress, thus reducing overall losses and balancing portfolio performance (Baur and McDermott ,2010).

We simply process the dataset, such as calculating the log-return of all trading days, merging them with the same trading days, removing the nulls, and dividing the obtained dataset into TRAIN and TEST categories, and at the same time obtaining the mean, standard deviation, skewness, and kurtosis of the overall samples of the three stocks (GLD, S&P500, and TLT), and the data are shown in the Table 1. Since the normal distribution skewness is equal to 0 and kurtosis is equal to 3, it can be concluded that s&p500 has the highest kurtosis, and the other two are not very different; gld and s&p500 skewness are both greater than 0 and are both right-skewed distributions, and only TLT skewness is less than 0, and is a left-skewed distribution. As can be seen by the following three graphs, s&p500 has the most scatter that does not fall on a straight line and is the most lacking in normality.

	Mean	Standard Deviation	Skewness	Kurtosis
GLD	-0.000267423	0.011261452	0.269883390	6.715777517
S&P500	-0.0003209349	0.0127703495	0.3862758739	14.5466004611
TLT	-0.0001348477	0.0093313747	-0.0123890118	4.1829846526

Table 1



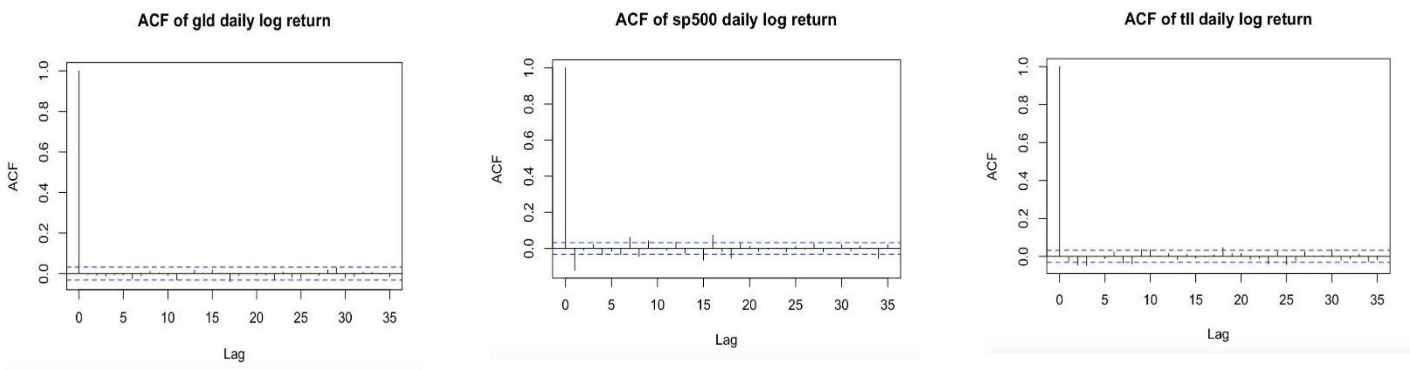
4. Empirical results

4.1 GARCH specification and derivation

First, we need to ascertain the presence of ARCH effects in the time series data, which can be preliminarily determined by examining the autocorrelation and partial autocorrelation plots. Upon confirmation of the existence of ARCH effects, we shall proceed to construct a GARCH model. The model construction entails two steps: establishing an ARMA(p,q) model initially, followed by the formulation of the variance equation, i.e., the GARCH(m,n) model. When selecting the most suitable parameters, we can utilize the auto.arima function in RStudio to search for the optimal p and q, and conduct a grid search based on the AIC criterion to determine the best m and n, thus obtaining the most suitable GARCH model.

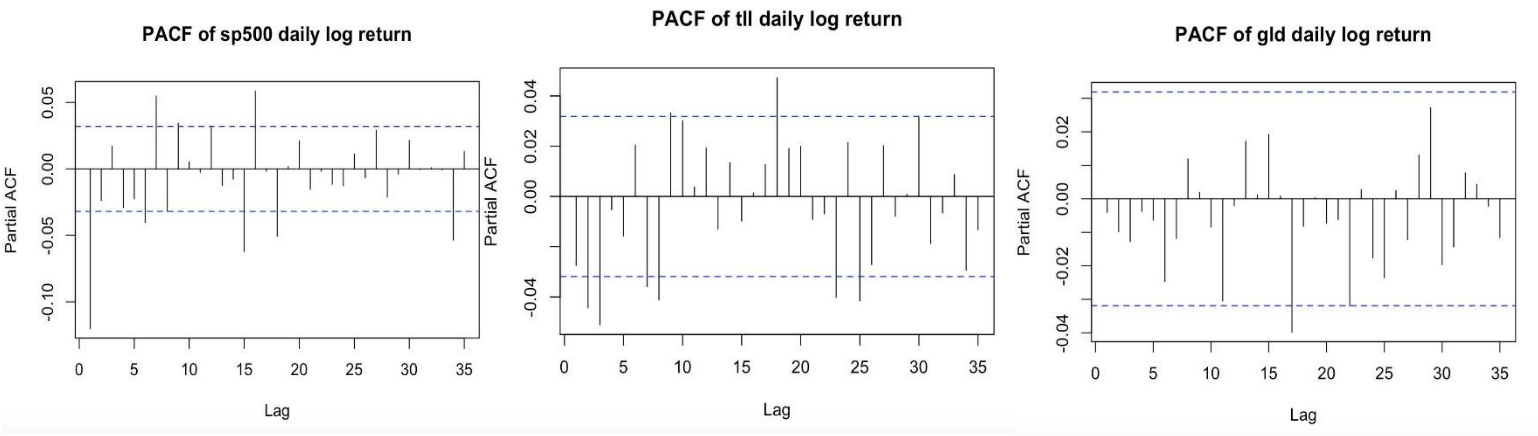
In our study, we initially examined whether the three selected asset projects exhibit the so-called ARCH effect (autoregressive conditional heteroskedasticity effect). If an asset possesses the ARCH effect, it implies that its return is related to past returns and shocks (such as sudden market events). To determine the presence of the ARCH effect, we employed mathematical tools to compute the autocorrelation and partial autocorrelation coefficients of these asset projects. The autocorrelation coefficient is used to describe the similarity between a sequence and its lagged sequence, while the partial autocorrelation coefficient can help us understand the relationship between a sequence and its specific lagged sequence, considering the influence of other lagged sequences.

Following the acquisition of these coefficients, we constructed corresponding autocorrelation function (ACF) and partial autocorrelation function (PACF) plots, to facilitate our understanding of these relationships from an intuitive, visual perspective.
In the realm of time series analysis, autocorrelation is indicative of the correlation between observations at different points in time. The ACF plot, with lag order on the horizontal axis and autocorrelation coefficient values (ranging between -1 and 1) on the vertical axis, displays the autocorrelation coefficient values at different lag orders in the time series.



As per the displayed illustration, it is discernible that a portion transcends the dotted line within the autocorrelation function (ACF) graphs of the trio of assets. This indicates that these assets all adhere to the prerequisites of the moving average (MA) model. In essence, the moving average model signifies a methodology for manipulating temporal sequence data. If an asset's ACF eclipses the dotted line, it implies that a certain degree of regularity exists in its data, rendering it apt for processing and forecasting using the moving average model.

Partially autocorrelation function (PACF) serves as a statistical instrument utilized for delineating varying lag orders within temporal sequence data. In the realm of temporal sequence analysis, partial autocorrelation depicts the correlation between observations at two distinct temporal points, post the elimination of influences from other lag orders. The PACF graph, with lag orders (lag) defining the abscissa, and partial autocorrelation coefficients (values ranging between -1 to 1) outlining the ordinate, exhibits the values of partial autocorrelation coefficients under various lag orders in a temporal sequence.



Upon scrutinising the diagram, it becomes manifestly clear that portions of the partial autocorrelation function (PACF) graphs for each of the three assets surpass the delineated hypothetical line. This indicates an adherence to characteristics typifying the autoregressive (AR) model in all three assets. In other words, the future valuations of these assets can be prophesied based upon their historical values.

Subsequently, we embark on the construction of the mean equation for the GARCH model, specifically the ARMA (p,q) model. The auto.arima function is utilised within the data to autonomously uncover and validate the most fitting 'p' and 'q' values—demonstrating adequacy. The data residuals should not possess correlativity, therefore the 'p-value' ought to exceed 0.05. Conversely, the squared residuals should show correlativity (indicating the presence of the ARCH effect, for which GARCH modelling is employed), hence the 'p-value' should be less than 0.05.

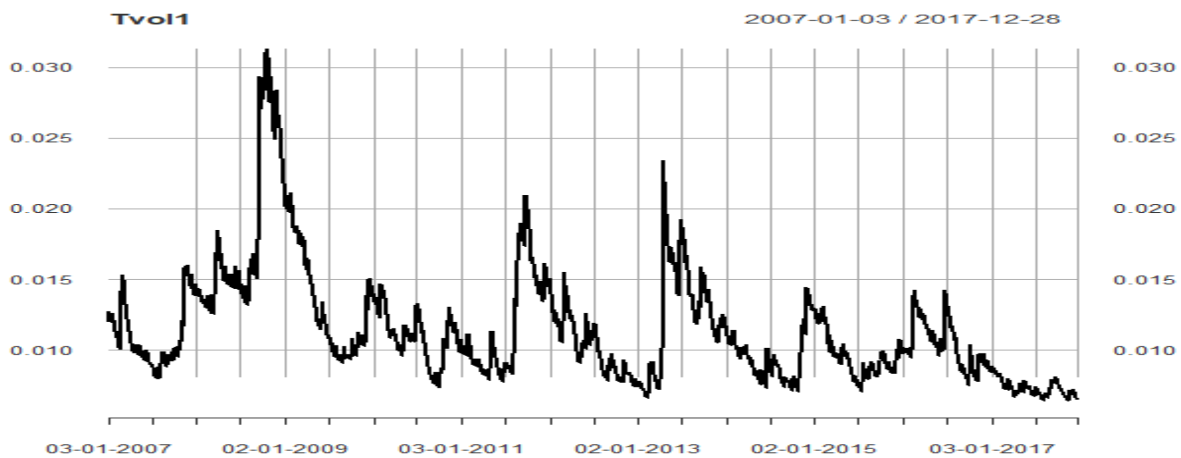
GLD

Pertaining to the asset GLD, the calculations yield p=0 and q=0. Upon conducting an adequacy validation, the ensuing results are yielded:

Residual Elements of the Data	p-value=0.704
Squared Residuals of the Data	p-value<= 2.2e-16

The results indicate that p and q align with our expectations.

We are endeavouring to establish a GARCH (m, n) model, with the primary objective of eradicating the ARCH effect in residuals of an ARMA (p, q) model, and subsequently ascertain the suitable values for m and n. Utilising the minimalisation of AIC through a grid search method, we meticulously select from an array of potential m and n values. Throughout this process, we employ a double-loop system, fitting a GARCH model for each pair of m and n, concurrently calculating their respective AIC values. The outcome indicates the optimal values for m and n are 2 and 1, thus GARCH (2, 1) emerges as the superior choice. Subsequently, we employ the identified p, q, m, and n values to construct the optimal mean and variance equations. Post model fitting, we evaluate and visualise the results to scrutinise the quality of the model. Initially, we forecast in-sample data (as we've partitioned our data into a set for sampling and a set for forecasting), then create a time series object, ultimately visualising the conditional volatility using the plot function. Through this visualisation, we gain a more lucid understanding of the model's predictive capacity, as well as volatility in the time series, thereby determining whether our model is effective and capable of accurately forecasting future variations.



Furthermore, the examination of the lack of correlation in the estimated GARCH residuals should yield a p-value greater than 0.05, indicating that there should be no correlation in the standard residual squares (eliminating the ARCH effect).

Residuals of the Data	p-value = 0.9346
Squared Residuals of the Data	p-value = 3.042e-05

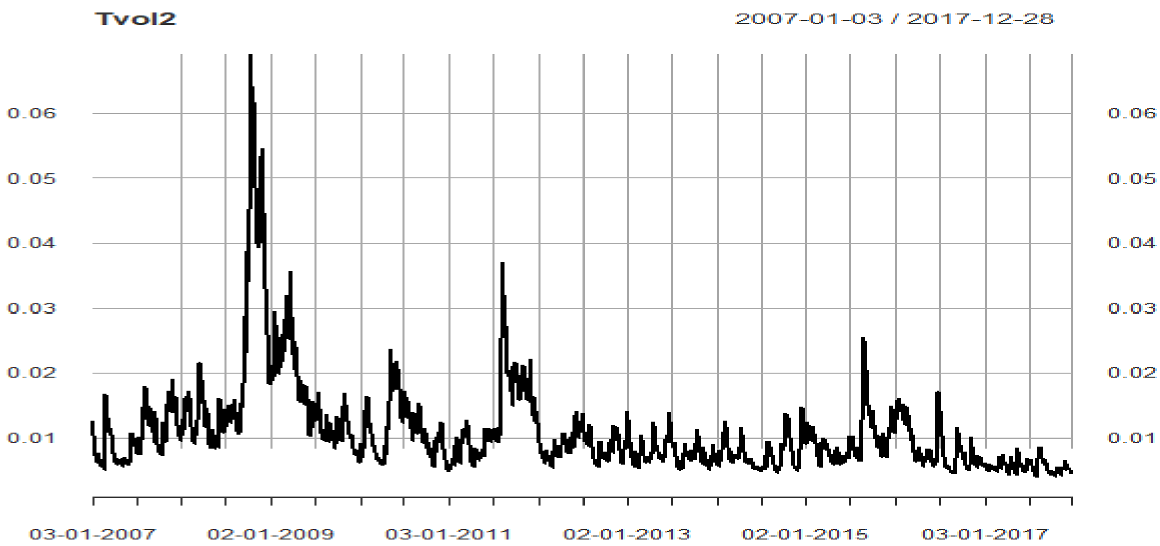
The results meet the requirements. This indicates that our GARCH specification is reasonable.

S&P500

Regarding the asset S&P500, we employ the same approach. We obtain p=0, q=2 and conduct an adequacy test, yielding the following results:

Residuals of the Data	p-value = 0.1336
Squared Residuals of the Data	p-value < 2.2e-16

The results indicate that p and q align with our expectations. Based on the ARMA(0,2) Mean Equation, we proceed to construct the Variance Equation for the GARCH(m,n) model. After searching, we identify the optimal results as m=2 and n=1, resulting in the GARCH(2,1) model. Initially, we perform in-sample forecasting and visualize the conditional volatility using the plot function, resulting in the following graph:



Furthermore, we examine the adequacy of the GARCH fit through residual standard tests (testing if it is a suitable garch). We obtained the following results:

Residuals of the Data	p-value = 0.6309
Squared Residuals of the Data	p-value = 0.6243

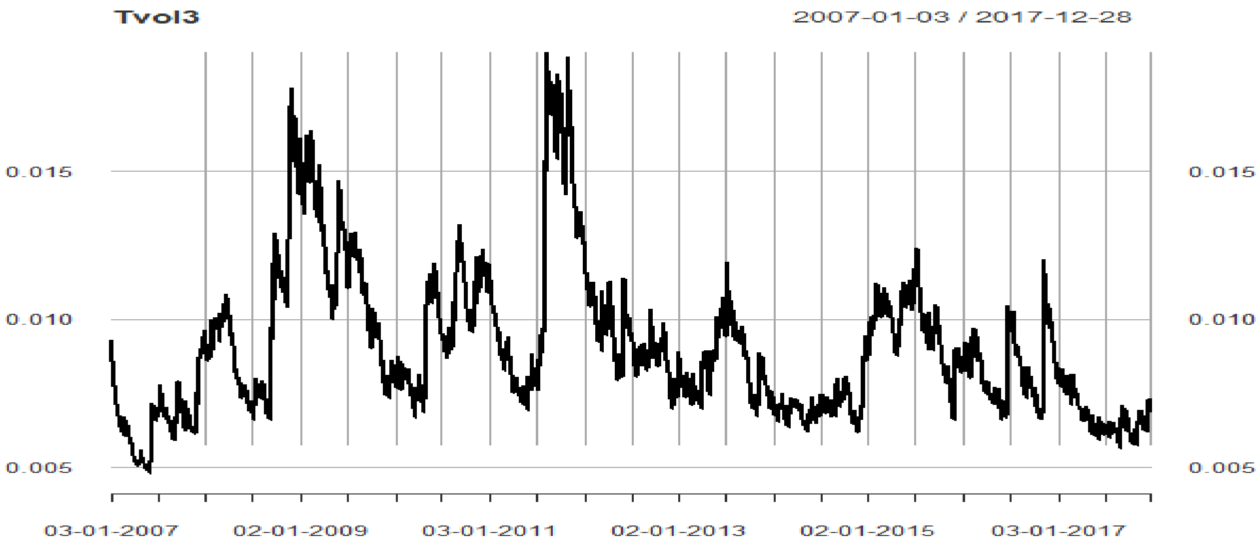
The results meet the requirements, indicating that our GARCH specification is reasonable.

TLT

For the asset TLT, we employ the same methodology. We obtain p=0, q=1 and conduct an adequacy test, yielding the following results:

Residuals of the Data	p-value = 0.3834
Squared Residuals of the Data	p-value < 2.2e-16

The results indicate that p and q are in line with our expectations. Based on the above ARMA (0,1) Mean Equation, we proceed to construct the Variance equation for the GARCH(m,n) model. After thorough search and selection, we determine the optimal result to be m=1, n=1. Thus, GARCH(1,1) emerges as the most favorable outcome! Initially, we undertake in-sample forecasting, creating a time series object and subsequently visualizing the conditional volatility using the plot function. The resulting graph is as follows:



And examine the adequacy of the GARCH fitting through standardized residual test (testing if it is a suitable GARCH). We obtained the following results:

Residuals of the Data	p-value = 0.691
Squared Residuals of the Data	p-value = 0.6594

The results meet the requirements. This indicates that our GARCH specification is reasonable.
The construction of the equal-weight investment portfolio is multiplied by the return on each asset to calculate the past returns of the investment portfolio, with the aim of verifying whether the volatility of the investment portfolio is within our expected range. The function VarTest is used to estimate the maximum potential loss of the investment portfolio at a 99% confidence level. The results are as follows:

Expected Exceedance	10
Actual Exceedance	17

The results are in line with our expectations, indicating that the VaR is appropriate. This suggests that the selected investment portfolio's historical volatility falls within our expected range.

4.2 Copula

Our analysis applies the GARCH-copula procedure to standardized residuals from fitted GARCH models, capturing the risk characteristics of three financial assets. The following tables display the parameter estimates for the marginal distributions and the correlation matrix for the fitted copula. Table 2 presents the estimated parameters for the t-distributions fitted to the standardized residuals of each asset. Table 3 displays the correlation coefficients estimated from the fitted normal copula, capturing the linear dependencies among the assets.

	Mean Estimate	Standard Deviation	Degrees of Freedom	Loglikelihood	AIC
GLD	-0.0428	0.7855	5.17	-3816.548	7639.095
S&P500	-0.0797	0.7723	4.88	-3804.278	7614.556
TLT	-0.0248	0.9195	13.36	-3905.539	7817.078

Table 2: Marginal Distribution Parameter Estimates

	GLD	S&P500	TLT
GLD	1	0.0335	0.1580
S&P500	0.0335	1	-0.4301
TLT	0.1580	-0.4301	1

Table 3: Correlation Matrix of the Fitted t-Copula

The parameter estimation and copula fitting processes were conducted using R programming, leveraging the 'fitdistrplus' and 'copula' packages. The Student's t-distribution provided a more flexible fit for the residuals, accommodating for excess kurtosis observed in financial return distributions. The copula model was then employed to understand the dependency structure, as reflected in the correlation matrix.

The AIC values from the marginal distribution fits and the correlations from the copula model are critical inputs for subsequent risk modeling, including VaR estimation. These results help us to appreciate the complexities of asset dependencies within the portfolio and provide a foundation for robust financial risk management.

4.3 Monte Carlo Simulation and Back Testing

Firstly, let me introduce the generous operation of processing data and using Monte Carlo Simulation.

Step1: Simulation to generate standardized residuals: use the Monte Carlo method to generate standardized residuals for each asset. For this study, we used the data from December 29, 2017 to December 30, 2021(the last trading day of 2017 and 2021) and estimated the VaR in 99% confidence level. We generated 10000 simulation return in this section.

Step2: Reconstruct daily log returns: Reconstruct the daily log returns of the assets by combining the standardized residuals from the simulation and the predictions from the GARCH model.

Step3: Calculate Portfolio Daily Log Return: Calculate the daily log return of the entire portfolio based on the portfolio's weight allocation.

Step4:

- Calculate VaR: Based on the simulated daily log returns of the portfolio, calculate the VaR at a given confidence level. We sorted the M analogue values of each day and took the smallest value (for the smallest 1%, at this time, is based on gain distribution which VaR is negative value)

- Visualization of VaR(Figure 1): The calculated VaR is displayed visually to observe the risk level at different points in time.

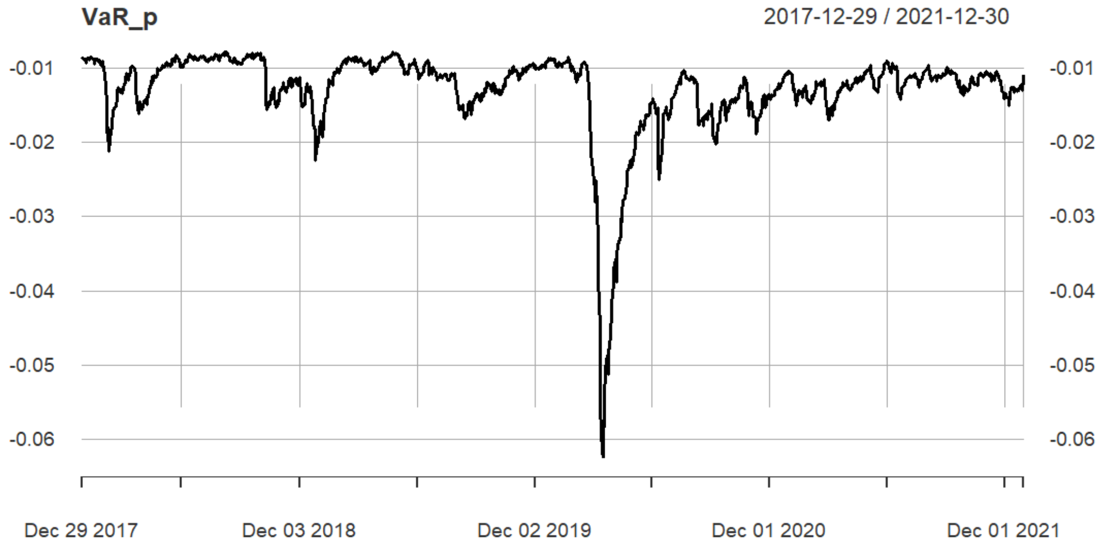


Figure 1

Step5: Backtesting of VaR: The accuracy of the VaR model is tested using historical data to ensure its reliability in practical application. For this part we set two hypotheses and try to validate them:

- h0: the VaR simulation is suitable
- h1: the VaR simulation is NOT suitable

In back-testing of VaR, the sample observed to exceed the VaR estimate is 17 times, which is greater than 10 times.

On the contrary, from a statistical perspective, its p-value is greater than 0.01(0.0461 in unconditional coverage condition and 0.0130 in conditional coverage condition) so H0 is accepted in 99% confidence level.

To answer this question we take it into different views:

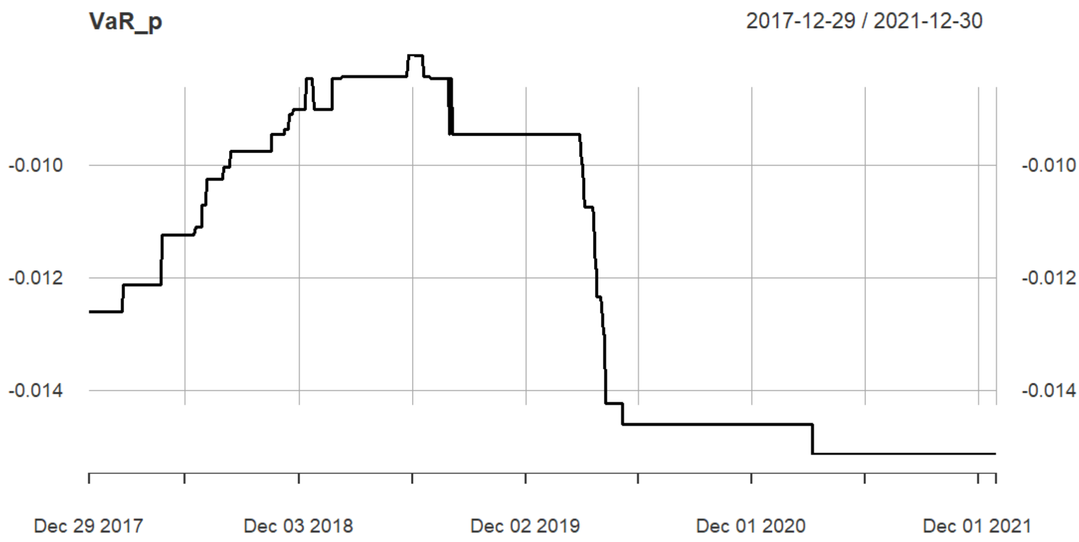
- From a risk management perspective: When looking at risk management, exceeding the expected frequency in reality may warrant attention even if it is not statistically significant. This might imply the need for a reassessment of the risk model or consideration of risk factors that the model may not be capturing.

- Statistical significance: From a statistical significance standpoint, a P-value greater than the significance level suggests that the model may still be appropriate overall. However, this does not rule out the possibility of problems with the model in specific situations.

- Finally, for Comprehensive consideration: Best practice involves taking into account both aspects of information. If the actual frequency continues to exceed expectations and there are other indicators or circumstances suggesting that the risk might be underestimated, adjustments to the model or a more in-depth analysis should be considered, even if statistical tests do not reach the significance level.

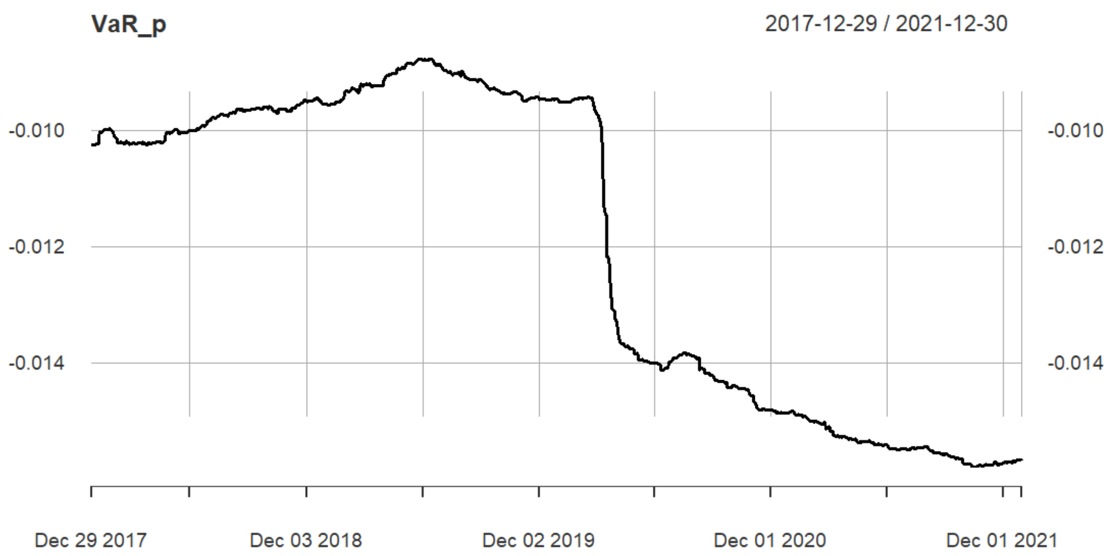
4.4 Historical simulation

We assume that the historical distribution of log returns is a good approximation of the distribution of log returns we face in the next holding period. Based on the quantitative values associated with the historical distribution of log returns, the expected portfolio value-at-risk can be obtained. Using the rolling-window technique, we set the window size to 600. A graph of daily VaR from Rstudio can be obtained.



4.5 Variance-Covariance method

According to this approach, the value at risk of a portfolio can be derived by estimating the variance and covariance of the log returns of a number of predefined risk factors, as well as the sensitivity of the portfolio to these risk factors. Widow size is still set to 600. A graph of daily VaR from Rstudio can be obtained.



5. Conclusion

For model selection, three models were analyzed for GARCH(m,n), and it was concluded that for S&P500 and GLD, the GARCH(2,1) model is used. For TLT, a higher accuracy is achieved with the GARCH(1,1) model.
The Student-t model outperforms the Gaussian model.
Monte Carlo simulation can be statistically utilized with a confidence level of 99 percent.

Regarding image fitting, both the GARCH model and Monte Carlo simulation exhibit better performance compared to traditional models. However, there are some issues that need to be considered in conjunction with practical choices.

Overall, this requires a balanced and comprehensive approach. In risk management, any signs that may indicate underestimation of risk should not be disregarded even if statistical tests do not demonstrate significance

6. Reference

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov & F. Csáki (Eds.), 2nd International Symposium on Information Theory (pp. 267-281). Akadémiai Kiadó. Available at: https://link.springer.com/chapter/10.1007/978-1-4612-1694-0_15 (Accessed on 16th December 2023).

Angelidis, T. and Benos, A., 2005. Value-at-Risk for Greek Stocks. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2623469 (Accessed on 3rd December 2023).

Banerjee, A.K. (2022) 'You sneeze, and the markets are paranoid: the fear, uncertainty and distress sentiments impact of the COVID-19 pandemic on the stock-bond correlation', The Journal of Risk Finance, 23(5), pp. 652–668. doi:10.1108/JRF-04-2022-0095. (Accessed: 8 December 2023).

Baur, D.G. and McDermott, T.K. (2010), "Is gold a safe haven? International evidence", Journal of Banking & Finance, Vol. 34 No. 8, pp. 1886-1898, doi: 10.1016/j.jbankfin.2009.12.008.
Bhatia, V., Das, D. and Kumar, S., 2020. Gold ETFs as a safe haven: An empirical investigation. Resources Policy, [online] 68, 101734. Available at: https://www.sciencedirect.com/science/article/pii/S2214845022000977 (Accessed on Dec 2nd).

Bollerslev, T. (1986) 'Generalized autoregressive conditional heteroskedasticity', Journal of Econometrics, 31(3), pp. 307-327. Available at: https://www.sciencedirect.com/science/article/abs/pii/0304407686900631 (Accessed on 14th December 2023).

Investopediaa (2016) Market Risk. Available at: http://www.investopedia.com/terms/m/marketrisk.asp (Accessed on 2nd December 2023).

Investopediab (2016) How Do You Calculate Value at Risk (VaR) in Excel? Available at: https://www.investopedia.com/ask/answers/033115/how-can-you-calculate-value-risk-var-excel.asp (Accessed on 9th December 2023).

Jones, C.W. (2019) 'European sovereign debt crisis overview'. Salem Press Encyclopedia. Available at: https://search-ebscohost-com.ez.xjtlu.edu.cn/login.aspx?direct=true&db=ers&AN=89158178&site=eds-live&scope=site (Accessed: 8 December 2022).

Lourme, A. and Maurer, F., 2017. Testing the Gaussian and Student's t copulas in a risk management framework. Economic Modelling, 67, pp.203-214. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0264999316308483#preview-section-cited-by (Accessed on 13th December 2023).

Lu, X., Lai, K. and Liang, L., 2014. Portfolio value-at-risk estimation in energy futures markets with time-varying copula-GARCH model. Annals of Operations Research, 219, pp.333-357. Available at: https://www.researchgate.net/publication/251087088_Portfolio_value-at-risk_estimation_in_energy_futures_markets_with_time-varying_copula-GARCH_model (Accessed on 7th December 2023).

Mittal, I., Pradhan, A.K. and Tiwari, A.K. (2021) 'Optimizing the market-risk of major cryptocurrencies using CVaR measure and copula simulation', Macroeconomics and Finance in Emerging Market Economies, 14(3), pp. 291-307–307. doi:10.1080/17520843.2021.1909828.

Mourik, T. (2003) Market risks of insurance companies. Discussion Paper IAA Insurer Solvency Assessment Working Party. Available at: https://actuaries.org/AFIR/Colloquia/Maastricht/Mourik.pdf (Accessed: 8 December 2023).

Patel, K., Pereira, R. and Delurgio, S., 2018. Investor sentiment and S&P 500 index returns. Journal of Behavioral and Experimental Finance, [online] 19, pp.42-56. Available at: https://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1138&context=open_access_theses (Accessed on Dec 9th).

Semantic Scholar, 2021. Value at Risk Estimation A GARCH-EVT-Copula Approach. Semantic Scholar. Available at: https://www.semanticscholar.org/paper/Value-at-Risk-Estimation-A-GARCH-EVT-Copula-Bob/3ad6582bd3a1cd931872a6566c31cf46d0cb36d0 (Accessed on 9th December 2023).

Springer, 2021. Using Conditional Copula to Estimate Value-at-Risk - Springer. SpringerLink. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=818884 (Accessed on 10th December 2023).

Wang, Y. and Lee, J., 2019. The role of treasury bond ETF in financial markets. Finance Research Letters, [online] 30, pp.118-125. Available at: https://www.researchgate.net/publication/228459239_The_role_of_asymmetry_Evidence_from_Chinese_Treasury_bond_market (Accessed on Dec 8th).

Wohlner, R. (2013) Here's Why Diversification Matters. Available at:

http://money.usnews.com/money/blogs/the-smarter-mutual-fund-investor/2013/05/31/heres-why-diversification-matters (Accessed on 2nd December 2023).

7. Appendix

```
# Data preprocessing and portfolio construction
# Read data
gld = read.csv('GLD.csv', header = TRUE)
sp500 = read.csv('SPY.csv', header = TRUE)
tll = read.csv('TLT.csv', header = TRUE)

#Calculate the daily log-return
r_gld = cbind(gld$Date[-1], diff(log(gld$Close)))
r_sp500 = cbind(sp500$Date[-1], diff(log(sp500$Close)))
r_tll = cbind(tll$Date[-1], diff(log(tll$Close)))
# Merge the same trading days
df = merge(r_gld, r_sp500, by='V1', all=FALSE)
df = merge(df, r_tll, by='V1', all=FALSE)
colnames(df) = c('Date', 'r_gld', 'r_sp500', 'r_tll')
df = na.omit(df)# Remove the null value
# Divide the data set 20070103-20171228 is a fitting data set, and 20171228-20211230 is a set of prediction VaR and evaluation of VaR.
timestamp = df[,1]
loc = which(timestamp==20171228)
loc
train_df = df[1:loc,]
test_df = df[(loc+1):dim(df)[1],]
# Get the daily return of two sets
train_return = train_df[,2:4]
test_return = test_df[,2:4]
# Build a portfolio weight
N = 3
# equal weight portfolio
w = rep(1,N)/N
View(w)


#A suitable GARCH model is fitted to the return rate on the train.
#
install.packages('rugarch')
install.packages('forecast')
install.packages('e1071')
library('rugarch')
library('xts')
library('forecast')
library('e1071')
#Convert to date data
train_time = as.Date(as.character(train_df[,1]), '%Y%m%d')
test_time = as.Date(as.character(test_df[,1]), '%Y%m%d')
# ------ Overall statistical description
# The overall sample average, standard deviation, bias and peak of the three stocks
# (Normal distribution bias = 0; peak = 3)
c(mean(df[,2]), sd(df[,2]), skewness(df[,2]), kurtosis(df[,2]))
c(mean(df[,3]), sd(df[,3]), skewness(df[,3]), kurtosis(df[,3]))
c(mean(df[,4]), sd(df[,4]), skewness(df[,4]), kurtosis(df[,4]))
# Normality, if the scattering point does not fall on the straight line, it lacks normality, so T is used when fitting the marginal distribution below.distribution(t_G)
```

```r
qqnorm(df[,2], main = 'gld daily log return')
qqline(df[,2])
qqnorm(df[,3], main = 'sp500 daily log return')
qqline(df[,3])
qqnorm(df[,4], main = 'tll daily log return')
qqline(df[,4])
#Self-correlation graph (used to judge MA) and partial self-correlation graph (used to judge AR) determine whether it is related to past return and past shock.
#PACF judges that the first line of AR exceeds the dotted line, and there must be AR.
#ACFDon't worry about it at 0 degrees. It's equal to 1.
acf(df[,2], main = 'ACF of gld daily log return')
acf(df[,3], main = 'ACF of sp500 daily log return')
acf(df[,4], main = 'ACF of tll daily log return')
pacf(df[,2], main = 'PACF of gld daily log return')
pacf(df[,3], main = 'PACF of sp500 daily log return')
pacf(df[,4], main = 'PACF of tll daily log return')
# ------Modelling
# for the first stock index return
#First, build the Mean Equation of the GARCH model: ARMA(p,q) model
# Use the auto.arima function to automatically search for the most suitable p and q values
fit_meanEq_1 = auto.arima(train_return[,1])
summary(fit_meanEq_1)#arma= (0，0) mean eq :rt =et
# Test mean equation的adequency
Box.test(residuals(fit_meanEq_1), lag = 12, type = 'Ljung-Box')     # The residuals should not be relevant. The p-value should be greater than 0.05
Box.test(residuals(fit_meanEq_1)^2, lag = 12, type = 'Ljung-Box')      # The residual square should be correlated (the existence of ARCH effect will be modelled with the GARCH model) p-value should be less than 0.05
p = fit_meanEq_1$arma[1]
q = fit_meanEq_1$arma[2]
p
q

# The Mean Equation based on ARMA(p,q) above continues to build the Variance equation: GARCH(m,n) model, in order to eliminate the residual ARCH effect
# Use the AIC minimum grid to search for the appropriate m and n
max_lag = 5
AIC_metrics = matrix(rep(0, max_lag^2), max_lag, max_lag)
for (i in 1:max_lag){
  for (j in 1:max_lag){
    GARCHij = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(i,j)), mean.model = list(armaOrder = c(p,q), include.mean = FALSE), distribution.model = 'std')
    GARCHij_fit = ugarchfit(GARCHij, train_return[,1])
    AIC_metrics[i,j] = infocriteria(GARCHij_fit)['Akaike',]
  }
}
m = which(AIC_metrics == min(AIC_metrics), arr.ind = TRUE)[1]
n = which(AIC_metrics == min(AIC_metrics), arr.ind = TRUE)[2]

AIC_metrics
m
n
# Build the best mean equation and variety equation based on the above p, q and m, n
GARCH1 = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(m,n)), mean.model = list(armaOrder = c(p,q), include.mean = FALSE), distribution.model = 'std')
GARCH1_fit = ugarchfit(GARCH1, train_return[,1])
GARCH1_fit
# return fitted(This is the prediction in the sample.)
return_fitted_1 = fitted(GARCH1_fit)
# conditional volatility fitted
ConVolatility_1 = sigma(GARCH1_fit)
# residuals
residuals_1 = residuals(GARCH1_fit)
# standardized residuals
standRes_1 = residuals_1/ConVolatility_1
# model assessment
# plot the conditional volatility
Tvol1 = xts(ConVolatility_1, train_time)
plot(Tvol1)
plot(Tvol1, format.labels = '%d-%m-%Y')
# Test the adequency (standard residual test) after being fitted by GARCH
Box.test(standRes_1, lag = 12, type = 'Ljung-Box')        # The standard residual should not have a correlation p-value should be greater than 0.05
Box.test(standRes_1^2, lag = 12, type = 'Ljung-Box')       # The standard residual square should not have a correlation p-value greater than 0.05 (ARCH effect eliminated)
# Predict the prediction set
garchroll_1 = ugarchroll(GARCH1, data = df[,2] , n.start = loc, refit.window = 'moving',  refit.every = 10)
preds_1 = as.data.frame(garchroll_1)#Turn the prediction into a data framework.
preds_1_mu = as.xts(preds_1$Mu, test_time)
preds_1_sigma = as.xts(preds_1$Sigma, test_time)
# The following are the repeated steps to carry out the same modelling process for the rest of the stocks.
# for the second stock index return
# First build the Mean Equation of the GARCH model: ARMA(p,q) model
# Use the auto.arima function to automatically search for the most suitable p and q values.
fit_meanEq_2 = auto.arima(train_return[,2])
summary(fit_meanEq_2)
p = fit_meanEq_2$arma[1]
q = fit_meanEq_2$arma[2]
p
q

Box.test(residuals(fit_meanEq_2), lag = 12, type = 'Ljung-Box')
Box.test(residuals(fit_meanEq_2)^2, lag = 12, type = 'Ljung-Box')
```

```r

max_lag = 5
AIC_metrics = matrix(rep(0, max_lag^2), max_lag, max_lag)
for (i in 1:max_lag){
  for (j in 1:max_lag){
    GARCHij = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(i,j)), mean.model = list(armaOrder = c(p,q), include.mean = FALSE), distribution.model = 'std')
    GARCHij_fit = ugarchfit(GARCHij, train_return[,2])
    AIC_metrics[i,j] = infocriteria(GARCHij_fit)['Akaike',]
  }
}
m = which(AIC_metrics == min(AIC_metrics), arr.ind = TRUE)[1]
n = which(AIC_metrics == min(AIC_metrics), arr.ind = TRUE)[2]
m
n
AIC

GARCH2 = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(m,n)), mean.model = list(armaOrder = c(p,q), include.mean = FALSE), distribution.model = 'std')
GARCH2_fit = ugarchfit(GARCH2, train_return[,2])
GARCH2_fit

# return fitted
return_fitted_2 = fitted(GARCH2_fit)
# conditional volatility fitted
ConVolatility_2 = sigma(GARCH2_fit)
# residuals
residuals_2 = residuals(GARCH2_fit)
# standardized residuals
standRes_2 = residuals_2/ConVolatility_2

# model assessment
# plot the conditional volatility
Tvol2 = xts(ConVolatility_2, train_time)
plot(Tvol2)
plot(Tvol2, format.labels = '%d-%m-%Y')

Box.test(standRes_2, lag = 12, type = 'Ljung-Box')          # The standard residual should not have correlation p-value should be greater than 0.05
Box.test(standRes_2^2, lag = 12, type = 'Ljung-Box')        # The standard residual square should not have correlation p-value should be greater than 0.05 (ARCH effect eliminated)

# Predict the prediction setgarchroll_2 = ugarchroll(GARCH2, data = df[,3] , n.start = loc, refit.window = 'moving',  refit.every = 10)
preds_2 = as.data.frame(garchroll_2)
preds_2_mu = as.xts(preds_2$Mu, test_time)
preds_2_sigma = as.xts(preds_2$Sigma, test_time)



# for the third stock index return

fit_meanEq_3 = auto.arima(train_return[,3])
summary(fit_meanEq_3)
p = fit_meanEq_3$arma[1]
q = fit_meanEq_3$arma[2]
p
q


Box.test(residuals(fit_meanEq_3), lag = 12, type = 'Ljung-Box')
Box.test(residuals(fit_meanEq_3)^2, lag = 12, type = 'Ljung-Box')


max_lag = 5
AIC_metrics = matrix(rep(0, max_lag^2), max_lag, max_lag)
for (i in 1:max_lag){
  for (j in 1:max_lag){
    GARCHij = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(i,j)), mean.model = list(armaOrder = c(p,q), include.mean = FALSE), distribution.model = 'std')
    GARCHij_fit = ugarchfit(GARCHij, train_return[,3])
    AIC_metrics[i,j] = infocriteria(GARCHij_fit)['Akaike',]
  }
}
m = which(AIC_metrics == min(AIC_metrics), arr.ind = TRUE)[1]
n = which(AIC_metrics == min(AIC_metrics), arr.ind = TRUE)[2]
m
n
AIC

```

```
GARCH3 = ugarchspec(variance.model = list(model = 'sGARCH', garchOrder = c(m,n)), mean.model = list(armaOrder = c(p
,q), include.mean = FALSE), distribution.model = 'std')
GARCH3_fit = ugarchfit(GARCH3, train_return[,3])
GARCH3_fit

# return fitted
return_fitted_3 = fitted(GARCH3_fit)
# conditional volatility fitted
ConVolatility_3 = sigma(GARCH3_fit)
# residuals
residuals_3 = residuals(GARCH3_fit)
# standardized residuals
standRes_3 = residuals_3/ConVolatility_3

# model assessment
# plot the conditional volatility
Tvol3 = xts(ConVolatility_3, train_time)
plot(Tvol3)
plot(Tvol3, format.labels = '%d-%m-%Y')
# Test the adequency (standard residual test) after GARCH fitting
Box.test(standRes_3, lag = 12, type = 'Ljung-Box')          # The standard residual should not have correlation p-
value should be greater than 0.05
Box.test(standRes_3^2, lag = 12, type = 'Ljung-
Box')     # The standard residual square should not have correlation p-
value should be greater than 0.05 (ARCH effect eliminated)

# Predict the prediction set
garchroll_3 = ugarchroll(GARCH3, data = df[,4] , n.start = loc, refit.window = 'moving',  refit.every = 10)
preds_3 = as.data.frame(garchroll_3)
preds_3_mu = as.xts(preds_3$Mu, test_time)
preds_3_sigma = as.xts(preds_3$Sigma, test_time)


#Because the non-
normal distribution is judged at the beginning, it is more appropriate to use std or sstd instead of norm in garch.

#--------------------------------------------------------
# Fit the copula function
#
install.packages('fitdistrplus')
install.packages('copula')
library('fitdistrplus')
library('copula')

standRes_1 = as.vector(standRes_1)
standRes_2 = as.vector(standRes_2)
standRes_3 = as.vector(standRes_3)

#The code Generalised t-distributed p, d, q functions appear in the courseware
dt_G = function(x, mean, sd, nu){#density
  dt((x-mean)/sd,nu)/sd
}
pt_G = function(q, mean, sd, nu){#CDF
  pt((q-mean)/sd,nu)
}
qt_G = function(x, mean, sd, nu){#quantile
  qt(x,nu)*sd+mean
}

# The standard residuals fitted by the above three GARCH models are fitted to their marginal distribution.
fit_r1 = fitdist(standRes_1, 't_G', start = list(mean = mean(standRes_1), sd = sd(standRes_1), nu = 5))
summary(ft_r1)
fit_r2 = fitdist(standRes_2, 't_G', start = list(mean = mean(standRes_2), sd = sd(standRes_2), nu = 5))
summary(ft_r2)
fit_r3 = fitdist(standRes_3, 't_G', start = list(mean = mean(standRes_3), sd = sd(standRes_3), nu = 5))
summary(ft_r3)

# Build a normal copula
u = matrix(nrow = length(standRes_1), ncol = N)
u[,1] = pt_G(standRes_1, mean = as.list(fit_r1$estimate)$mean, sd = as.list(fit_r1$estimate)$sd, nu = as.list(fit_r
1$estimate)$nu)
u[,2] = pt_G(standRes_2, mean = as.list(fit_r2$estimate)$mean, sd = as.list(fit_r2$estimate)$sd, nu = as.list(fit_r
2$estimate)$nu)
u[,3] = pt_G(standRes_3, mean = as.list(fit_r3$estimate)$mean, sd = as.list(fit_r3$estimate)$sd, nu = as.list(fit_r
3$estimate)$nu)
norm.cop = normalCopula(dim = N, dispstr = 'un')
n.cop = fitCopula(norm.cop, u, method = 'ml')
coef(n.cop)

#Obtain the parameters of three marginal distributions
mean_r1 = as.list(fit_r1$estimate)$mean
sd_r1 = as.list(fit_r1$estimate)$sd
nu_r1 = as.list(fit_r1$estimate)$nu

mean_r2 = as.list(fit_r2$estimate)$mean
sd_r2 = as.list(fit_r2$estimate)$sd
nu_r2 = as.list(fit_r2$estimate)$nu

mean_r3 = as.list(fit_r3$estimate)$mean
sd_r3 = as.list(fit_r3$estimate)$sd
nu_r3 = as.list(fit_r3$estimate)$nu
```

```
#-------------------------------------------------------
#
# Calculate VaR
#

# create the correlation matrix using the fitted normal copula
cor = matrix(data=c(1,coef(n.cop)[1],coef(n.cop)[2],coef(n.cop)[1],1,coef(n.cop)[3],coef(n.cop)[2],coef(n.cop)
[3],1),nrow=3,ncol=3)
cor
# Monte Carlo Simulation
L = t(chol(cor))#Decompose cor
set.seed(1234)#Set a random number of seeds
M = 10000
T = length(test_time)
Sim_R1=matrix(nrow=M,ncol=T)
Sim_R2=matrix(nrow=M,ncol=T)
Sim_R3=matrix(nrow=M,ncol=T)
for (i in 1:M){
  z=rnorm(N*T)
  z=matrix(z,N,T)
  z_tilde=L%*%z
  R1=q_tG(pnorm(z_tilde[1,]),mean=mean_r1, sd=sd_r1, nu=nu_r1)
  R2=q_tG(pnorm(z_tilde[2,]),mean=mean_r2, sd=sd_r2, nu=nu_r2)
  R3=q_tG(pnorm(z_tilde[3,]),mean=mean_r3, sd=sd_r3, nu=nu_r3)
  Sim_R1[i,] = R1
  Sim_R2[i,] = R2
  Sim_R3[i,] = R3
}
# Fusion simulation value (simulated is standardlised residuals, VaR is the calculation of return) and prediction v
alue
return_reintroduce_the_heteroscedasticity = function(pred_model, sigma_model, Simulated_returns){
  #sigma_matrix = coredata(sigma_model)
  diagonal_sigma = diag(as.numeric(sigma_model), nrow = length(sigma_model))
  simulated_residuals = diagonal_sigma%*%t(Simulated_returns)
  simulated_log_returns = simulated_residuals + matrix(rep(as.numeric(pred_model), ncol(simulated_residuals)), ncol
 = ncol(simulated_residuals))
  simulated_log_returns
}

return_1 = return_reintroduce_the_heteroscedasticity(preds_1_mu, preds_1_sigma, Sim_R1)
return_2 = return_reintroduce_the_heteroscedasticity(preds_2_mu, preds_2_sigma, Sim_R2)
return_3 = return_reintroduce_the_heteroscedasticity(preds_3_mu, preds_3_sigma, Sim_R3)

# Calculate the simulation value of the log return of the portfolio according to the selected portfolio
portfolio_return = w[1]*return_1+w[2]*return_2+w[3]*return_3
alpha = 0.01 # confidence level
VaR_p = rep(NA, T)
for (j in 1:T){
  # Sort the M analogue values of each day
  portfolio_return_sorted = sort(portfolio_return[j,])
# Take the smallest first a value (eg. The smallest 1%, at this time, is based on gain distribution, VaR is negativ
e)
  VaR_p[j] = portfolio_return_sorted[floor(M*alpha)]
}
# Draw daily VaR
VaR_p = as.xts(VaR_p, test_time)
plot(VaR_p)
#-------------------------------------------------------
#
# Back-Test
# H0:VaR is suitable
# H1:VaR is not suitable
return_pf = w[1]*test_return[,1]+w[2]*test_return[,2]+w[3]*test_return[,3]
VaRTest(alpha, return_pf, VaR_p)


#-------------------------------------------------------
#
# historical simulation
#

VaR_p = rep(NA, T)
windowSize = 600
for (i in loc:dim(df)[1]-1){
  startloc = i-windowSize+1
  endloc = i
  selected_df = df[startloc:endloc,]
  gain = w[1]*selected_df[,2] + w[2]*selected_df[,3] + w[3]*selected_df[,4]
  VaR_p[i-loc+1] = sort(gain)[floor(windowSize*alpha)]
}

#Draw daily VaR
VaR_p = as.xts(VaR_p, test_time)
plot(VaR_p)

# Back Test
VaRTest(alpha, return_pf, VaR_p)
backTest(alpha, return_pf, VaR_p)

```

```r
#-------------------------------------------------------
#
# Variance-Covariance Method
#

VaR_p = rep(NA, T)
windowSize = 600
for (i in loc:dim(df)[1]-1){
  startloc = i-windowSize+1
  endloc = i
  selected_df = df[startloc:endloc,]
  varcov = matrix(as.numeric(cov(selected_df[,2:4])), N, N)
  weight = as.vector(as.numeric(w))
  port_sd = sqrt(t(weight)%*%varcov%*%weight)
  VaR_p[i-loc+1] = qnorm(alpha)*port_sd
}

# Draw daily VaR
VaR_p = as.xts(VaR_p, test_time)
plot(VaR_p)

# Back Test
VaRTest(alpha, return_pf, VaR_p)
backTest(alpha, return_pf, VaR_p)
```