ACTL3301 – Quantitative Risk Management

Risk Modelling Report

Dhwanish Kshatriya

(z5421168)

19 July 2024

Methodology

Preparation

In preparation for the investigation, data retrieval and processing were performed as explained in *Section 1*. This section also defines the terminology and notation used throughout the investigation.

Evaluating Tail Risk

Throughout the investigation, risk was evaluated using the metrics of Value at Risk (VaR) and Expected Shortfall (ES). As the cumulative distribution function (CDF) of the company-data is not continuous, a few adjustments had to be made to calculate each metric.

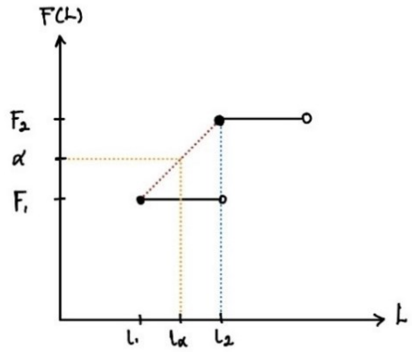

Figure 1 - VaR Adjustment

First, we define $VaR_\alpha(L)$ as the smallest value of loss l such that such that the $\Pr(L > l) \leq 1 - \alpha$. Since our empirical CDF is discrete, there are cases where the αth percentile occurs at a discontinuity, as reflected *Figure 1*. Under such scenarios, when we use the `quantile()` function in RStudio, it interpolates and give us a smaller VaR value (l_α) than the true value (l_2). This is reflected by the red and orange lines in *Figure 1*. Therefore, an adjustment is made which sets the VaR as the smallest loss in the eCDF that is greater than or equal to l_α, correctly finding the VaR (represented as l_2 in *Figure 1*).

Next, we define $ES_\alpha(L)$ as the expected loss given that the loss $l \geq VaR_\alpha$. For our empirical CDF this can be expressed as:

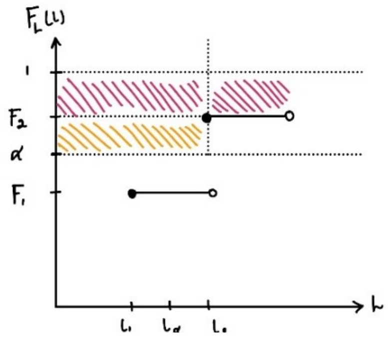

Figure 2 - ES Adjustment

$$ES_\alpha(L) = \mathbb{E}[L | L \geq VaR_\alpha(L)] + \frac{q_\alpha(L) \cdot (\Pr(L \leq q_\alpha) - \alpha)}{1 - \alpha} \quad (1)$$

The first term in equation (1) can be estimated by the mean loss for loss values $l \geq VaR_\alpha(L)$ using the `mean()` function in RStudio. This is reflected by the red area in *Figure 2*. Then, there is an additional area over the discontinuity in *Figure 2* reflected by the second term in equation (1). To account for this, we further add:

$$\frac{VaR_\alpha(L) \times (F_2 - \alpha)}{1 - \alpha} = \frac{l_2 \times (F_2 - \alpha)}{1 - \alpha}$$

Using the above methodology, $VaR_\alpha(L)$ and $ES_\alpha(L)$ were calculated for L_A, L_N and $0.5 \times \{L_A + L_N\}$ at the levels $\alpha \in \{0.95, 0.99, 0.995\}$. The summarised results can be found in *Table 1* under the appendix.

Evaluating Correlation

As an assessment of correlation between the potential losses of AMD and NVDA, the rank correlation metrics of Kendall's tau (ρ_τ) and Spearman's rho (ρ_S) were used.

As shown in *Derivation 1*, we know $\rho_\tau = \Pr(\text{Concordance}) - \Pr(\text{Discordance})$. To estimate these probabilities, we count the number of unique concordant and discordant pairs of losses, and divide by the number of total unique pairs. Thus, for each unique pair, we count according to the following:

- A pair of losses are concordant if $(L_{A,t_i} - L_{A,t_j}) \times (L_{N,t_i} - L_{N,t_j}) > 0$.
- A pair of losses are discordant if $(L_{A,t_i} - L_{A,t_j}) \times (L_{N,t_i} - L_{N,t_j}) < 0$.

- The total number of unique pairs of L is `choose(n, 2)`, where n is the number of losses per stock.

Spearman's rho (ρ_S) is defined as the linear correlation (ρ) between the distribution function of each loss. Mathematically, this can be represented as:

$$\rho_S(L_A, L_N) = \rho\left(F_{L_A}(l_A), F_{L_N}(l_N)\right)$$

This was estimated by finding the eCDF of the losses of each stock using the `ecdf()` function in RStudio, and then finding the (Pearson's) correlation between each eCDF using the `cor()` function.

Simulating Copula Models & Joint Risk:

Throughout the investigation, various dependence structures were tested to evaluate the portfolio risk. This involved making underlying dependence assumptions, and thus simulating copulas to model the joint distribution of sample stock returns. The simulations were repeated for $i \in [1, 5000]$, and then using Sklar's theorem, the simulated copulas and empirical margins of each stock were used to find the joint distribution. After simulating the required copula $(U_1 \quad U_2)^T$, the following formula was used for each i to generate sample returns:

$$\begin{pmatrix} X_{A,i} \\ X_{N,i} \end{pmatrix} = \begin{pmatrix} \hat{F}_A^{-1}(U_1) \\ \hat{F}_N^{-1}(U_1) \end{pmatrix}, \text{ where } \hat{F}_A^{-1}, \hat{F}_N^{-1} \text{ are the inverse empirical CDFs.}$$

These were then converted into sample losses using $L_t = 1 - \exp\{X_t\}$. Then a portfolio was created using $L_t = 0.5 \times (L_{A,t} + L_{N,t})$, and its risk was evaluated using the VaR and ES metrics as described above. The process for deriving the above equation can be found in the *Derivation 2* section under the appendix. Further, the results for portfolio risk under each assumption are summarised from *Table 2* to *Table 8*. Finally, the simulated copulas and their simulation methods are described below:

Independence Copula:

To simulate a portfolio under the assumption of independent returns X_A and X_N, then an independence copula was simulated. First, we started by simulating two independent $U(0,1)$ random variables U_1, U_2. Here, $(U_1, \ U_2)^T$ follows the distribution of an independence copula.

Co & Counter Monotonicity Copulas

Then, to simulate a comonotonicity copula, we set $U_2 = U_1$ (as simulated above) to ensure perfect positive correlation. Similarly, to simulate a counter monotonicity copula, we set $U_2 = 1 - U_1$ to ensure both perfect negative correlation and $U_2 \sim U(0,1)$. For each of these copulas, their respective $(U_1, \ U_2)^T$ followed the distribution of the desired Copula.

Gaussian & t Copula:

To simulate a portfolio with Gaussian and t copulas, similar processes were followed. Both processes involved the following steps:

1. Simulating iid, $U(0,1)$ random variables u_1, u_2.
2. Estimating $\rho = \text{corr}(\boldsymbol{X}_A, \boldsymbol{X}_N) \approx \hat{\rho}$ (sample correlation) using the `cor()` function in RStudio.
3. Setting $Z_i = \Phi^{-1}(u_i)$ to transform uniform samples into iid, $N(0, 1)$ samples Z_1, Z_2
4. Setting $Y_1 = Z_1, Y_2 = \sqrt{1 - \rho^2} \times Z_2$ so to obtain a correlated normal samples Y_1, Y_2 with correlation ρ. A further explanation for this can be found in the appendix under *Section 2*.

Then, to obtain a Gaussian copula, we apply $U_i = \Phi(Y_i)$, resulting in $(U_1 \quad U_2)^T \sim C_\rho^{\mathrm{Ga}}$. However, simulating an t copula involves a few additional steps. These were repeated for 5000 times for each $\nu \in \{3, 10, 10^4\}$.

5. Simulate a Gamma$(0.5\nu, 0.5\nu)$ random variable G, and then take G^{-1} to obtain $W \sim IG(0.5\nu, 0.5\nu)$.
6. Set $\tilde{Y}_i = \sqrt{W} \times Y_i$ to transform correlated normal samples Y_1, Y_2 to samples with distribution $(\tilde{Y}_i \quad \tilde{Y}_i)^T \sim t_d(\nu, \mathbf{0}, \mathbf{\Sigma})$. Here Σ is defined equivalently to the definition under *Section 2*.

Finally, to obtain a t copula, we apply $U_i = t_d(\tilde{Y}_i, \nu)$ where $t_d(\nu)$ is the CDF of a t distributiuon with ν degrees of freedom. This results in $(U_1 \quad U_2)^T \sim C_\rho^{\mathrm{t}}$.

Estimating Tail Dependence

We have defined $U_1 = F(X_N)$ and $X_2 = F(X_A)$ for each copula above. Next, we define tail dependence as $\Pr(F(X_N) > \alpha | F(X_A) > \alpha))$. This can then be expanded using the conditional probability formula, and then estimated by using the `count()` function in RStudio.

$$\Pr(U_1 > \alpha, U_2 > \alpha) = \frac{\Pr(U_1 > \alpha, U_2 > \alpha)}{\Pr\,(U_2 > \alpha)} \approx \frac{\mathrm{count}(U_1 > \alpha, U_2 > \alpha)}{\mathrm{count}(U_2 > \alpha)}$$

The following was applied to each copula, for each confidence level. Results for tail dependence are summarised in *Table 2* to *Table 8*.

Results & Discussion

Correlation

Using the method described in the sections above, the rank correlation metrics of Kendall's tau (ρ_τ) and Spearman's rho (ρ_S) were estimated. There are various benefits to using such metrics instead of Pearson's correlation (ρ), however for this investigation we will simply note that these metrics better capture non-linear relationships and relationships in ordinal data.

Specifically, we obtained the estimates $\hat{\rho}_\tau = 0.447$ and $\hat{\rho}_S = 0.617$. In this investigation, Kendall's tau was used to measure the concordance and discordance of the losses of each stock. Since $\hat{\rho}_\tau = 0.447$, we know that there is moderate concordance between NVDA and AMD's potential losses. This means that stock losses move in the same direction a moderate proportion of the time. Similarly, Spearman's rho was used to measure the correlation between the empirical CDFs. A value $\hat{\rho}_S = 0.617$ indicates moderate to strong positive correlation between stock losses. This means that as the loss of one stock increases, the loss of another stock tends to increase too. Throughout the remainder of the investigation, we will refer to these values to assess the validity of specific dependence assumptions.

Analysis of Individual Investment Risk

First, we will start by comparing the relative rail risk of NVDA and AMD based on the VaR and ES metrics calculated in *Table 1*. Here, we can see that for all significance levels α, $VaR_\alpha(L_N) < VaR_\alpha(L_A)$. This means that for $\alpha \in \{0.95, 0.99, 0.995\}$, the potential α percentile losses are smaller for NVDA than they are for AMD, indicating it is a less risky investment. However, VaR does not effectively capture the risk associated with experiencing upper tail losses. To do so, we compare the expected shortfall of each stock, observing that $ES_\alpha < ES_\alpha(L_A)$ for $\alpha \in \{0.95, 0.99, 0.995\}$. This means that even under extreme events when experiencing upper tail losses greater than the α percentile, the expected loss is smaller for NVDA and AMD. Therefore, according to both metrics, NVDA is seemingly a less risky investment especially when considering tail losses.

Analysis & Comparison of Portfolio Risk

The following section will provide a comparison of risk associated with a portfolio that follows $L = 0.5 \times \{L_A + L_N\}$. Specifically, we will compare the VaR and ES values simulated under each assumption to the values obtained empirically. Then, we will provide a recommendation as to which copula model best estimates the risk of this portfolio.

Independence Copula

Starting with the independence copula model, we observe significant differences in VaR. Particularly, VaR is significantly lower under the independence assumption (*Table 2*) than the empirical model (*Table 1*) for each $\alpha \in \{0.95, 0.99, 0.995\}$. Further, the difference between each VaR_α value between the empirical and independence models is higher for $\alpha = 0.99$ and 0.995. Similarly, the ES is also significantly lower under the independence assumption when compared to our empirical model for each α. There is a similar pattern, with ES deviating more for the larger significance levels. Further, our analysis of correlation above and the fact that estimated Pearson's correlation $\hat{\rho} = 0.559 \gg 0$ indicates that the independence assumption does not hold. Therefore, this model is a poor estimate of portfolio risk.

Co/ Counter Monotonicity Copulas

Next, we consider the comonotonicity and counter monotonicity copula models. We observe that the VaR and ES calculated under the counter monotonicity copula (*Table 8*) model are orders of magnitude lower than that of the empirical portfolio (*Table 1*). This indicates that portfolio risk is not accurately modelled using this model. A possible reason for this significant difference is that, since losses exhibit perfect negative dependence under this model, the losses of each stock counteract one-another. Relating back to our analysis of rank correlation, both Spearman's rho and Kendall's tau indicated a positive dependence, and thus perfect negative correlation is likely an invalid assumption. This exact issue does not apply for the comonotonicity copula model, however, as it assumes positive dependence. Regardless, we observe that the model (relatively) significantly overestimates the empirical VaR and ES. While these differences are not as pronounced as the independence or counter monotonicity copula, they are still significant enough to indicate that the model is inaccurate. A possible reason for this inaccuracy could be due to the fact correlation is significantly lower than 1, as discussed in the sections above. Therefore, the stock losses are unlikely to be perfectly positively correlated.

Gaussian & t Copulas

Next, we will look at Gaussian copula model. Our previous analysis indicates a clear dependence structure between stock losses. Accordingly, the Gaussian copula model was simulated using the estimated Pearson's correlation $\hat{\rho} = 0.559$.

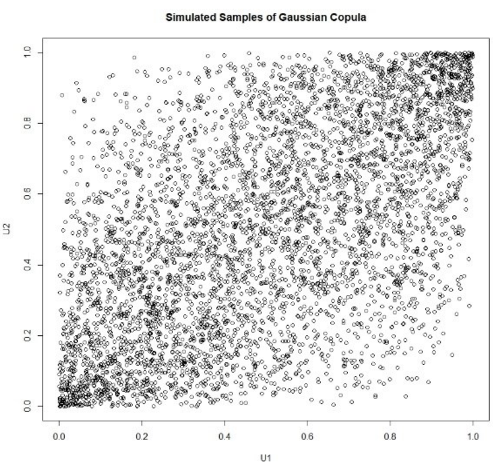

Figure 3 - Gaussian Copula Samples

We can observe this dependence structure in the simulated pairs (U_1, U_2), with *Figure 3* showcasing relatively significant coupling of pairs around coordinates $(0, 0)$ and $(1, 1)$. Alternatively, there is a smaller magnitude of pairs of samples around the coordinates of $(1, 0)$ and $(0, 1)$. This is consistent with the positive correlation previously discussed. To assess how effectively the Gaussian copula model resembles tail risk, we can see *Figure 5*, which showcases a lack of pairs of upper and lower tail losses. This reinforces the Gaussian copula's relative inability capture tail dependence.

When comparing VaR against the empirical portfolio, we can see that the gaussian copula model slightly underestimates the VaR of the empirical portfolio for $\alpha = 0.95$ and 0.99. However, for $\alpha = 0.995$, the VaR values are equal when rounded to 3 decimal places. Similarly, for ES, the gaussian copula model slightly underestimates the empirical portfolio's ES for $\alpha = 0.95$ and 0.995, but is equal to 3 decimal places when considering $\alpha = 0.99$.

To better consider the tail risk, we can use a variance mixture structure and t-copula, specifically with degrees of freedom $\nu = \{3, 10, 10^4\}$. Similarly, the t-copula model was simulates using the estimated Pearson's correlation $\hat{\rho} = 0.559$. It is important to note that as $\nu \to \infty$, the t-copula resembles a Gaussian Copula. This is well reflected in the plots of the pairs (U_1, U_2) between the Gaussian Copula and t-copula with $\nu = 10^4$ (*Figure 3* and *Figure 6*). As expected, the estimates for VaR and ES under this t-copula are equivalent to the Gaussian copula (equal to when rounded to 3 decimal places), and thus it models portfolio risk similarly to the Gaussian copula when compared to the empirical portfolio risk. Whilst it may seem like the t-copula with $\nu = 10^4$ would capture tail risk more than the Gaussian model, the tail dependence estimates in *Table 6* and *Table 3* showcase that both models estimate a similar dependence.

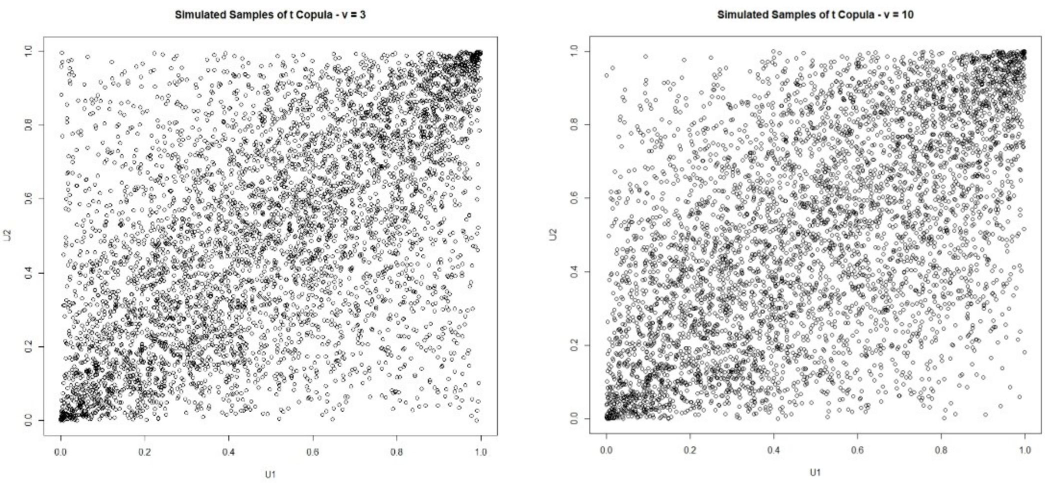

Figure 4 - t copula Samples

Finally, we consider the t-copulas with degrees of freedom $\nu = 3$ and 10. It is important to note that for smaller ν, the tail of a t distribution is fatter and thus capture tail loss more. This is reflected by the increased density of samples around the coordinates $(0, 0)$ and $(1, 1)$ *Figure 4*. We also observe a relatively higher amount of pairs (U_1, U_2) about the points (1,0) and $(0, 1)$ compared to the Gaussian copula. However, this is not necessarily desirable, as our correlation values indicate a positive dependence between copulas. When comparing risk metrics against the empirical portfolio, *Table 4* and *Table 5* show us that both these t-copula models significantly overestimate the VaR and ES for $\alpha = 0.99$ and 0.995. This means that for the given data, the t-copulas may be overemphasising the tail risk. Alternatively, for $\alpha = 0.95$, both t-copula models underestimate the VaR and ES when compared to the empirical portfolio and perform similarly to the Gaussian copula.

Conclusions

Based on the discussions above, we observe the Gaussian copula to model to most accurately model the risk of the empirical portfolio. As a t-copula converges to a Gaussian copula as $\nu \to \infty$, there may be situations in which for a specific ν, the t-copula model better reflects the risk associated with the portfolio discussed. Furthermore, when considering right tail risk for $\alpha > 0.99$, it may be appropriate to model the risk using a t-distribution with 3 to 10 degrees of freedom, however, this is dependent on an entity's goals and intentions with the portfolio.

Appendix 1 – Additional Information

Section 1 - Data Retrieval & Processing

The daily stock prices of NVIDIA Corporation (NVDA) and Advanced Micro Devices Inc (AMD) were collected from Yahoo Finance for the period of 1st January 2010 to 1st June 2024. Then, the daily adjusted closing prices (denoted S_t) were used to calculate the daily log-returns, $X_t = \ln(S_t) - \ln S_{t-1}$. As X_t cannot be calculated for the first date in the dataset (as the day prior doesn't exist), an NA value was produced and thus the first row of data was removed. Finally, daily (potential) losses were calculated using $L_t = 1 - \exp\{X_t\}$.

For the remainder of the report, denote $X_{A,t}, X_{N,t}$ as the return rates at time t for AMD and NVDA respectively, and $L_{A,t}$ and $L_{N,t}$ as their potential losses. Also, define X_A, X_N as the of returns of AMD and NVDIA respectively. Similarly, define L_A, L_N as the losses of $1 investment into AMD and NVDA respectively. Finally, confidence levels of $\alpha = 95\%, 99\%$ and 99.5% will be used throughout the investigation, and degrees of freedom $\nu = 3, 10, 10^4$ for the t-distribution copula.

Section 2 - Introducing Correlation into Random Samples

If $\boldsymbol{\Sigma} = \boldsymbol{A}\boldsymbol{A}^T = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$, and Z_1, Z_2 are iid $N(0,1)$ random variables, then by Cholesky's decomposition we get that:

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \boldsymbol{A}\begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \rho & \sqrt{1-\rho^2} \end{pmatrix}\begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \Rightarrow \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \sim N(0, \boldsymbol{\Sigma})$$

Table 1 – Empirical Risk Results

Let $L = \frac{1}{2}\{L_N + L_A\}$.

α	$VaR_\alpha(L_N)$	$VaR_\alpha(L_A)$	$VaR_\alpha(L)$	$ES_\alpha(L_N)$	$ES_\alpha(L_A)$	$ES_\alpha(L)$	Tail Dependence
0.95	0.0410933	0.0514770	0.0437543	0.0604674	0.0766516	0.0624285	0.4505495
0.99	0.0701235	0.0922033	0.0761931	0.0937298	0.1232152	0.0930898	0.2432432
0.995	0.0821780	0.1182233	0.0867645	0.1108820	0.1442557	0.1052372	0.1052632

Table 2 – Portfolio Risk & Tail Dependence (Independence)

Independence Copula			
α	$VaR_\alpha(L)$	$ES_\alpha(L)$	Tail Dependence
0.95	0.03247059	0.04678313	0.03703704
0.99	0.05485390	0.07312291	0
0.995	0.06695198	0.08538291	0

* Note that the tail dependence of the independence copula was non-zero under specific seeds, however for seed 19 the following results were given.

Table 3 - Joint Risk & Tail Dependence (Gaussian)

Gaussian Copula			
α	$VaR_\alpha(L)$	$ES_\alpha(L)$	Tail Dependence
0.95	0.03993386	0.05943555	0.24590164
0.99	0.07334160	0.09345180	0.11290323
0.995	0.08689266	0.10797342	0.07692308

Table 4 - Joint Risk & Tail Dependence (t, df = 3)

t-Copula ($\nu = 3$)			
α	$VaR_\alpha(L)$	$ES_\alpha(L)$	Tail Dependence
0.95	0.03950501	0.06074777	0.4065041
0.99	0.07487680	0.09607596	0.3125000
0.995	0.09114205	0.10920308	0.2380952

Table 5 - Joint Risk & Tail Dependence (t, df = 10)

t-Copula ($\nu = 10$)			
α	$VaR_\alpha(L)$	$ES_\alpha(L)$	Tail Dependence
0.95	0.03992407	0.06064204	0.2936508
0.99	0.07257050	0.09927999	0.2500000
0.995	0.08924661	0.11895069	0.2413793

Table 6 - Joint Risk & Tail Dependence (t, df = 10^4)

t-Copula ($\nu = 10^4$)			
α	$VaR_\alpha(L)$	$ES_\alpha(L)$	Tail Dependence
0.95	0.03983469	0.05934616	0.2398374
0.99	0.07336419	0.09276822	0.1129032
0.995	0.08543139	0.10700723	0.0800000

Table 7 - Joint Risk & Tail Dependence (Comonotonicity)

Comonotonicity Copula			
α	$VaR_\alpha(L)$	$ES_\alpha(L)$	Tail Dependence
0.95	0.04578012	0.06683089	1
0.99	0.08036585	0.10545081	1
0.995	0.09618691	0.12488818	1

Table 8 - Joint Risk & Tail Dependence (Counter Monotonicity)

Counter Monotonicity Copula			
α	$VaR_\alpha(L)$	$ES_\alpha(L)$	Tail Dependence
0.95	0.002272737	0.003850475	0
0.99	0.005821807	0.007145182	0
0.995	0.006974384	0.007648306	0

Derivation 1 - Proof of Rank Correlation Formula

$$\begin{aligned}\rho_\tau(X,Y) &= \mathbb{E}\left[\operatorname{sign}\left(\left(X-\tilde{X}\right)\times\left(Y-\tilde{Y}\right)\right)\right]\\ &= \Pr\left(\left(X-\tilde{X}\right)\times\left(Y-\tilde{Y}\right)=1\right)-\Pr\left(\left(\left(X-\tilde{X}\right)\times\ \left(Y-\tilde{Y}\right)=-1\right)\right)\\ &= \Pr(\text{Same Sign})-\Pr(\text{Differnt Sign})\\ &= \Pr(\text{Concordance})-\Pr(\text{Discordance})\end{aligned}$$

From here, we can make the estimate:

$$\rho_\tau \approx \frac{(\#\, Concordance\ Pairs) - (\#\, Discordant\ Pairs)}{\#\, Pairs}$$

Derivation 2 - Expanding Sklar's Theorum

Sklar’s theorem states that:

$$F_X(x_A, x_N) = C(F_A(x_A), F_N(x_N))$$

Then given $\binom{U_1}{U_2} \sim C_{\text{desired}}$, we can evaluate at $x_A = F_A^{-1}(U_1), x_N = F_N^{-1}(U_2)$, to get:

$$F_X\left(F_A^{-1}(U_1), F_N^{-1}(U_2)\right) = C(U_1, U_2)$$

Thus, $\boldsymbol{X_i} = \binom{X_A}{X_N}$ are our simulated samples under the desired dependence assumption.

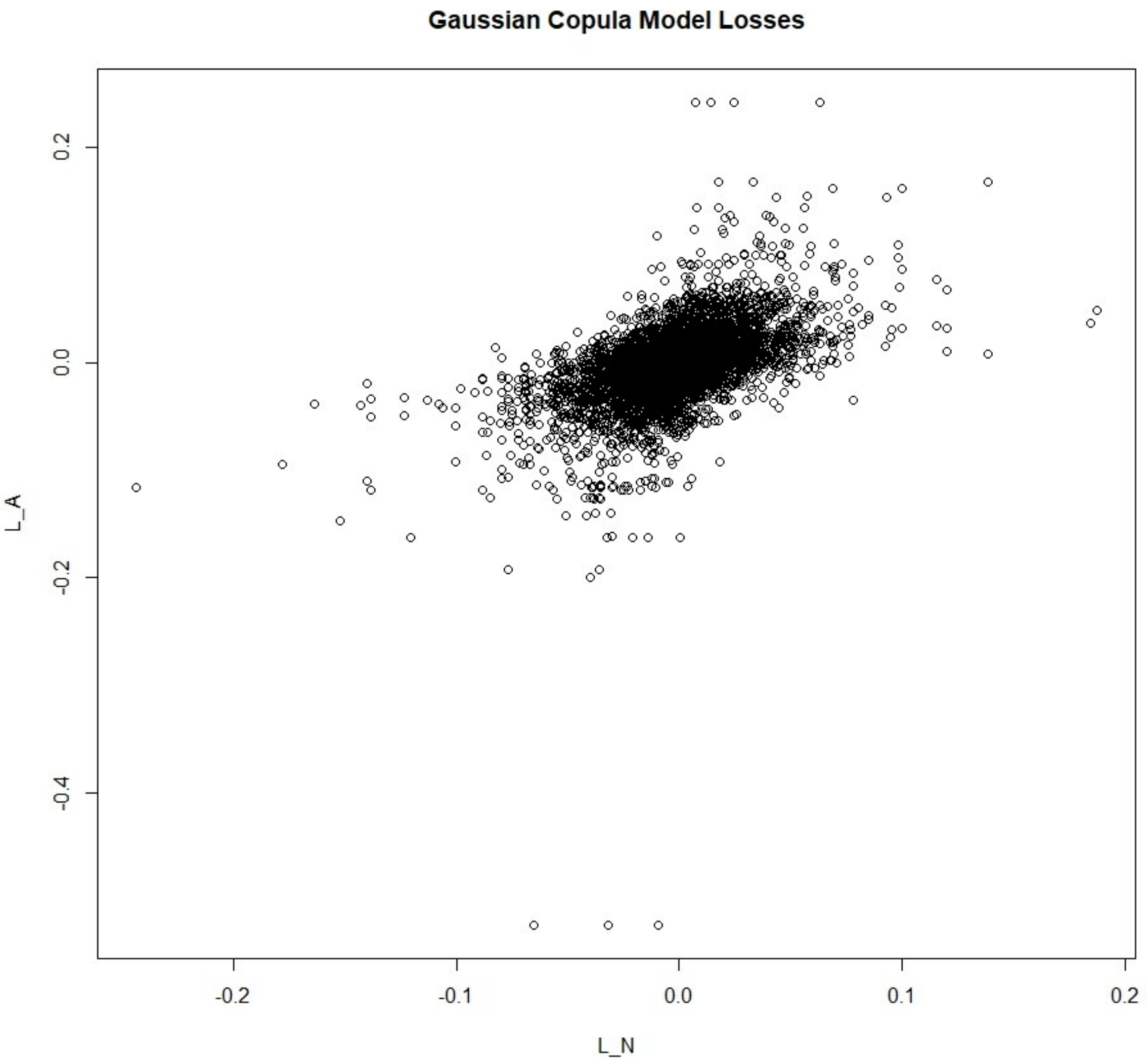

Figure 5 - Gaussian Copula Losses

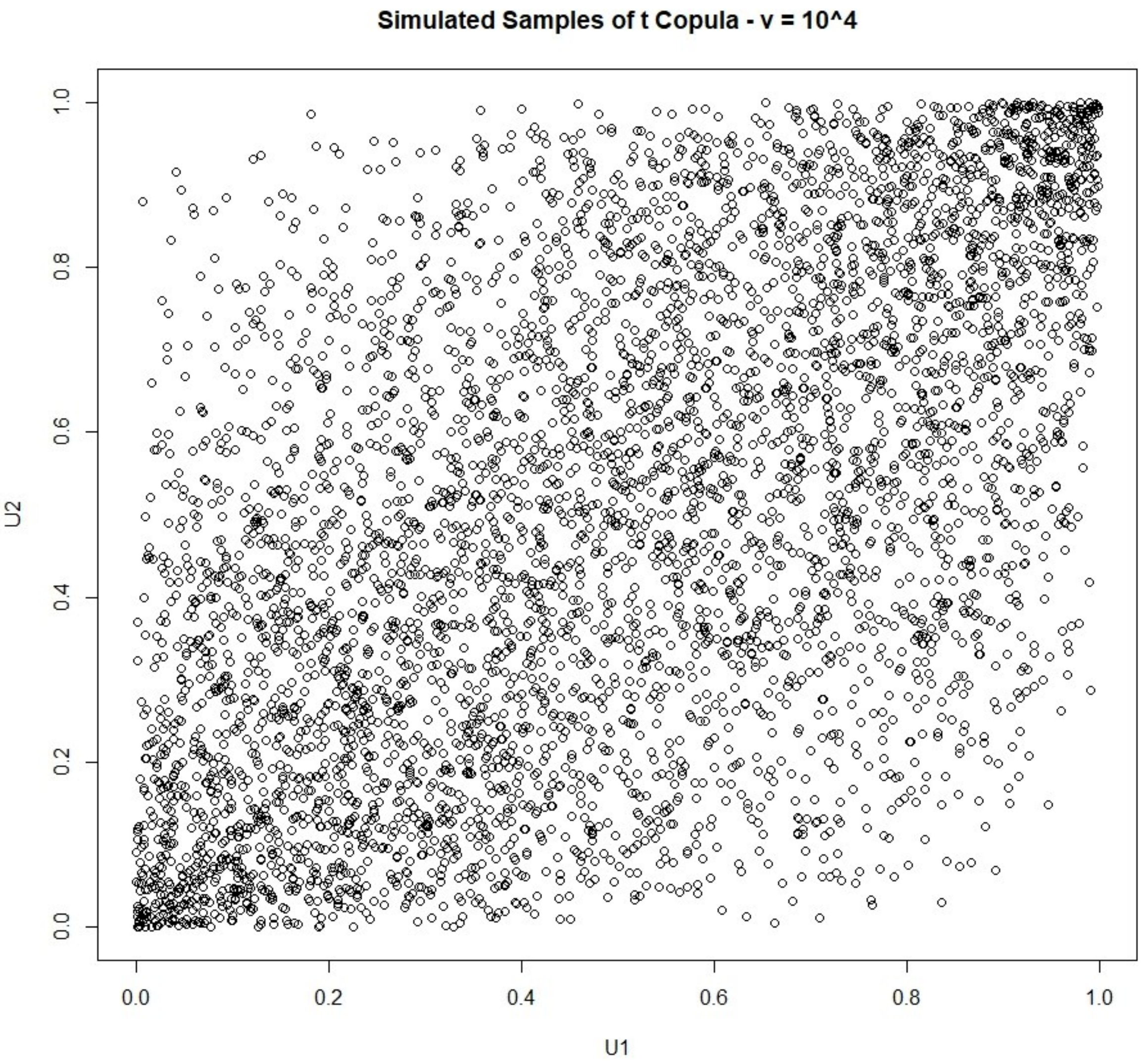

Figure 6 - t Copula Samples ($v = 10^4$)

Appendix 2 – Programming Code

```
# Install and load packages

# install.packages("dplyr")

# install.packages("invgamma")

# library(dplyr)

library(invgamma)


#                              Data                              Preparation
################################################################

# Import Data

setwd("C:/Users/dhwan/OneDrive/Documents/MyFiles/ACTL3301/")

# setwd("~/Desktop/3301 Ass1")


set.seed(19)


amd <- read.csv("AMD.csv")

nvda <- read.csv("NVDA.csv")


# Order

amd$Date <- as.Date(amd$Date, format="%Y-%m-%d")

nvda$Date <- as.Date(nvda$Date, format="%Y-%m-%d")


amd <- amd[order(amd$Date), ]

nvda <- nvda[order(nvda$Date), ]


# Daily Return

amd$Xt <- c(NA, diff(log(amd$Adj.Close)))

nvda$Xt <- c(NA, diff(log(nvda$Adj.Close)))


amd <- na.omit(amd)

nvda <- na.omit(nvda)
```

```
# Potential losses
amd$Lt <- 1 - exp(amd$Xt)
nvda$Lt <- 1 - exp(nvda$Xt)

# View(amd)
# View(nvda)

# Task 1 ######################################################################

alpha <- c(0.95, 0.99, 0.995)

l1 <- amd$Lt
l2 <- nvda$Lt

# Function to calculate VaR as the largest loss less than the given quantile
calculate_var <- function(losses, a) {
 threshold <- quantile(losses, a, type = 1)
 min(losses[losses >= threshold])
}

# Function to calculate ES
calculate_es <- function(losses, var, a, cdf) {
 F_var <- cdf(var)
 delta <- abs(F_var - a)
 # cat("F_var:", F_var, "a:", a, "delta:", delta, "\n")
 ((var * delta) / (1 - a)) + mean(losses[losses >= var])
}

var_95 <- calculate_var(l1, 0.95)
var_99 <- calculate_var(l1, 0.99)
var_995 <- calculate_var(l1, 0.995)
```

```
es_95 <- calculate_es(l1, var_95, 0.95, ecdf(l1))
es_99 <- calculate_es(l1, var_99, 0.99, ecdf(l1))
es_995 <- calculate_es(l1, var_995, 0.995, ecdf(l1))

amd_risk <- data.frame(
  Alpha = alpha,
  VaR = c(var_95, var_99, var_995),
  ES = c(es_95, es_99, es_995)
)


# Calculations for NVDA
var_95 <- calculate_var(l2, 0.95)
var_99 <- calculate_var(l2, 0.99)
var_995 <- calculate_var(l2, 0.995)

es_95 <- calculate_es(l2, var_95, 0.95, ecdf(l2))
es_99 <- calculate_es(l2, var_99, 0.99, ecdf(l2))
es_995 <- calculate_es(l2, var_995, 0.995, ecdf(l2))

nvda_risk <- data.frame(
  Alpha = alpha,
  VaR = c(var_95, var_99, var_995),
  ES = c(es_95, es_99, es_995)
)

# Empirical Joint ##################################################################
l <- 0.5 * (l1 + l2)

# Calculations for NVDA
```

```
var_95 <- calculate_var(l, 0.95)
var_99 <- calculate_var(l, 0.99)
var_995 <- calculate_var(l, 0.995)

es_95 <- calculate_es(l, var_95, 0.95, ecdf(l))
es_99 <- calculate_es(l, var_99, 0.99, ecdf(l))
es_995 <- calculate_es(l, var_995, 0.995, ecdf(l))

e_risk <- data.frame(
  Alpha = alpha,
  VaR = c(var_95, var_99, var_995),
  ES = c(es_95, es_99, es_995)
)

# Task 2
###########################################################################

# Calculations for Portfolio

u1 <- runif(5000, 0, 1)
u2 <- runif(5000, 0, 1)

U1_independent <- u1
U2_independent <- u2

X1 <- sapply(U1_independent, function(u) calculate_var(nvda$Xt, u))
X2 <- sapply(U2_independent, function(u) calculate_var(amd$Xt, u))

l1_indep <- 1 - exp(X1)
l2_indep <- 1 - exp(X2)
```

```
l = 0.5 * (l1_indep + l2_indep)

var_95 <- calculate_var(l, 0.95)
var_99 <- calculate_var(l, 0.99)
var_995 <- calculate_var(l, 0.995)

es_95 <- calculate_es(l, var_95, 0.95, ecdf(l))
es_99 <- calculate_es(l, var_99, 0.99, ecdf(l))
es_995 <- calculate_es(l, var_995, 0.995, ecdf(l))

portfolio_risk <- data.frame(
 Alpha = alpha,
 VaR = c(var_95, var_99, var_995),
 ES = c(es_95, es_99, es_995)
)

# Print
e_risk
amd_risk
nvda_risk
portfolio_risk


# Task 3 ###########################################################################
# Calculate Kendall's Tau
calculate_kendalls_tau <- function(L1, L2) {
 n <- length(L1)
 num_conc <- 0
 num_disc <- 0
```

```
  for (i in 1:(n-1)) {
    for (j in (i+1):n) {
      if ((L1[i] - L1[j]) * (L2[i] - L2[j]) > 0) {
        num_conc <- num_conc + 1
      } else if ((L1[i] - L1[j]) * (L2[i] - L2[j]) < 0) {
        num_disc <- num_disc + 1
      }
    }
  }

  tau <- (num_conc - num_disc) / choose(n, 2)
  return(tau)
}

kendalls_tau <- calculate_kendalls_tau(amd$Lt, nvda$Lt)

kendalls_tau

# Calculate Spearman's Rho
ecdf_1 = ecdf(l1)
ecdf_2 = ecdf(l2)

F_l1 = ecdf_1(l1)
F_l2 = ecdf_2(l2)

spearmans_rho <- cor(F_l1, F_l2)

correlation <- cor(l1, l2)
correlation

# Estimate Spearman's Rho
#estimate_spearmans_rho <- function(L1, L2) {
```

```
#rank_L1 <- rank(L1)
#rank_L2 <- rank(L2)
#n <- length(L1)
#diff_squared <- sum((rank_L1 - rank_L2)^2)
#cat("diff_squared:", diff_squared, "n:", n)

#rho <- 1 - (6 * diff_squared) / (n * (n^2 - 1))
#return(rho)
#}

#spearmans_rho_estimate <- estimate_spearmans_rho(amd$Lt, nvda$Lt)

# Validation Result
#spearmans_rho_validate <- cor(amd$Lt, nvda$Lt, method = "spearman")

# Print
spearmans_rho
#spearmans_rho_estimate
#spearmans_rho_validate

# Task 4 ###########################################################################

# Simulate Gaussian Copula
# Simulate m.v. Normal(0, rho)

# Transform u1, u2 into z1, z2 N(0,1) r.v.
z1 <- qnorm(u1)
z2 <- qnorm(u2)

# Consider correlation
```

```
rho <- cor(amd$Xt, nvda$Xt)

y1 <- z1
y2 <- (rho * z1) + (sqrt(1 - rho^2) * z2)

# U = F(Y)
U1 <- pnorm(y1)
U2 <- pnorm(y2)

U1_gaussian <- U1
U2_gaussian <- U2

# Plot
plot(U1, U2, main="Simulated Samples of Gaussian Copula", xlab="U1", ylab="U2")

# Skylar's theorem
X1 <- sapply(U1_gaussian, function(u) calculate_var(nvda$Xt, u))
X2 <- sapply(U2_gaussian, function(u) calculate_var(amd$Xt, u))

L1_gauss <- 1 - exp(X1)
L2_gauss <- 1 - exp(X2)

plot(L1_gauss, L2_gauss, main="Gaussian Copula Model Losses", xlab="L_N", ylab="L_A")

#
#print(X1)
#print(X2)

l = 0.5 * ((1 - exp(X1)) + (1 - exp(X2)))

# Calculations for Portfolio
# ecdf_l <- rank(l) / length(l)
```

```
var_95 <- calculate_var(l, 0.95)
var_99 <- calculate_var(l, 0.99)
var_995 <- calculate_var(l, 0.995)

es_95 <- calculate_es(l, var_95, 0.95, ecdf(l))
es_99 <- calculate_es(l, var_99, 0.99, ecdf(l))
es_995 <- calculate_es(l, var_995, 0.995, ecdf(l))

sim_portfolio_risk <- data.frame(
  Alpha = alpha,
  VaR = c(var_95, var_99, var_995),
  ES = c(es_95, es_99, es_995)
)

sim_portfolio_risk



# Task 5 ######################################################################
# Define the parameters
params <- list(
  list(shape = 0.5 * 3, rate = 0.5 * 3, df = 3),
  list(shape = 0.5 * 10, rate = 0.5 * 10, df = 10),
  list(shape = 0.5 * 10000, rate = 0.5 * 10000, df = 10000)
)

# Generate W values
W <- lapply(params, function(p) 1 / rgamma(5000, p$shape, p$rate))
```

```
# Simulate y values
y_sim <- lapply(W, function(W_i) {
  list(
    y1 = sqrt(W_i) * y1,
    y2 = sqrt(W_i) * y2
  )
})

# Simulate U values
U <- lapply(seq_along(params), function(i) {
  list(
    U1 = pt(y_sim[[i]]$y1, params[[i]]$df),
    U2 = pt(y_sim[[i]]$y2, params[[i]]$df)
  )
})

U1_t3 <- U[[1]]$U1
U2_t3 <- U[[1]]$U2

U1_t10 <- U[[2]]$U1
U2_t10 <- U[[2]]$U2

U1_t10000 <- U[[3]]$U1
U2_t10000 <- U[[3]]$U2

# Define a function to calculate risk metrics
calculate_risk_metrics <- function(U1, U2, Xt_n, Xt_amd) {
  list(
    X1 = sapply(U1, function(u) calculate_var(Xt_n, u)),
    X2 = sapply(U2, function(u) calculate_var(Xt_amd, u)),
    l = 0.5 * ((1 - exp(sapply(U1, function(u) calculate_var(Xt_n, u)))) + (1 - exp(sapply(U2, function(u) calculate_var(Xt_amd, u)))))
```

```
 )
}

# Calculate risk metrics for each parameter set
risk_metrics <- lapply(seq_along(params), function(i) {
 metrics <- calculate_risk_metrics(U[[i]]$U1, U[[i]]$U2, nvda$Xt, amd$Xt)

 list(
  VaR = sapply(c(0.95, 0.99, 0.995), function(alpha) calculate_var(metrics$l, alpha)),
  ES = sapply(c(0.95, 0.99, 0.995), function(alpha) calculate_es(metrics$l, calculate_var(metrics$l, alpha), alpha, ecdf(metrics$l)))
 )
})

# Create data frames for each parameter set
sim_portfolio_risk <- lapply(seq_along(params), function(i) {
 data.frame(
  Alpha = c(0.95, 0.99, 0.995),
  VaR = risk_metrics[[i]]$VaR,
  ES = risk_metrics[[i]]$ES
 )
})

# Plot results
plot_titles <- c("Simulated Samples of t Copula - v = 3", "Simulated Samples of t Copula - v = 10", "Simulated Samples of t Copula - v = 10^4")
lapply(seq_along(params), function(i) {
 plot(U[[i]]$U1, U[[i]]$U2, main=plot_titles[i], xlab="U1", ylab="U2")
})

# Display results
sim_portfolio_risk
```

```
# Task 6
###################################################################
#####

# Comonotonicity copula

U1 <- u1

U2 <- u1


U1_co <- U1

U2_co <- U2


plot(U1_co, U2_co, main="Simulated Samples of (U1, U2)", xlab="U1", ylab="U2")


X1 <- sapply(U1, function(u) calculate_var(nvda$Xt, u))

X2 <- sapply(U2, function(u) calculate_var(amd$Xt, u))


l = 0.5 * ((1 - exp(X1)) + (1 - exp(X2)))


var_95 <- calculate_var(l, 0.95)

var_99 <- calculate_var(l, 0.99)

var_995 <- calculate_var(l, 0.995)


es_95 <- calculate_es(l, var_95, 0.95, ecdf(l))

es_99 <- calculate_es(l, var_99, 0.99, ecdf(l))

es_995 <- calculate_es(l, var_995, 0.995, ecdf(l))


sim_portfolio_risk <- data.frame(

  Alpha = c(0.95, 0.99, 0.995),

  VaR = c(var_95, var_99, var_995),

  ES = c(es_95, es_99, es_995)
```

```
)

sim_portfolio_risk

# Countermonotonicity copula
U1 <- u1
U2 <- 1 - U1

U1_counter <- U1
U2_counter <- U2

plot(U1_counter, U2_counter, main="Simulated Samples of (U1, U2)", xlab="U1", ylab="U2")

X1 <- sapply(U1, function(u) calculate_var(nvda$Xt, u))
X2 <- sapply(U2, function(u) calculate_var(amd$Xt, u))

l = 0.5 * ((1 - exp(X1)) + (1 - exp(X2)))

var_95 <- calculate_var(l, 0.95)
var_99 <- calculate_var(l, 0.99)
var_995 <- calculate_var(l, 0.995)

es_95 <- calculate_es(l, var_95, 0.95, ecdf(l))
es_99 <- calculate_es(l, var_99, 0.99, ecdf(l))
es_995 <- calculate_es(l, var_995, 0.995, ecdf(l))

sim_portfolio_risk <- data.frame(
  Alpha = c(0.95, 0.99, 0.995),
  VaR = c(var_95, var_99, var_995),
  ES = c(es_95, es_99, es_995)
)
```

```
sim_portfolio_risk

# Task 7
#######################################################################

# Pr(A | B) = Pr (AB) / Pr(B)

calc_tail_dep <- function(A, B, alpha) {
  sum(A[B > alpha] > alpha) / sum(B > alpha)
}


alpha_values <- c(0.95, 0.99, 0.995)

tail_dep_indep <- sapply(alpha_values, function(alpha) {
  calc_tail_dep(U1_independent, U2_independent, alpha)
})

tail_dep_indep

tail_dep_gaussian <- sapply(alpha_values, function(alpha) {
  calc_tail_dep(U1_gaussian, U2_gaussian, alpha)
})

tail_dep_gaussian

tail_dep_t3 <- sapply(alpha_values, function(alpha) {
  calc_tail_dep(U1_t3, U2_t3, alpha)
})

tail_dep_t3
```

```
tail_dep_t10 <- sapply(alpha_values, function(alpha) {
  calc_tail_dep(U1_t10, U2_t10, alpha)
})

tail_dep_t10

tail_dep_t10000 <- sapply(alpha_values, function(alpha) {
  calc_tail_dep(U1_t10000, U2_t10000, alpha)
})

tail_dep_t10000

tail_dep_co <- sapply(alpha_values, function(alpha) {
  calc_tail_dep(U1_co, U2_co, alpha)
})

tail_dep_co

tail_dep_counter <- sapply(alpha_values, function(alpha) {
  calc_tail_dep(U1_counter, U2_counter, alpha)
})

tail_dep_counter

X1_empirical <- nvda$Lt
X2_empirical <- amd$Lt

U1_empirical <- ecdf(X1_empirical)(X1_empirical)
U2_empirical <- ecdf(X2_empirical)(X2_empirical)

tail_dep_empirical <- sapply(alpha_values, function(alpha) {
  calc_tail_dep(U1_empirical, U2_empirical, alpha)
```

```r
})

tail_dep_empirical

## IGNORE
####################################################################
####

L <- 0.5 * (nvda$Lt + amd$Lt)

export <- data.frame(
  date = nvda$Date,
  X1 = nvda$Xt,
  X2 = amd$Xt,
  L1 = nvda$Lt,
  L2 = amd$Lt,
  L = L
)

write.csv(export, "z5421168_Dhwanish_Kshatriya", row.names = FALSE)
```