

STAT0011-38: Decision and Risk In-Course Assessment (ICA) 2023

Group 7

2023-03-17

Individual Contributions

The contributions below only lists what each individual is in charge of. We confirm that during the project, each individual had participated in all tasks.

19021499

- Searched and processed data in Task 2a
- Implemented parametric approach in Task 2a

19184272

- Discussed strengths and weaknesses of parametric approach in Task 2a
- Compared strengths and weaknesses of parametric and MC simulation in Task 2c

20019013

- Discussed strengths and weaknesses of MC simulation approach in Task 2b
- Compared strengths and weaknesses of parametric and MC simulation in Task 2c

20019563

- Implemented simulations in Task 2b
- Constructed tables and organised report contents

20129199

- Developed question and solutions in Task 1
- Developed a marking scheme in Task 1

18000274

- Converted question and solutions to electronic document in Task 1

Task 1

Task 2

Task 2a

The code and solution using parametric method is shown below. Here we assume that the distribution of the log-returns follows a normal distribution and have homogeneous variance.

```
# Load Data:
data <- read.csv("SP500_NK225.csv")
SP500_close <- data$SP500.price
NK225_close <- data$NK225.price

# Calculate the log-returns for the Nikkei and S&P indices:
SP500_log_ret <- diff(log(SP500_close))
NK225_log_ret <- diff(log(NK225_close))

# Calculate variance-covariance matrix of log-returns for
# the Nikkei and S&P indices:
log_ret_cov <- cov(data.frame(SP500_log_ret, NK225_log_ret))

# Create vector of portfolio weights, since it's equal
# weight it would be (0.5, 0.5)
weight <- matrix(c(0.5, 0.5), nrow = 1)

# Calculate portfolio variance
portfolio_var <- weight %*% log_ret_cov %*% t(weight)

# Create vector of expected log-returns for each stock
mean <- matrix(c(mean(SP500_log_ret), mean(NK225_log_ret)), nrow = 1)

# Calculate portfolio expected log-return
portfolio_mean <- weight %*% t(mean)

# Calculating the 95% AND 99% VaR:
VaR_95 <- qnorm(0.95) * sqrt(portfolio_var) - portfolio_mean
VaR_99 <- qnorm(0.99) * sqrt(portfolio_var) - portfolio_mean
```

Table 1: Parametric Value-at-Risk

99%	95%
0.055869	0.0394168

Strength of Parametric Method

1. **Simple and easy to implement:** requires minimal assumptions and is easy to implement compared to approaches such as Monte Carlo simulation.
2. **Provides precise estimates:** provides deterministic solution of estimated VaR, which is particularly useful for risk management purposes as it provides a fixed number estimate.
3. **Computing efficiency:** only algebraic calculation is required, making the algorithm a lot faster than doing simulations using MC method, which reduces computational costs.

Weakness of Parametric Method

1. **Assumes normal distribution:** log-returns of financial assets are often not exactly normally distributed, making this assumption invalid and causing inaccurate estimation.
2. **Sensitive to model specification:** accuracy of VaR estimates is sensitive to model specification, such as the choice of distribution and the parameter estimation method.
3. **Ignores complex dependency structure:** only consider correlation between assets' returns so cannot effectively discover more complex dependencies between assets, the tail dependency (especially when describing cases of extreme losses) is not considered, may lead to estimate that fails to explain dynamics during market crisis.
4. **Ignores variance clustering:** assumes constant variance (homoscedasticity) for log-returns. However, financial data often exhibit variance clustering, which may make the estimate inaccurate.

Task 2b

2b.1 Data Input

Among the six indices provided, we selected **S&P500** and **Nikkei 225** considering the most appropriate choices to construct a portfolio. To simplify and make the naming more organised, we will name the two indices as “SP500” and “NK225”, respectively.

We have selected S&P500 and Nikkei 225 to construct our portfolio. The stock price data, due to availability, are downloaded from both [Yahoo Finance] and [investing.com]. The price for both stocks are stored to a single .csv file named “SP500_NK225.csv”.

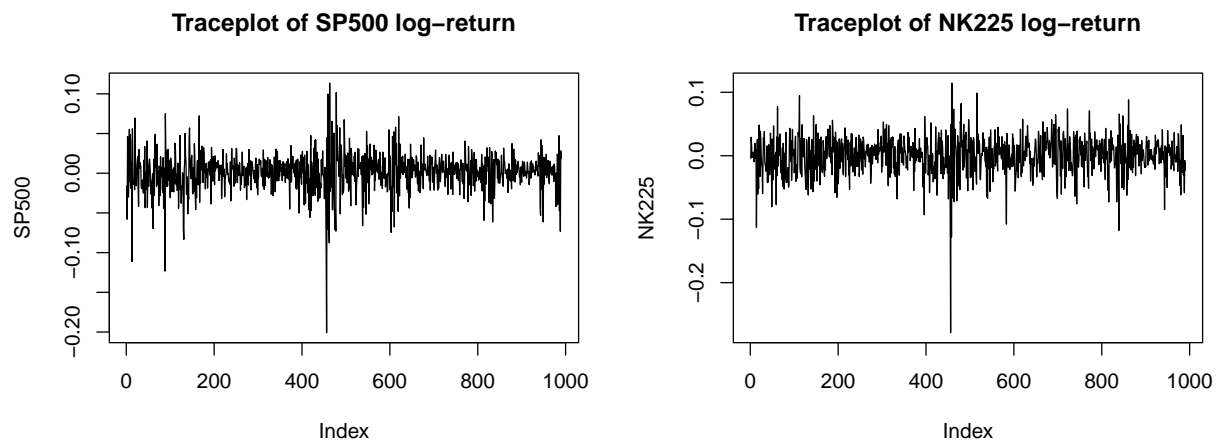
```
# Read-in csv data of stock prices
data <- read.csv("SP500_NK225.csv")
SP500.price <- data$SP500.price
NK225.price <- data$NK225.price

# Calculate log-return (notice here the variables SP500 and
# NK225 without subscript represent log-returns)
SP500 <- diff(log(SP500.price))
NK225 <- diff(log(NK225.price))

# Remove temporary variables
rm(data, SP500.price, NK225.price)
```

2b.2 Time Series Modelling

Taking a look at the plots of the log-returns for the 2 indices, we can observe that log-returns evolve over time, with non-constant variances which comes in cluster. Therefore, models for the time-varying conditional mean and variance will be considered and tested in the following sections.



Firstly, we check for the existence of autocorrelation for the log-returns using the Ljung-Box test. The null hypothesis is that the first m lags autocorrelation coefficients are jointly zero, and we reject H_0 for S&P500 index due to a small p-value and do not reject it for Nikkei 225 index due to a p-value of 0.7646. Therefore, we expect a serial correlation in the conditional mean only for the log-return of S&P500 but not for the latter.

As for the conditional variance, the Ljung-box test for squared log-returns are used, resulting in significant p-values for both stocks, which indicates the presence of volatility clusters, as shown in the traceplots mentioned above.

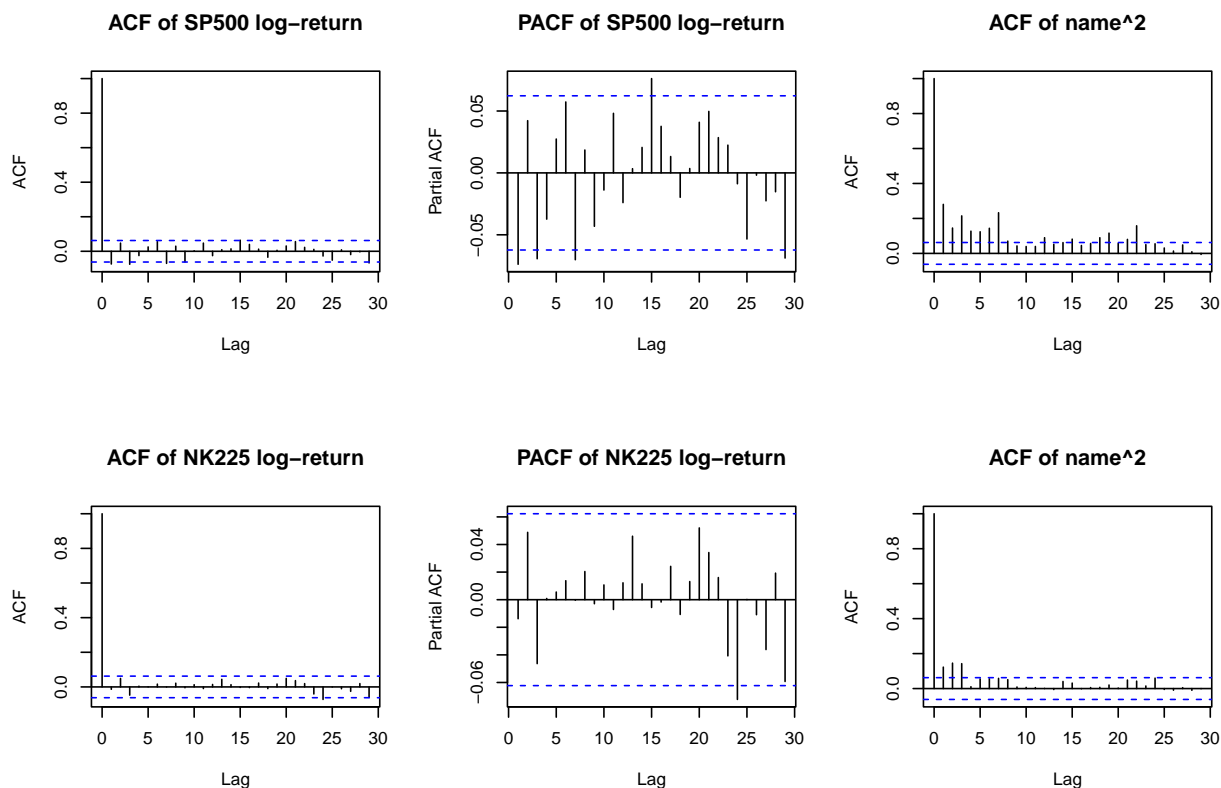
```
# Check for autocorrelation via Ljung-Box test
SP500.Box1 <- Box.test(SP500, lag = 10, type = c("Ljung-Box"),
  fitdf = 1)$p.value # Significant
SP500.Box2 <- Box.test(NK225, lag = 10, type = c("Ljung-Box"),
  fitdf = 1)$p.value # Passed

# Check for volatility clustering via Ljung-Box test
NK225.Box1 <- Box.test(SP500^2, lag = 10, type = c("Ljung-Box"),
  fitdf = 1)$p.value # Significant
NK225.Box2 <- Box.test(NK225^2, lag = 10, type = c("Ljung-Box"),
  fitdf = 1)$p.value # Significant
```

Table 2: p-values for Ljung-Box Tests on Log-Returns

	SP500	NK225
Box.test (Log-return)	0.001	0.765
Box.test (Log-return Squared)	0.000	0.000

The revealed autocorrelation and volatility clustering can also be seen from the bellowed ACF and PACF plots.



From the above ACF-PACF plots, we can conclude that for S&P500:

1. The ACF plot has values which are almost all within the 95% confidence interval and statistically close to zero;
2. The PACF cuts off to zero after the first lag, and lags 3, 7, 15, 29 are all significant (we will choose lag as 1, this will later be proved to be sufficient in addressing SP500 log-returns' autocorrelation, and will help us avoid potential over-fitting problem.);
3. The ACF of squared return witness a quick geometrical decay and there are some significant lags which indicate a need for a GARCH model, which is consistent with the Ljung-box test results.

And for Nikkei 225:

1. Similar ACF plots are presented;
2. Only lag 24 is significant in the PACF plot, supporting the previous conclusion that the log-return of Nikkei 225 index does not have obvious autocorrelation.
3. Similar ACF of squared return as S&P500, showing some significance in the first few lags, and the GARCH model might be used.

Therefore, we only consider building AR(1)-GARCH(1,1) model for S&P500 to account for its serial correlation, while a single GARCH model for Nikkei 225. From the lecture notes we can know that a GARCH(1,1) model is enough for addressing the volatility for stock's log-returns, so we will apply the technique here.

We have selected the conditional distributions for both models to be *Skewed Standardised t Distribution* (sstd), represented by the argument `cond.dist = "sstd"`. According to our findings, the time series modelling of the two stocks' log-return under sstd are able to pass the Ljung-Box test on model residuals' autocorrelation, and are able to give less-significant results on further Kolmogorov-Smirnov Goodness-Of-Fit Test and Anderson-Darling Goodness-of-Fit test after the probability integral transformation on residuals.

The detailed analysis we performed on finding this optimal distribution can be found under Appendix A.

```
# SP500 AR(1)-GARCH(1,1) model
SP500.model <- garchFit(formula = ~arma(1, 0) + garch(1, 1),
  data = SP500, trace = FALSE, cond.dist = "sstd")
SP500.res <- residuals(SP500.model, standardize = TRUE)
SP500.coef <- SP500.model@fit$coef

# NK225 GARCH(1,1) model
NK225.model <- garchFit(formula = ~garch(1, 1), data = NK225,
  trace = FALSE, cond.dist = "sstd")
NK225.res <- residuals(NK225.model, standardize = TRUE)
NK225.coef <- NK225.model@fit$coef
```

Table 3: ICs for Selected Time-Series Models

	SP500	NK225
AIC	-4.970	-4.317
BIC	-4.935	-4.287
SIC	-4.970	-4.317
HQIC	-4.956	-4.306

The information criteria above are all reasonably low in their values, causing no further concerns.

To check the fitting of our model, we shall inspect if there is autocorrelation or volatility clustering left for the model residuals. This can be checked again using Ljung-Box test and ACF-PACF plots, as shown below.

```

# Check SP500 modelling
SP500.Box1 <- Box.test(SP500.res, lag = 10, type = c("Ljung-Box"),
  fitdf = 1)$p.value # Passed
SP500.Box2 <- Box.test(SP500.res^2, lag = 10, type = c("Ljung-Box"),
  fitdf = 1)$p.value # Passed

# Check NK225 modelling
NK225.Box1 <- Box.test(NK225.res, lag = 10, type = c("Ljung-Box"),
  fitdf = 1)$p.value # Passed
NK225.Box2 <- Box.test(NK225.res^2, lag = 10, type = c("Ljung-Box"),
  fitdf = 1)$p.value # Passed

```

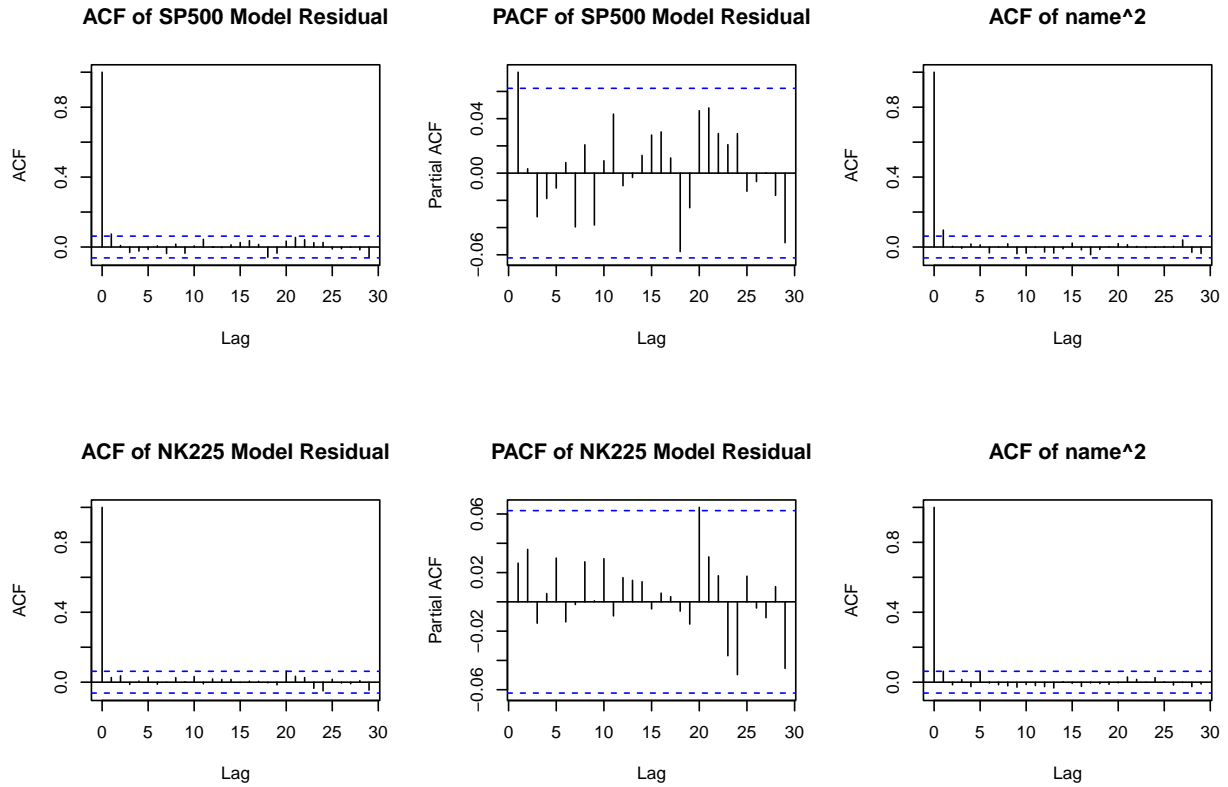


Table 4: p-values for Ljung-Box Tests

	SP500	NK225
Box.test (Residuals)	0.333	0.126
Box.test (Residuals Squared)	0.844	0.368

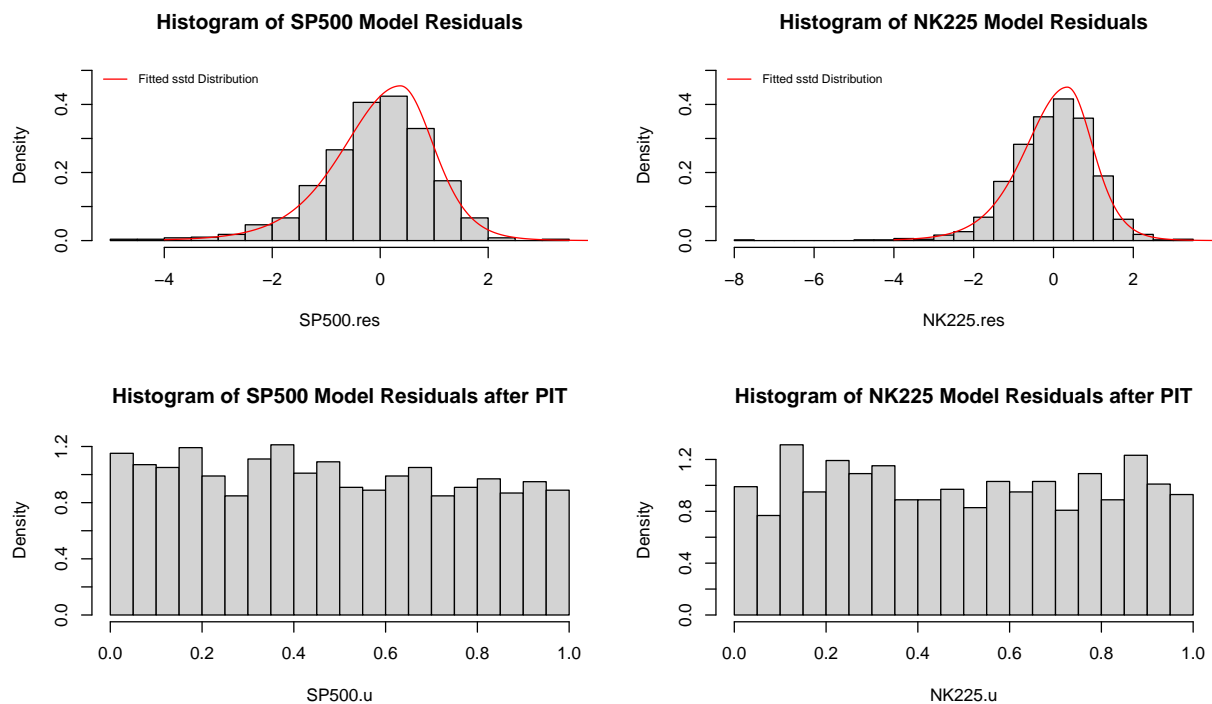
It turns out that both the p-values for residuals and squared residuals are larger than 0.05, indicating that they are independent, and therefore both the AR model and GARCH model are well chosen. This is also consistent with the performance of the plots, with almost all the ACFs and PACFs within the 95% confidence interval, as shown in figure below.

2b.3 Constructing Copula

Now we consider fitting copula models and doing the simulations.

We firstly determine the marginal distributions as well as their parameters. In order to combine the time series model with the copula theory, the following distribution used for probability integral transformation (PIT) should be the same as the conditional distribution of the AR-GARCH models specified earlier, i.e. we approximate the residuals to have a *skewed standardized t distribution* with the skewness and shape parameters set to be the previous `garchFit` modelling results.

```
# Perform PIT
SP500.u <- psstd(SP500.res, nu = SP500.coef["shape"], xi = SP500.coef["skew"])
NK225.u <- psstd(NK225.res, nu = NK225.coef["shape"], xi = NK225.coef["skew"])
```



Through the histograms, density lines generated from the chosen distributions fit considerably well to the distribution of residuals, and the distribution of value after PIT are shown to be fairly uniform. We can check this more formally using Kolmogorov-Smirnov and Anderson-Darling goodness-of-fit tests.

```
# Checking PIT effectiveness
SP500.LcKS <- LcKS(SP500.u, cdf = "punif")$p.value
SP500.ad.test <- as.numeric(ad.test(SP500.u, null = "punif")$p.value)

NK225.LcKS <- LcKS(NK225.u, cdf = "punif")$p.value
NK225.ad.test <- as.numeric(ad.test(NK225.u, null = "punif")$p.value)
```

Table 5: p-values for Goodness-of-Fit Tests

	SP500	NK225
LcKS	0.047	0.051
ad.test	0.418	0.726

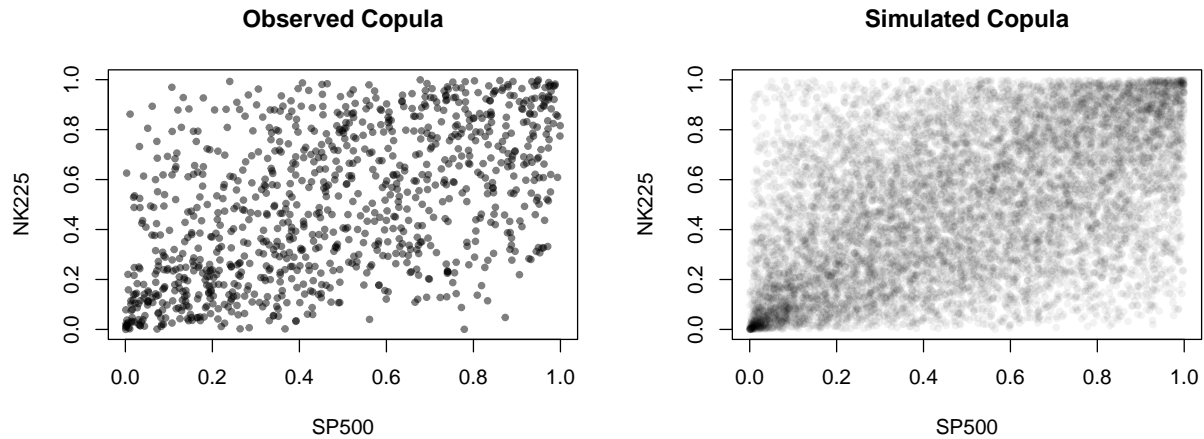
As we see from the test results.

Although one of the test results of uniformity after PIT is significant, this is the best result we may obtain after trying out supported distributions under the `garchFit` function (see Appendix A). However, considering that the resulted p-values are not extremely small, and that the above histograms shows satisfying results for PIT, we believe this approximation is acceptable.

We would now use the transformed data to find a suitable copula.

```
# Determine copula
model <- BiCopSelect(SP500.u, NK225.u, familyset = NA, selectioncrit = "AIC",
  indeptest = TRUE, level = 0.05)
# model Bivariate copula: Survival BB1 (par = 0.11, par2 =
# 1.45, tau = 0.35)

# Simulate from copula - Survival BB1 (BB1 90 degree
# rotate)
n <- 10000
sim.u <- BiCopSim(n, family = 17, par = model$par, par2 = model$par2)
```



From the plot of observed data, we see there is a fairly obvious trace of lower tail decency structure, and slight upper dependency We would expect to capture this characteristic using a suitable copula.

The result of `BiCopSelect` suggest us to use a *Survival BB1* (BB1 with 90 degree rotation) copula based on optimal AIC criterion. We will make 10^4 simulations using `BiCopSim`. The result is displayed as the plot on the right. We see that the simulated copula indeed captures the characteristics we identified earlier, which shows the simulation to be rather successful.

We will now breakdown the simulated data to our two stocks and apply inverse PIT to prepare for further procedures.

```
# Separate simulated results for SP500 and NK225
SP500.sim <- qstd(sim.u[, 1], nu = SP500.coef["shape"], xi = SP500.coef["skew"])
NK225.sim <- qstd(sim.u[, 2], nu = NK225.coef["shape"], xi = NK225.coef["skew"])
```

2b.4 Re-introducing Time Series Effects

The final step before calculating the values of VaR is to re-introduce the autocorrelation and GARCH effects of SP500 and NK225 indices into the simulation results. This can be done using the code below. We will use the end of the observed data to initiate our re-introducing procedure.

```
# Re-introduce AR-GARCH for SP500
mu <- SP500.coef["mu"]
ar1 <- SP500.coef["ar1"]
omega <- SP500.coef["omega"]
alpha1 <- SP500.coef["alpha1"]
beta1 <- SP500.coef["beta1"]

sigma2 <- omega + alpha1 * tail(SP500.model@residuals, 1)^2 +
  beta1 * tail(SP500.model@sigma.t, 1)^2
SP500.out <- mu + ar1 * tail(SP500, 1) + sqrt(sigma2) * SP500.sim

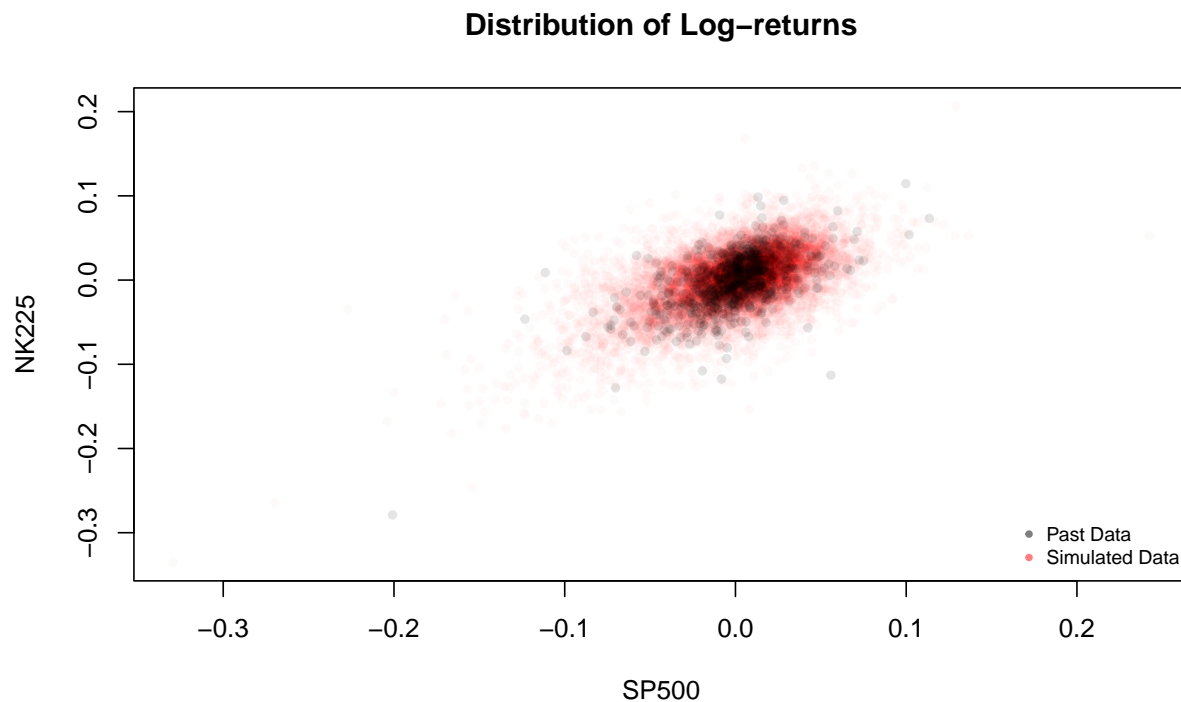
rm(mu, ar1, omega, alpha1, beta1, sigma2)

# Re-introduce GARCH for NK225
mu <- NK225.coef["mu"]
omega <- NK225.coef["omega"]
alpha1 <- NK225.coef["alpha1"]
beta1 <- NK225.coef["beta1"]

sigma2 <- omega + alpha1 * tail(SP500.model@residuals, 1)^2 +
  beta1 * tail(SP500.model@sigma.t, 1)^2
NK225.out <- mu + sqrt(sigma2) * NK225.sim

# Remove temporary variables
rm(mu, omega, alpha1, beta1, sigma2)
```

As shown from the scatter plot of log-returns. We can see a good overlap of real data and simulated data, another indicator that our models are working considerably well.



2b.5 Compute Value-at-Risk

We now calculate the 99% and 95% VaR of the portfolio on the weekly basis using the point we obtained from our Monte Carlo copula simulation. The results is shown below (for clarity, `varsim` stores negative values representing VaR, so when rendering the table below we took its absolute value):

```
portsim <- matrix(0, nrow = n, ncol = 1)
varsim <- matrix(0, nrow = 1, ncol = 2)

portsim <- log(1 + ((exp(SP500.out) - 1) + (exp(NK225.out) -
1)) * (1/2))
varsim <- quantile(portsim, c(0.01, 0.05))
```

Table 6: Monte Carlo Value-at-Risk

99% VaR	95% VaR
0.09383	0.0548

Strength of Monte Carlo Simulation approach

1. **Flexibility:** Monte Carlo simulation can deal with non-linear cases where the linear correlation coefficient cannot provide accurate measure, and allow for the joint distribution other than normal distribution. For example, in this context, it can simulate the data well using the survival BB1 copula model.
2. **Accurate estimation and easier to measure uncertainty:** Monte Carlo simulation is able to provide a more accurate estimation and confidence interval of VaR, as it considers the entire distribution of returns instead of merely the mean and standard deviation of the returns.
3. **Possibility of forecasting:** Monte Carlo simulation allows for the generation of a large number of possible scenarios for future returns, which can be used to forecast the potential outcomes of a portfolio.
4. **Incorporation of Correlations:** Monte Carlo simulation takes into consideration of correlations between variables, which can provide a more accurate estimation of VaR in a portfolio of assets.

Weakness of Monte Carlo Simulation approach

1. **Time-consuming:** Monte Carlo simulation approach can be computationally intensive and time-consuming, especially for portfolios with many assets or complex models. As in our simulation, a large amount of parameters and matrices are produced and simulated, together with multiple models and simulations that might cause dimensionality problems.
2. **High dependence on inputs:** The accuracy of Monte Carlo simulation depends on the quality and accuracy of the input data used to generate the simulations. In addition, it might also have difficulty in modelling extreme events, for example the financial crisis in the year 2008, if the historical data used to do the simulations do not contain relevant information. As in the figure of original log-return and simulated log-return distributions in our task, there is an extreme point on the lower-left corner that the model doesn't have any simulated data.

Task 2c

We see MC approach provides a more conservative estimates of VaR. We shall prefer using MC approach than parametric approach in this case for the following reasons:

1. Parametric method assumes normality for log-returns, but as shown by the Jarque-Bera test results below, there is evidence against this assumption.
2. Parametric method assumes homogeneous variance for log-returns, but from Task 2b's traceplots, Ljung-Box tests, and ACF plots, we see evidence of volatility clusters.
3. Task 2b's copula reveals dependency between the two stocks' log-returns with noticeable (lower) tail dependency; these characteristics are not considered in parametric method.

These reasons makes the results from parametric approach inaccurate and leads to overly optimistic estimate and fails to describe portfolio's dynamics on extreme losses.

Table 7: p.value for Jarque-Bera Test on Normality for Log-Returns

SP500	NK225
0	0

Appendix A

Selecting Conditional Distribution

The code below performed `garchFit` for each combinations of `p` and `q` for `GARCH(p,q)` model for supported conditional distributions. This will take approximately 2-3 minutes to run. Considering the length of the code and result, the code chunk and output here is suppressed. In the following paragraphs we will provide some selected information that are useful when making modelling decision. If one is interested in detailed results we have for all possible combinations of model set-ups, simply run the code in RStudio and inspect the variables `SP500.list` and `NK225.list` in the global environment.

```
## Warning: The programme here (which is hidden in the PDF) will run for
## approximately 2-3 minutes, thanks for your patience, please check accompanied
## .rmd file for details :)
```

The table below shows that under the selected `sstd` conditional distribution, the ICs for each selection of `p` and `q` for the `GARCH(p,q)` models. We can observed that there is no significant improvements in the information criteria for higher `p` and `q`, so we decide to use the conventional `GARCH(1,1)` model to address data's conditional variance and avoid introducing unnecessary complexity.

Table 8: Model ICs for SP500 under `sstd` Distribution

	GARCH(1,1)	GARCH(1,2)	GARCH(2,1)	GARCH(2,2)	GARCH(2,3)	GARCH(3,2)	GARCH(3,3)
AIC	-4.970	-4.969	-4.968	-4.966	-4.963	-4.963	-4.963
BIC	-4.935	-4.929	-4.929	-4.922	-4.913	-4.913	-4.908
SIC	-4.970	-4.969	-4.968	-4.967	-4.963	-4.963	-4.963
HQIC	-4.956	-4.953	-4.953	-4.950	-4.944	-4.944	-4.942

Table 9: Model ICs for NK225 under `sstd` Distribution

	GARCH(1,1)	GARCH(1,2)	GARCH(2,1)	GARCH(2,2)	GARCH(2,3)	GARCH(3,2)	GARCH(3,3)
AIC	-4.317	-4.320	-4.315	-4.318	-4.316	-4.316	-4.314
BIC	-4.287	-4.285	-4.280	-4.278	-4.271	-4.271	-4.264
SIC	-4.317	-4.320	-4.315	-4.318	-4.316	-4.316	-4.314
HQIC	-4.306	-4.307	-4.301	-4.303	-4.299	-4.299	-4.295

To select a suitable conditional distribution for the time-series model, we required the selected distribution to:

1. Address the conditional mean and variance issue in the original data, we inspect this by checking if the model residuals pass the Ljung-Box tests;
2. Perform a PIT transformation that results in a proper uniform distribution, we inspect this by checking if the transformed data pass the Kolmogorov-Smirnov and Anderson-Darling goodness-of-fit tests.

The bellowed table displays some of the most optimal set-ups we obtained.

Table 10: Test p-values and ICs for Selected Conditional Distributions for SP500 under AR(1)-GARCH(1,1)

	norm	snorm	std	sstd	ged	sged
Box.test (Residuals)	0.771	0.285	0.688	0.333	0.777	0.406
Box.test (Residuals Squared)	0.509	0.295	0.157	0.126	0.351	0.192
AIC	-4.899	-4.942	-4.945	-4.970	-4.937	-4.965
BIC	-4.874	-4.913	-4.915	-4.935	-4.908	-4.930
SIC	-4.899	-4.942	-4.945	-4.970	-4.938	-4.965
HQIC	-4.889	-4.931	-4.933	-4.956	-4.926	-4.951
LcKS	0.025	0.015	0.337	0.046	0.245	0.025
ad.test	0.003	0.009	0.020	0.051	0.022	0.044

Table 11: Test p-values and ICs for Selected Conditional Distributions for NK225 under GARCH(1,1)

	norm	snorm	std	sstd	ged	sged
Box.test (Residuals)	0.905	0.904	0.842	0.844	0.867	0.877
Box.test (Residuals Squared)	0.593	0.597	0.341	0.368	0.489	0.528
AIC	-4.248	-4.289	-4.296	-4.317	-4.271	-4.304
BIC	-4.228	-4.264	-4.271	-4.287	-4.246	-4.274
SIC	-4.248	-4.289	-4.296	-4.317	-4.271	-4.304
HQIC	-4.241	-4.280	-4.286	-4.306	-4.262	-4.292
LcKS	0.209	0.471	0.029	0.419	0.018	0.385
ad.test	0.026	0.362	0.075	0.726	0.046	0.526

We see that *skewed standardised t distribution* provides the most satisfying test results and the lowest possible ICs. Thus, we decided to use it as our conditional distribution.