The University of New South Wales
ACTL5301 2023

Models for risk management Term 2 2023 Assignment

Sunday 23rd July, 2023
Alex.Huang - z5210578

Introduction:

After the previous global financial crisis, a mathematical model widely adopted to compute the value of derivatives, the Gaussian copula has been criticised for its inability to measure tail dependence. The following report will conduct an analysis of the estimations based on the historical daily adjusted price of two stocks, Apple Inc. and Alphabet Inc., along with mathematical risk measures, multivariate-t distribution, and the Gaussian copula. The methodology that can be used to replicate the results will be covered first. Key findings will be discussed at the end.

1 Detailed steps:

1.1 Data collection:

Two Listed stocks, Apple Inc (AAPL) and Alphabet Inc (GOOG), were selected. The daily adjusted closing prices between the 1st of January 2010 and 1st of June 2023 were collected from Yahoo Finance and used for the following analysis.

1.2 Daily return series calculation:

We used S_t to represent the stock price of day t, and S_{t-1} to represent the stock price of day $t-1$. The daily return series for Apple Inc and Alphabet Inc were denoted by $X_{A,t}$ and $X_{G,t}$ respectively, were computed by the logarithmic differences between the stock prices of two consecutive days, is given by the equation below:

$$X_t = ln\frac{S_t}{S_{t-1}}$$

We further assumed the daily return on day one to be zero. Hence, the daily return denoted as above can be computed with R by assigning zero to the first day and taking the differences of the logarithmic prices of two consecutive days [Appendix-1].

1.3 Losses of the investment of one dollar:

Assuming that we invest one dollar in one stock at date $t-1$, the value at date t can then be calculated as e^{X_t} with the daily return X_t computed above. Accordingly, the losses of the investment of this one dollar if we choose to invest in Apple Inc or Alphabet Inc, denoted by $L_{A,t}$ and $L_{G,t}$ respectively, can then be computed as below [Appendix-2]:

$$L_t = 1 - e^{X_t}$$

1.4 Comparison of the value at risk and expected shortfall:

Historically, Value at Risk (VaR) and Expected Shortfall (ES) have been regarded as two predominant matrices for measuring the tail risk associated with stock losses.
The empirical value at risk can be computed by taking the $1-\alpha$ quantile of the dollar investment loss above, where α denotes the significance level. Whilst, the expected shortfall of the α level can be calculated by taking the average of all empirical observations with greater losses than the value at risk at α level. [Appendix-3]

α	Apple VaR	Alphabet VaR	Apple ES	Alphabet ES
5%	0.02689929	0.02544856	0.04052336	0.03915147
1%	0.04701582	0.04783018	0.06506210	0.06217268
0.5%	0.05663183	0.05486286	0.07780762	0.07323966

Table 1: Risk measures for individual stocks

We observed that Apple Inc and Alphabet Inc have had similar levels of tail risks from 2010 to 2023.

1.5 Portfolio construction:
An equally weighted portfolio consisting of AAPL and GOOG was constructed for subsequent analysis. Accordingly, the loss per dollar invested in this portfolio at time t, $L_{portfolio}$ can then be computed by taking the average of $L_{A,t}$ and $L_{G,t}$.

1.6 Portfolio risk measures:
Assuming independent returns between AAPL and GOOG, we can simulate the portfolio by generating two independent uniform margins from the interval [0, 1]. These margins are then transformed based on the empirical distribution to derive the portfolio return as mentioned above. This provides us with $VaR_{\alpha}(L)$ and $ES_{\alpha}(L)$ at significance level α. [Appendix-4]

α	VaR portfolio	ES portfolio
5%	0.01874018	0.02753221
1%	0.03225434	0.04236467
0.5%	0.03900738	0.04925862

Table 2: Risk measures based on simulation with independent copula

1.7 Kendall's τ:
Kendall's τ is a statistical measure that quantifies the correlation between the ranks of two variables by assessing the likelihood of observing concordant versus discordant pairs of data points. This value ranges from -1 to 1, with -1 indicating perfect disagreement and 1 indicating perfect agreement. The computation of Kendall's τ involves the following steps:

1. Create duplicate loss vectors $\tilde{L}_A$ and $\tilde{L}_G$, with j representing the date, rather than t.

2. Let t be any value from 1 to one less than the total number of observations, and j any value from $t+1$ to the total number of observations.

3. Treat each $(L_{A,t} - \tilde{L}_{A,j})$ and $(L_{G,t} - \tilde{L}_{G,j})$ as a pair.

Using the above approach, Kendall's τ for the historical losses of investments in AAPL and GOOG can be calculated as shown below [Appendix-5]:

$$\begin{aligned}\rho_\tau(L_A, L_G) &= \mathbf{E}[\text{sign}(X - \tilde{X})(Y - \tilde{Y})] \\ &= \frac{\text{number of concordant pairs} - \text{number of discordant pairs}}{\text{total number of pairs}} \\ &= 0.3940957\end{aligned}$$

Consequently, there is a moderate positive rank correlation between the stock prices of AAPL and GOOG. When the stock price of AAPL increased, the stock price of GOOG also exhibited a moderate likelihood of increasing. The assumption of independence between the stock prices made in 1.6 has been refuted. Copulas and adjusted multivariate models may be used to model the portfolio's performance.

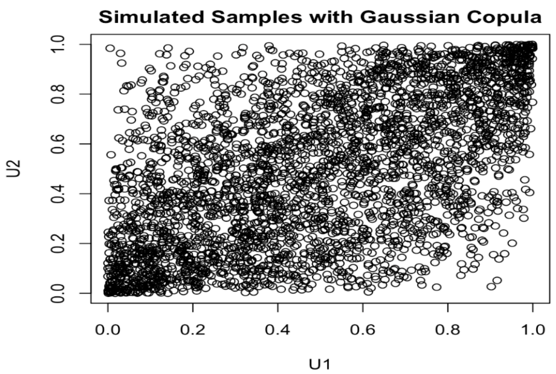
1.8 Gaussian copula and simulation:
As mentioned above, the two stocks AAPL and GOOG had positively correlated performance over the collected period. The Gaussian copula was a widely accepted methodology to take into account this correlation. The joint distribution modelled based on the uniform margins of the loss vectors of the two stocks and the Gaussian copula can be further utilised to quantitatively analyse the tail performance of the portfolio with risk measurements like value at risk and expected shortfall.

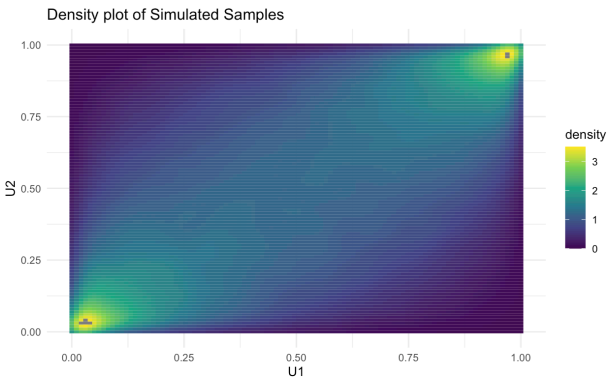
The Sklar's theorem has stated that there is a unique copula C for any continuous multivariate distribution F_X with marginal distributions $F_1 \ldots F_d$. Henceforward, a multivariate distribution can be simulated based on the marginal distributions and a copula C.

Therefore, the simulation could be constructed based on the following steps [Appendix-6]:

1. Compute the correlation coefficient ρ between the return vectors X_A and X_G.
2. Generate two sets of random uniform numbers Z_1 and Z_2.
3. Apply the Gaussian copula transformation on these Gaussian random variables using the calculated correlation coefficient ρ. Specifically, $U_1 = Z_1$ and $U_2 = \rho Z_1 + \sqrt{1-\rho^2} Z_2$. These transformations yield coupled Gaussian distributed variables which are then converted to uniform marginals U_1 and U_2.
4. Transform the simulated uniform marginals to the return rates based on the empirical marginal distribution of the uniform marginals of the collected rates X_A and X_G.
5. Derive the return of the portfolio with the transformed uniform marginals.
6. Conduct risk measure analysis based on the derived return of the portfolio.

This summarises the methodology used for our simulation, and detailed code based on library Copula and alternative solutions are covered in Appendix-6.





After plotting the simulated samples of U_1 and U_2, a moderate level of positive correlation could be observed, which is coherent to our conclusion in 1.7.

α	VaR portfolio	ES portfolio
5%	0.02301201	0.03424118
1%	0.04069086	0.05385854
0.5%	0.04917931	0.06330733

Table 3: Risk measures for the Gaussian simulation

Given that the independence of the two stocks has been disproven, we would expect the Gaussian copula to provide a more accurate simulation. However, the Gaussian copula has faced substantial criticism for its over-simplicity and inability to capture Extreme Value Dependence. In the following sections, we will analyze the collected data using alternative methods and compare the risk measures on tail dependences to determine the best performing estimate.

1.9 Multivariate t-distribution:

A multivariate t-distribution is a variation of the multivariate normal distribution that can be written as below:

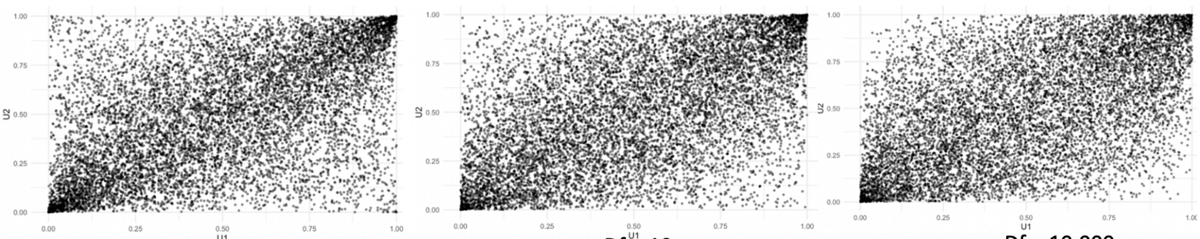
$$\mu + \sqrt{W} A \mathbf{Z} \sim t_d(\nu,\, \mu,\, \Sigma = AA^T)$$

Where, W is a positive random variable that satisfies $W \sim IG(\frac{1}{2}\nu,\, \frac{1}{2}\nu)$. The extra parameters that the t-distribution introduces compared to the multivariate normal distribution not only enhance flexibility and allow for a better fit to data but also improve the tail behaviour of the thin-tailed multivariate normal distribution. Consequently, the multivariate t-distribution can be an excellent alternative to simulation with the Gaussian copula.

1.10 Simulation with Multivariate t-distribution:

As discussed above, the multivariate t-distribution offers enhanced flexibility and improved tail behaviour, potentially leading to a more accurate estimation of tail risk. However, the degree of freedom in the t-distribution can influence tail performance. Consequently, we will consider three degrees of freedom, $\nu = \{3, 10, 10000\}$, in our simulation. Additionally, we assume the transformed uniform margins are given by $(U_1, U_2)^T \sim C^t_{\nu,\Sigma}$. This implies that the uniform margins follow a t-distribution, denoted as $t_2(\nu, 0, \Sigma = AA^T)$.
Following Sklar's theorem introduced above, we can construct our simulation based on the following steps [Appendix-7]:

1. Compute the correlation coefficient ρ between the return vectors X_A and X_G.
2. Construct the two by two correlation matrix Σ with diagonal elements equals to one.
3. Generate the multivariate t-distribution samples:
   (a) Generate a series of W values from the inverse gamma distribution.
   (b) Generate independent standard normal values Z.
   (c) Combine the square root of W, Cholesky decomposition of Σ and $\mathbf{Z}$ to form T that follows the distribution: $T = \sqrt{W} AZ$.
4. Transform the simulated multivariate t-samples to uniform marginals, U_1 and U_2, using the cumulative distribution function of the t-distribution.
5. Convert the uniform marginals back to the empirical margins of the data.
6. Compute the simulated portfolio loss, L_{hat}, based on the transformed empirical margins.
7. Conduct risk measure analysis based on the simulated portfolio.

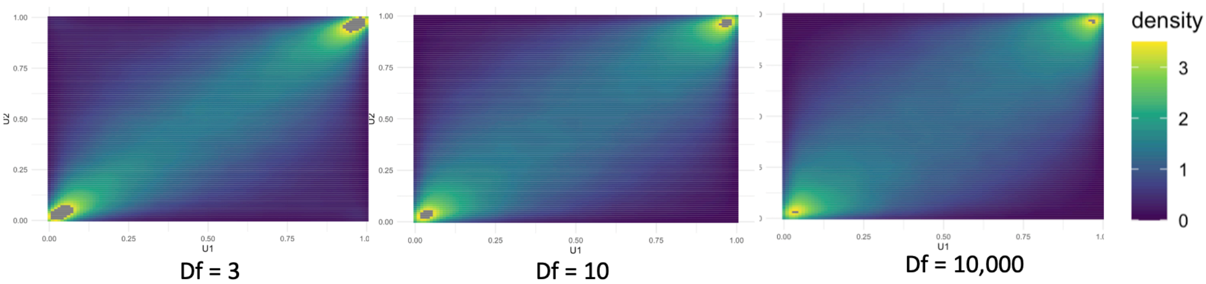


Df = 3 Df = 10 Df = 10,000

Based on our sampled uniform margins U_1 and U_2, we observed a moderate level of positive correlation. When degrees of freedom ν went up, the weaker the correlation behaved, and fewer simulated samples were observed in corners, indicating a weaker tail dependence between the uniform margins.

1.11 t-distribution and the degree of freedom ν:

The visualizations provided below shows a distinct concentration in the diagonal region, consistent with the previously mentioned positive correlation between AAPL and GOOG.



Moreover, the most densely populated regions in the plots are the grey areas located in the bottom-left and top-right corners. The decreasing size of these grey regions underscores our earlier observation about the diminishing tail dependence as captured by the t-distribution when the degree of freedom increases. Comparing the three density plots with the one generated using the Gaussian copula in Section 1.8, it becomes evident that as ν increases, the multivariate t-copula approaches the behaviour of a Gaussian copula, neglecting tail dependencies.

Henceforward, t-copula would not only have the issue of symmetric assumption innate from a multivariate normal distribution, but the added parameters also increase the complexity of model estimation, potentially leading to overfitting issues. Thus, given its adaptability and capability to capture non-elliptical dependence, the Gaussian copula might be favoured in certain scenarios.

2 Comparison of results and conclusion:

Following the section above, the risk measures for different copulas were computed as below:

α	Independent	Gaussian	$t_{\nu=3}$	$t_{\nu=10}$	$t_{\nu=10000}$
5%	0.01874018	0.02301201	0.02299731	0.02331809	0.02339855
1%	0.03225434	0.04069086	0.04163350	0.04094092	0.04029836
0.5%	0.03900738	0.04917931	0.05066143	0.04933221	0.04843529

Table 4: Value at risk (VaR) for different copulas and degrees of freedom

α	Independent	Gaussian	$t_{\nu=3}$	$t_{\nu=10}$	$t_{\nu=10000}$
5%	0.02753221	0.03424118	0.03491201	0.03444255	0.03404691
1%	0.04236467	0.05385854	0.05597512	0.05369724	0.05196076
0.5%	0.04925862	0.06330733	0.06642272	0.06280950	0.06015303

Table 5: Expected Shortfall (ES) for different copulas and degrees of freedom

The t-copula with a degree of freedom $\nu = 3$ generated the highest risk measure, followed by the t-copula with a degree of freedom $\nu = 10$. The Gaussian copula simulation yielded significantly lower risk measures than the t-copula when the degree of freedom was low. In practice, this suggested a high likelihood of underestimating the risk borne. Alternatively, the tail dependence coefficient, denoted below, can serve as a qualitative measure when assessing the simulated uniform margins.

$$\lambda_{\text{u or l}} = \lim_{u \to 0^+ \text{ or } 1^-} \mathbf{P}\left(Y > F_Y^{-1}(u) \mid X > F_X^{-1}(u)\right)$$

In conclusion, the t-copula appeared to provide the most accurate estimation for risk measures. However, there is potential for overfitting in the above risk measures, necessitating further adjustments. While the Gaussian copula may not be the optimal choice in this context, its versatility remains significant and may prove beneficial in other scenarios.

Appendices :

Appendix 1 logarithmic differences:

```r
data$XG <- c(0, diff(log(data$GOOG)))
data$XA <- c(0, diff(log(data$AAPL)))
```

R code for computing logarithmic differences

Appendix 2 Losses of the investments of one dollar in the two stocks:

```r
data$LG <- 1 - exp(data$XG)
data$LA <- 1 - exp(data$XA)
```

Losses of the investments of one dollar in the two stocks

Appendix 3 Tail risk comparison:

```r
alphas <- c(0.05, 0.01, 0.005)
results_task4 <- data.frame(Alpha = numeric(),
                            AAPL_VaR = numeric(),
                            GOOG_VaR = numeric(),
                            AAPL_ES = numeric(),
                            GOOG_ES = numeric())

for (alpha in alphas) {
  aapl_var <- quantile(data$LA, 1-alpha,  na.rm=TRUE)
  goog_var <- quantile(data$LG, 1-alpha,  na.rm=TRUE)

  aapl_es <- mean(data$LA[data$LA > aapl_var],  na.rm=TRUE)
  goog_es <- mean(data$LG[data$LG > goog_var], na.rm=TRUE)

  results_task4 <- rbind(results_task4, data.frame(Alpha = alpha,
                                                   AAPL_VaR = aapl_var,
                                                   GOOG_VaR = goog_var,
                                                   AAPL_ES = aapl_es,
                                                   GOOG_ES = goog_es))
}

print(results_task4)
```

Tail risk comparison

Appendix 4 Portfolio VaR and ES:

```r
data$Lportfolio <- 0.5 * (data$LA + data$LG)

results_task6 <- data.frame(Alpha = numeric(),
                            VaR = numeric(),
                            ES = numeric())

for (alpha in alphas) {
  portfolio_var <- quantile(data$Lportfolio, 1-alpha, na.rm=TRUE)
```

```r
  portfolio_es <- mean(data$Lportfolio[data$Lportfolio > portfolio_var], na.rm=TRUE)

  results_task6 <- rbind(results_task6, data.frame(Alpha = alpha,
                                                   VaR = portfolio_var,
                                                   ES = portfolio_es))
}

print(results_task6)

set.seed(12321)
n <- 10000
U_1 <- runif(n)
U_2 <- runif(n)

X_A_ind <- quantile(data$XA,U_1, type = 1)
X_G_ind <- quantile(data$XG,U_2, type = 1)
L_A_sim <- 1- exp(X_A_ind)
L_G_sim <- 1- exp(X_G_ind)
L_sim_ind <- 0.5*(L_A_sim+L_G_sim)

results_task6_ind <- data.frame(Alpha = numeric(),
                                VaR = numeric(),
                                ES = numeric())

for (alpha in alphas) {
  ind_portfolio_var <- quantile(L_sim_ind, 1-alpha, na.rm=TRUE)
  ind_portfolio_es <- mean(L_sim_ind[L_sim_ind > ind_portfolio_var], na.rm=TRUE)

  results_task6_ind <- rbind(results_task6_ind, data.frame(Alpha = alpha,
                                                           VaR = ind_portfolio_var,
                                                           ES = ind_portfolio_es))
}

print(results_task6_ind)

q_lower <- 0.05
q_upper <- 0.95

# Estimate lower tail dependence coefficient
lambda_L_empirical <- sum(U_1 <= q_lower & U_2 <= q_lower) / sum(U_1 <= q_lower)

```

```
# Estimate upper tail dependence coefficient
lambda_U_empirical <- sum(U_1 > q_upper & U_2 > q_upper) / sum(U_1 > q_upper)

cat("Empirical Lower Tail Dependence Coefficient:", lambda_L_empirical, "\n")
cat("Empirical Upper Tail Dependence Coefficient:", lambda_U_empirical, "\n")
```

Portfolio VaR and ES

```
data$Lportfolio <- 0.5 * (data$LA + data$LG)

results_task6 <- data.frame(Alpha = numeric(),
                            VaR = numeric(),
                            ES = numeric())

for (alpha in alphas) {
  portfolio_var <- quantile(data$Lportfolio, 1-alpha, na.rm=TRUE)
  portfolio_es <- mean(data$Lportfolio[data$Lportfolio > portfolio_var], na.rm=TRUE)

  results_task6 <- rbind(results_task6, data.frame(Alpha = alpha,
                                                   VaR = portfolio_var,
                                                   ES = portfolio_es))
}

print(results_task6)

set.seed(12321)
n <- 10000
U_1 <- runif(n)
U_2 <- runif(n)

X_A_ind <- quantile(data$XA,U_1, type = 1)
X_G_ind <- quantile(data$XG,U_2, type = 1)
L_A_sim <- 1- exp(X_A_ind)
L_G_sim <- 1- exp(X_G_ind)
L_sim_ind <- 0.5*(L_A_sim+L_G_sim)

results_task6_ind <- data.frame(Alpha = numeric(),
                                VaR = numeric(),
                                ES = numeric())

for (alpha in alphas) {
  ind_portfolio_var <- quantile(L_sim_ind, 1-alpha, na.rm=TRUE)
  ind_portfolio_es <- mean(L_sim_ind[L_sim_ind > ind_portfolio_var], na.rm=TRUE)

  results_task6_ind <- rbind(results_task6_ind, data.frame(Alpha =
```

```
     alpha,
                                                                   VaR = ind_
                                                                      portfolio
                                                                      _var,
                                                                   ES = ind_
                                                                      portfolio
                                                                      _es))
}

print(results_task6_ind)

q_lower <- 0.05
q_upper <- 0.95

# Estimate lower tail dependence coefficient
lambda_L_empirical <- sum(U_1 <= q_lower & U_2 <= q_lower) / sum(U_1
   <= q_lower)

# Estimate upper tail dependence coefficient
lambda_U_empirical <- sum(U_1 > q_upper & U_2 > q_upper) / sum(U_1 >
   q_upper)

cat("Empirical Lower Tail Dependence Coefficient:", lambda_L_
   empirical, "\n")
cat("Empirical Upper Tail Dependence Coefficient:", lambda_U_
   empirical, "\n")
```

Portfolio VaR and ES

Appendix 5 Kendal's τ:

```
# Initialise
total_row <- nrow(data)
number_con <- 0
number_dis <- 0

# Loop
for (i in 1:(total_row-1)) {
  for (j in (i+1):total_row) {

    concordant <- (data$LA[i] - data$LA[j]) * (data$LG[i] - data$LG[j
       ]) > 0
    discordant <- (data$LA[i] - data$LA[j]) * (data$LG[i] - data$LG[j
       ]) < 0

    number_con <- number_con + concordant
    number_dis <- number_dis + discordant
  }
}

# Kendall's tau
tau <- (number_con - number_dis) / (number_con + number_dis)

```

```
print(tau)
```

Calculation of Kendall's tau

Appendix 6 Gaussian copula simulation:

```
# Set number of iterations
n_samples <- 100*nrow(data)

# Transform returns into uniform margins
data$U1 <- rank(data$XA, na.last = "keep") / (nrow(data) + 1)
data$U2 <- rank(data$XG, na.last = "keep") / (nrow(data) + 1)

# Estimate the correlation coefficient
rho_hat <- cor(data$XA, data$XG, use = "complete.obs")

# Create the Gaussian copula
gaussian_cop <- normalCopula(rho_hat)

set.seed(54321)

simulated_U <- rCopula(n_samples, gaussian_cop)
plot(simulated_U, main = "Simulated Samples with Gaussian Copula",
    xlab = "U1", ylab = "U2")

# plot with ggplot2
simulated_df <- as.data.frame(simulated_U)
colnames(simulated_df) <- c("U1", "U2")

ggplot(simulated_df, aes(x = U1, y = U2)) +
  stat_density2d(aes(fill = ..density..), geom = "tile", contour =
      FALSE, n = 100) +
  scale_fill_viridis_c(limits=c(0, 3.5)) +
  labs(title = "Density plot of Simulated Samples", x = "U1", y = "U2
      ") +
  theme_minimal()

X_hat_A <- quantile(data$XA, simulated_U[, 1], na.rm = TRUE)
X_hat_G <- quantile(data$XG, simulated_U[, 2], na.rm = TRUE)
L_hat <- 0.5 * (1 - exp(X_hat_A) + 1 - exp(X_hat_G))
results_task8 <- data.frame(Alpha = numeric(), VaR = numeric(), ES =
    numeric())

for (alpha in alphas) {
  simulated_var <- quantile(L_hat, 1 - alpha)
  simulated_es <- mean(L_hat[L_hat > simulated_var])
  results_task8 <- rbind(results_task8, data.frame(Alpha = alpha, VaR
      = simulated_var, ES = simulated_es))
}

# Tail Dependence Computations
q_lower = 0.05
q_upper = 0.95
```

```
lambda_L_gaussian = sum(simulated_U[,1] <= q_lower & simulated_U[,2] <= q_lower) / sum(simulated_U[,1] <= q_lower)
lambda_U_gaussian = sum(simulated_U[,1] > q_upper & simulated_U[,2] > q_upper) / sum(simulated_U[,1] > q_upper)

print(paste("Gaussian Copula Lower Tail Dependence:", lambda_L_gaussian))
print(paste("Gaussian Copula Upper Tail Dependence:", lambda_U_gaussian))
print(results_task8)
```

Gaussian copula simulation

```
rho <- cor(data$XA, data$XG)

# Create uniform distributed random variables
U1 <- runif(n_samples, 0, 1)
U2 <- runif(n_samples, 0, 1)

# Apply the Gaussian copula transformation
C <- qnorm(c(U1, U2), mean = 0, sd = 1, lower.tail = TRUE)
C <- matrix(C, ncol = 2)
Z1 <- C[,1]
Z2 <- C[,2]
Y1 <- Z1
Y2 <- rho*Z1 + sqrt(1-rho^2)*Z2
simulated_U1 <- pnorm(Y1)
simulated_U2 <- pnorm(Y2)
simul_XA <- quantile(data$XA, simulated_U1, na.rm = TRUE, type = 8)
simul_XG <- quantile(data$XG, simulated_U2, na.rm = TRUE, type = 8)
simul_LA <- 1 - exp(simul_XA)
simul_LG <- 1 - exp(simul_XG)
L_hat = 0.5 * (simul_LA + simul_LG)
results_task8 <- data.frame()

for (alpha in alphas) {
  simulated_var <- quantile(L_hat, 1 - alpha)
  simulated_es <- mean(L_hat[L_hat > simulated_var])
  results_task8 <- rbind(results_task8, data.frame(Alpha = alpha, VaR = simulated_var, ES = simulated_es))
}

# Tail Dependence Computations
simulated_U <- data.frame(U1 = simulated_U1, U2 = simulated_U2)

q_lower = 0.05
q_upper = 0.95
lambda_L_gaussian = sum(simulated_U$U1 <= q_lower & simulated_U$U2 <= q_lower) / sum(simulated_U$U1 <= q_lower)
lambda_U_gaussian = sum(simulated_U$U1 > q_upper & simulated_U$U2 > q_upper) / sum(simulated_U$U1 > q_upper)

```

```
print(paste("Gaussian Copula Lower Tail Dependence:", lambda_L_gaussian))
print(paste("Gaussian Copula Upper Tail Dependence:", lambda_U_gaussian))
print(results_task8)
```

Gaussian copula simulation without the copula library

Appendix 7: t-copula simulation:

```
# Initialisation
n_samples <- 100*nrow(data)
rho_hat <- cor(data$XA, data$XG, use = "complete.obs")
sigma <- matrix(c(1, rho_hat, rho_hat, 1), 2)  # The 2x2 correlation matrix
dof_values <- c(3, 10, 10000)  # Degrees of freedom for t-copula
simulated_U_list <- list()
lambda_L_values <- numeric()  # For storing lower tail coefficients
lambda_U_values <- numeric()  # For storing upper tail coefficients

mu <- c(0,0)
rho_temp <- cov(data$XA, data$XG, use = "complete.obs")
sigma_temp <- matrix(c(1, rho_hat, rho_hat, 1), 2)  # The 2x2 correlation matrix
A <- t(chol(sigma_temp))

for (dof in dof_values) {
  # Simulate from the multivariate t-distribution
  W <- 1/rgamma(n_samples, shape=0.5*dof, scale=2/dof)  # Using the inverse gamma distribution
  Z <- MASS::mvrnorm(n_samples, mu=c(0,0), Sigma=diag(2))  # Independent standard normals
  T <- matrix(0, n_samples, 2)

  for (i in 1:n_samples) {
    T[i,] <- mu + sqrt(W[i]) * (A %*% Z[i,])
  }

  # Convert to uniform using the pt function
  U1 <- pt(T[,1], df=dof)
  U2 <- pt(T[,2], df=dof)

  # Compute tail dependence coefficients based on simulated U_t
  q_lower = 0.05
  q_upper = 0.95
  lambda_L_empirical = sum(U1 <= q_lower & U2 <= q_lower) / sum(U1 <= q_lower)
  lambda_U_empirical = sum(U1 > q_upper & U2 > q_upper) / sum(U1 > q_upper)

  # Store the coefficients
  lambda_L_values <- c(lambda_L_values, lambda_L_empirical)
  lambda_U_values <- c(lambda_U_values, lambda_U_empirical)
```

```r

  # Convert uniform margins to data margins using empirical CDF directly in loop
  X_hat_A <- quantile(data$XA, U1, na.rm = TRUE)
  X_hat_G <- quantile(data$XG, U2, na.rm = TRUE)

  # Compute the simulated portfolio loss
  L_hat <- 0.5 * (1 - exp(X_hat_A) + 1 - exp(X_hat_G))

  results_for_10 <- data.frame(Alpha = numeric(), VaR = numeric(), ES = numeric())

  for (alpha in alphas) {
    simulated_var <- quantile(L_hat, 1 - alpha)
    simulated_es <- mean(L_hat[L_hat > simulated_var])
    results_for_10 <- rbind(results_for_10, data.frame(Alpha = alpha, VaR = simulated_var, ES = simulated_es))
  }

  results_task10[[paste("Degrees of freedom =", dof)]] <- results_for_10
  simulated_U_list[[paste("Degrees of freedom =", dof)]] <- U_t
}

# Print results
for (key in names(results_task10)) {
  print(paste("Results for", key))
  print(results_task10[[key]])
}

# Print tail coefficients
results_lambda = data.frame(
  Degree_of_Freedom = dof_values,
  Lower_Tail_Dependence = lambda_L_values,
  Upper_Tail_Dependence = lambda_U_values
)
print(results_lambda)

results_task10
```

R code for simulation and analysis