UNIVERSITY
OF OSLO

Master's thesis

Portfolio management with vine copulas

Heming Smedsrud Aldrin

Data Science
30 ECTS study points

Department of Mathematics
Faculty of Mathematics and Natural Sciences

Autumn 2023

Heming Smedsrud Aldrin

Portfolio management with vine copulas

Supervisor:
Ingrid Hobæk Haff

Abstract

The purpose of portfolio management is to minimise risk while maximising returns. However, traditional portfolio construction often underestimates the true risk. This thesis captures this risk, using vine copulas to better model the non-linear dependencies, by decoupling the dependence structure from the marginal distributions. This approach is shown through an empirical study of a portfolio of stock market indexes for the G7 countries: Canada, France, Germany, Italy, Japan, UK and USA.

Previous studies have focused on the the C- and D-vines, however, this thesis also considers the more flexible R-vine. Different from other approaches that find the optimal weights for a longer period, this thesis finds the optimal weights for the next time step of a week. By converting all currencies to USD, the model solely focuses on the co-movements between the markets.

This thesis finds the optimal portfolio composition for the risk and excepted return for the log returns of the next week. Any mean and variance structure can be captured, by fitting an ARMA-EGARCH model with skewed Student's t-distributed residuals on the log returns of the stock market indexes. Then the residuals are used to fit vine copulas to capture non-linear dependencies. By fitting and simulating from different pair copula structures in addition to a multivariate normal distribution, this thesis aims to compare the capabilities of capturing non-linear dependencies. After converting back to log returns, a distribution for the portfolio as a whole can be estimated. And an efficiency frontier is created to show the optimal choices for the portfolio, showing the modelled risk.

The results suggest that not using copula does not give widely different portfolio compositions, rather that the risk is vastly underestimated. Due to time limitations, it was not possible to apply time varying vine copulas or other GARCH models, to compare with this model. These could have given better results, why I recommend future research to focus on these.

Contents

List of Figures

List of Tables

Preface

I would like give a special thank you to my supervisor, Ingrid, for all the help with understanding the dept of this topic. I would also say that I really appreciated the topic and the task, even though it felt overwhelming at times.

I would also like to thank my close family and friends who have helped me with this thesis directly and indirectly. You all know who you are, I could not have done it without you.

Chapter 1

Introduction

1.1 Relevance of the topic

In the world of portfolio management, investors face the challenge of optimising asset allocation to achieve desired investment objectives. Choosing different assets to invest in, creates a portfolio with an expected return and risk. Markowitz (1952) recognised that investors should seek to maximise the return for a given level of risk and established modern portfolio management by introducing the mean-variance framework. Its idea is that the risk and return of an individual asset should be evaluated based on its contribution to the portfolio's collective risk and return, rather than in isolation. Since then, financial markets became more interconnected yet unpredictable. Traditional portfolio construction methods, grounded in historical returns and variance, are limited in capturing the true essence of market behaviours, particularly during periods of financial distress. Moreover, the assumption of normality in asset returns, a common simplification in traditional models, frequently falls short in the face of real-world financial data, characterised by skewness, kurtosis, and anomalies like fat tails.

In the last 20 years, the concept of copulas has received more attention, deeming it possible to decouple the dependencies from the marginals. This makes it easier to model non-linear dependencies. However, previous empirical studies have focused on the C- and D-vines, while neglecting the more general R-vine. This lack will be filled in this thesis, allowing for a broader search of pair copula constructions with a possibly more robust vine. Furthermore, some studies do not consider the impact local currencies can have on dependencies between the stock indexes. Therefore, in this thesis we are removing currency effects, by transforming all currencies into USD, leaving us with a more direct evaluation of the market dependencies.

1.2 Research Question

Interest in understanding the intricacies of financial data led this thesis to explore the stock market indexes of the G7 countries: Canada, France, Germany, Italy, Japan, UK and USA. Inspired by the work of Nguyen and Liu (2023), this thesis tries to address the use of vines to model dependencies with a specific focus on trying the more flexible R-vines and comparing its performance with the other vines and a non vine alternative. This is done through the task of optimising the portfolio weights for the next time step, which is one week in this thesis.

The process involves applying an ARMA-EGARCH model on the time series with a skewed Student's t-distribution for the residuals. Then the residuals are fit on the

different vines to find any non-linear dependencies. Breaking down these parts of the model and analysing each part individually gives insights in how the processes interact. The ARMA-EGARCH models try to capture many characteristics. Financial data typically have little ARMA structure, however, some structure might be found. The GARCH model can then find time-varying volatility, with the EGARCH variant also allowing for different reaction to positive and negative chocks. This is relevant for financial data that often reacts stronger when negative chocks occur. While at the same time assuming skewed Student's t-distributions for the residuals, the ARMA-EGARCH model can better fit the skewed and heavy tailed nature, which is often found in financial data. All these effects are found to be significant, in the empirical study, each one building upon each other the structure for the marginal distribution for the log return series.

Financial models are often criticised for underestimating risk. This is why this thesis, on top of describing each marginal series, uses copulas to model non-linear dependencies. Which is particularly important when dealing with extreme events, where the tail dependence is of great importance.

By combining the ARMA-EGARCH and the copula methods, this thesis aims to better capture the behaviour in the tails, better representing the true risk. Especially focusing on how the R-vine performs differently from other vines.

After simulating the values for the next week, the problem of finding an optimal portfolio is explored by using efficiency frontiers, where CVaR is the measure of risk. In addition to the different vines, a model using multidimensional normal distribution for the dependence is also created. This is to compare how the use of vines to include non-linear dependence affect the risk assesment.

1.3 Outline

The thesis is organised as follows:

chapter 2 covers the methods used in this thesis. At first, the topic of financial data is introduced including a brief insight to portfolio management. To facilitate the understanding of the later carried out empirical study, all relevant mathematical measures and methods are explained and defined, including a short introduction about when and why the respective measures are useful. The chapter closes with a brief explanations for reasoning during implementation.

chapter 3 asserts the empirical study of this thesis. It starts with an overview of the data and continuous with the analysis of the implemented model steps. Then, when the models are fit on different vines, efficiency frontiers are presented to compare the approaches..

chapter 4 sums up the finding, presents some final remarks on limitations of this thesis and suggestions for future research on this topic.

appendix A features additional results.

appendix B consists of the R-code used in the work of this thesis.

Chapter 2

Method

To find the optimal portfolio compositions for an empirical study on the stock market indexes of the G7-countries, it is important to understand some basic principals that are common when working with financial data. However, these mathematical tools, deployed in Chapter 3, are not limited to a financial setting. The combination of several mathematical approaches introduced in this chapter are used to fit the most realistic model. Lastly, this chapter will focus on the challenges of a robust implementation of this scale, highlighting choices not suitable to be mentioned in the empirical part.

2.1 Financial data

This thesis, centred on financial data represented by a portfolio of stock market indexes, necessitates an introduction to key concepts and terminology prevalent in statistical finance.

2.1.1 Portfolio management

Investors should want to highest expected return as possible while also taking the minimum risk. As a stated by Markowitz (1952), an investor should want to maximise the expected return given a set amount of risk, or equally, minimise the risk given some expected return. Optimising a portfolio is then about finding the optimal weighting of different assets in a way that does not violate this criteria. Lets therefore first define a portfolio as follows.

Definition 1. *A* portfolio *is a collection of financial assets, where the value of the portfolio is the sum of the value of its assets. For the weights w_i of a portfolio, the following must hold for a portfolio of d assets*

$$\sum_{i}^{d} w_i = 1 \tag{2.1}$$

The expectation of a portfolio can then be defines as

Definition 2. *The expected return for a d-dimensional portfolio $\mathbf{R}$ of stock indexes R is the weighted sum of its parts:*

$$\mathrm{E}\left[\mathbf{R}\right] = w_1 \mathrm{E}\left[R_1\right] + \ldots + w_d \mathrm{E}\left[R_d\right], \tag{2.2}$$

where w_i for $i = 1, \ldots, d$, is the weights for each market and $\sum_{i=1}^{d} w_i = 1$.

Financial markets exhibit intricate relationships and dependencies that challenge traditional modelling approaches. Complex dependencies among assets, often nonlinear and time-varying, defy simplistic assumptions of independence. Moreover, the presence of skewed and heavy-tailed distributions can distort risk assessments, leading to sub optimal portfolio decisions. Those are the challenges this thesis tries to solve, and tools trying to do so are presented through the rest of this chapter.

2.1.2 Returns and log returns

Working directly with price data has certain drawbacks. In contrast, using returns offers a significant advantage as it normalises price changes, making them relative rather than absolute. This is important because a \$5 change in a \$10 stock is much more significant than a \$5 change in a \$100 stock. Returns are defined as follows

Definition 3. Returns *are the change in price from* p_{t-1} *to* p_t *expressed as:*

$$\textit{returns} = r_t = \frac{p_t - p_{t-1}}{p_{t-1}} = \frac{p_t}{p_{t-1}} - 1 \tag{2.3}$$

Log returns are the difference of the natural logarithms of two neighbouring points which. It can also be written in terms of the return and is:

$$\text{log returns} = \log(p_t) - \log(p_{t-1}) = \log(p_t/p_{t-1}) = \log(r_t + 1). \tag{2.4}$$

This means that log-returns over consecutive periods can be added to find the total return over the entire period. This additive nature makes modelling and interpreting more straightforward.

Notice however, when using log returns in this thesis, the mean μ is subtracted, shifting the series to be around zero:

$$\text{log returns} = \log(p_t) - \log(p_{t-1}) - \mu \tag{2.5}$$

2.1.3 Performance measures

In the financial context, there are multiple methods for assessing risk and performance. For the purposes of this thesis, a measure that offers a detailed view of the downside risk in investment portfolios is required. Such measures should effectively capture and account for tail risk.

Value at Risk (VaR) is a percentile of a distribution used as a statistical metric in finance to assess the potential maximum loss of a portfolio or investment over a specified time horizon with a given level of confidence. Its strength is its simplicity of being a single numerical value that represents the estimated amount that could be lost with a certain probability. This value can be expressed both as a price unit or as a percentage, quantifying the magnitude of the potential loss and making it easy to compare. VaR is therefore a useful measure that for example can be used as an estimate of how much asset is needed to cover potential losses.

The α VaR is defined as the loss where there is $\alpha\%$ chance of experiencing a loss of this size or larger, and $(100 - \alpha)\%$ for a smaller loss. This shows that it is the same as the $(1 - \alpha) \cdot 100\%$-percentile. Typically, the probabilities used for VaR are 1% or 5%, and while the time frame is usually a day, other values are also used.

Definition 4. *The* value at risk (VaR) *is for a continuous random variable* X

$$VaR_\alpha(X) = -F_X^{-1}(\alpha) = F_Y^{-1}(1-\alpha), \tag{2.6}$$

where X *is the profit,* $Y := -X$ *is the loss,* $F_X(x)$ *and* $F_Y(y)$ *are the cumulative distributions of* X *and* Y *respectively, and* F^{-1} *is the inverse cumulative distribution.*

There are several methods to calculate VaR, where using historical data is the simplest way to do this. However, Monte Carlo simulation is also often used and is what is used in this thesis.

The Conditional Value at Risk (CVaR), also known as Expected shortfall, provides a more comprehensive assessment of potential losses than VaR, by taking the expectation of the loss beyond this threshold. By conditioning on the occurrence of an event that surpasses the VaR level, CVaR offers better insights into the severity of extreme events and tail risk.

The computation of CVaR is derived from VaR and holds the same assumptions. Its mathematical formula is:

$$CVaR_\alpha(X) = -\operatorname{E}[X|X < -VaR_\alpha(X)] = \operatorname{E}[Y|Y > VaR_\alpha(X)] \tag{2.7}$$

An example of VaR and CVaR is shown in the in histogram in Figure 2.1. The long tail for the loss shows how the VaR value can give a misleading impression of the risk, while the CVaR captures the extreme values.

While it is naturally for an investor to want to minimise risk, they should also want to maximise expected returns. While CVaR is a good measure for risk, it completely neglects any potential gains. The idea of return-to-CVaR is then to indicate the return to risk ratio, which is:

$$\text{return-to-CVaR} = \frac{\operatorname{E}[X]}{CVaR_\alpha(X)}, \tag{2.8}$$

where X could be both the return and the log returns.

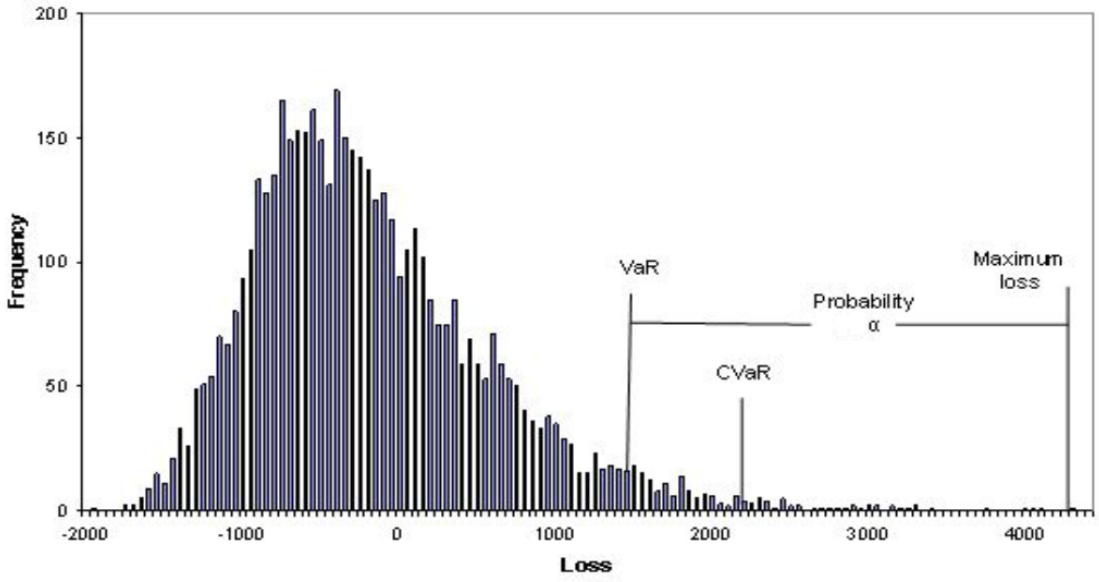


Figure 2.1: Histogram of potential losses. (Figure taken from Webby 2009)

2.1.4 Markowitz - Mean-variance

The Markowitz Mean-Variance concept, introduced by Markowitz in 1952, forms the foundation of modern portfolio theory. It is a framework for optimising a portfolio of assets. The idea is that the risk of an asset should not be assessed by itself, but rather for the whole portfolio. Where variance is used as the measure for risk (Markowitz 1952).

This idea comes from the fact that for a group of d assets that are not perfectly correlated, having the same expected return μ and variance σ^2, an investor should diversify instead of investing only in one asset. This is because the expectation is the same but the variance is σ^2/d for the diversified portfolio. Then, under the assumption that investors are risk averse, investors will only take on more risk if compensated by higher expected return.

The assumptions for the mean-variance method proposed by Markowitz, was linear dependence between assets and elliptical distributions, where the multivariate normal distribution is the natural choice. Then the returns matrix $\mathbf{R}$, for d assets can be written as:

$$\mathbf{R} \sim MVN_d(\mu, \mathbf{\Sigma}), \tag{2.9}$$

where μ d-vector containing the expected values and $\mathbf{\Sigma}$ the covariance matrix.

The concept of an efficient frontier is then the set of weights for a portfolio that gives the lowest risk given an expected return. This can be plotted in a graph, giving a better intuition for what risk and returns can be achieved. Efficiency frontiers are however not limited to using variance as a risk measure, and for CVaR is used instead in this thesis.

2.2 Time series

In statistical analysis, time series data stands out as a distinct and complex category that necessitates specialised analytical techniques. Unlike cross-sectional data where there are often many, as well as independent observations which are collected at a single point in time, time series data typically focuses on fewer variables and observations collected sequentially over time. This temporal dimension introduces unique characteristics and challenges that set time series apart.

In contrast to cross-sectional data, where the assumption of independence among observations often holds, time series data are inherently ordered, making it sensitive to the passage of time. The obvious potential for correlation between adjacent points in time can restrict the applicability of many conventional statistical methods relying on the assumption that these adjacent observations are independent and identically distributed.

Another aspect of time series data is the presence of trends and seasonality. Trends indicate long-term progression in the data, either upwards or downwards, while seasonality refers to regular, predictable patterns within a fixed period. These elements can obscure other characteristics of the data if not appropriately addressed.

Furthermore, time series data can exhibit non-stationarity, where its statistical properties such as mean, variance, and autocorrelation change over time. Traditional statistical methods usually assume stationarity, making them unsuitable for raw time series data. Techniques like differencing or transformation of data are often employed to achieve stationarity, enabling more effective analysis.

The analysis of time series data often aims at forecasting future values. This predictive aspect is crucial in numerous practical applications, ranging from weather forecasting to the focus of this thesis: financial market predictions.

2.2.1 Introduction to time series

A time series is a sequence of observations recorded in time. Such a sequence can be from a continuous time process, where observations can be recorded at any point in time. As it is impossible to collect an infinite amount of data points, the time series data itself will of course be in discrete, even though the process in continuous. The more common type of time series that this thesis focuses on however, is discrete time series, where the observations are recorded at specific and equally spaced time intervals. Many inherently continuous time series are approximated by discrete ones because they are simpler to work with. In the discrete case, x_t is used to represent the value at the t-th realisation of the series. A collection of such variables is referred to as a stochastic process.

When working with time series it is important to consider the resolution, which can make the same process look very different. By using longer time intervals, the process seems smoother by removing short time effects and noise, which might be favourable when looking at long term trends and effects. While shorter time intervals do the opposite, focusing on the short term effects.

2.2.2 Stationarity

Stationarity stands as a fundamental concept in time series analysis, denoting a property of a stochastic process, where the process generating the series does not change over time. This means a series can change over time, however, the way it changes does not change itself. This characteristic of stationary processes simplifies their analysis significantly and is, therefore, an underlying assumption for many procedures in time series analysis. A formal definition is given below:

Definition 5 (Shumway and Stoffer 2017)**.** *A* strictly stationary process *is a stochastic process where the unconditional joint probability distribution is constant with time. That is,*

$$Pr(x_{t_1} \leq c_1, \ldots, x_{t_k} \leq c_j) = Pr(x_{t_1+h} \leq c_1, \ldots, x_{t_k+h} \leq c_j) \tag{2.10}$$

for all $k = 1, 2, \ldots$*, all time points* $t_1, t_2, \ldots, t_k$*, all numbers* $c_1, c_2, \ldots, c_k$*, and all time shifts* $h = 0, \pm 1, \pm 2, \ldots$.

If the data is not stationary, it should be transformed to become so. A way of achieving this is through what is called differencing, introduced later in Equation (2.20) and Equation (2.21).

A less restrictive form of stationarity is called weak stationarity, which is the assumption that is used in practise for time series models and methods. Since this is the common usage in practice, in the course of this thesis the term stationarity will be used to refer to weak stationarity. It is defined as follows:

Definition 6 (Brockwell and Davis 2016)**.** *A process is* weakly stationary *if:*

- $\mathrm{E}[x_t] =$ *constant and independent of time.*
- $\mathrm{Var}(x_t) < \infty$ *and independent of time.*
- *the autocovariance function depends only on the lag* h *between two time points and not the times themselves:* $\gamma(s,t) = \gamma(t+h,t) = \gamma(h)$ *(see Definition 8)*

Another term of relevance is white noise (WN). It is a stronger form of stationarity with additional requirements listed in the following definition.

Definition 7. *A* white noise (WN) process *is a stricter form of weak stationarity having the additional requirements:*

- *Zero mean:* $\mathrm{E}[x_t] = 0$
- x_t *and* x_{t+h} *is uncorrelated for* $h \neq 0$*, i.e.* $\gamma(h) = 0$ *for* $h \neq 0$

The goal of analysing time series data is to find trends, patterns or fluctuations in the underlying structure. Therefore, the data needs to be split into signal and noise. If the residuals of a model are WN, the model is a good fit, as all the signal has been extracted from the data and the noise is the only remaining part.

2.2.3 Autocovariance and autocorrelation

Autocovariance and autocorrelation functions are fundamental concepts in the field of statistics and time series analysis. They play a crucial role in understanding the temporal dependencies and patterns within a sequence of data points, making them essential tools for various scientific disciplines.

The autocovariance function, denoted as $\gamma(k)$, quantifies the covariance between two observations in a time series. The observations can be any number of time steps apart from each other. This concept is called lag, where lag k represents a difference of k time units. In other words, the autocovariance function measures how the values of a time series at different points in time are related to each other in terms of their joint variability. Mathematically, the autocovariance function for a time series is defined as follows:

Definition 8. *The* autocovariance function *is defined as the second moment product for all s and t:*

$$\gamma_x(s,t) = \mathrm{Cov}(x_s, x_t) = \mathrm{E}[(x_s - \mu_s)(x_t - \mu_t)]. \tag{2.11}$$

The subscript of γ *is dropped when the time series referred to is obvious.*

For weakly stationary time series the autocovariance is independent of the time steps in between and the second argument will be dropped as follows:

$$\gamma(h) = \mathrm{Cov}(x_{t+h}, x_t) = \mathrm{E}[(xt + h - \mu_{t+h})(x_t - \mu_t)]. \tag{2.12}$$

The autocovariance function provides valuable insights into the stationary behaviour of the time series. Specifically, it helps identifying whether there are any systematic patterns or trends in the data at different time lags. Since it is more convenient to deal with measures between -1 and 1, the normalised version of the autocovariance function, called the autocorrelation function, is used instead.

Definition 9. *The* autocorrelation function (ACF) *is defined as:*

$$\rho(s,t) = \frac{\gamma(s,t)}{\sqrt{\gamma(s,s)\gamma(t,t)}} \tag{2.13}$$

For weakly stationary time series the ACF is independent of the time steps in between and the second argument will be dropped as follows:

$$\rho(h) = \frac{\gamma(t+h,t)}{\sqrt{\gamma(t+h,t+h)\gamma(t,t)}} = \frac{\gamma(h)}{\gamma(0)} \tag{2.14}$$

As the ACF measures the correlation between a point in time and one of lag k, it captures both the direct effect of a lag k, as well as the indirect effect through the points in between. Therefore, there is another measurement called the *partial* autocorrelation function (PACF) which only looks at the direct effect by removing the linear effect of the points in between.

Definition 10. *The* partial autocorrelation function (PACF) *of a zero mean stationary process, x_t, denoted ϕ_{hh}, for h = 1,2,..., is defined as:*

$$\phi(1,1) = \text{Corr}\,(x_{t+1}, x_t) = \rho(1) \tag{2.15}$$

and

$$\phi(h,h) = \text{Corr}\,(x_{t+h} - \hat{x}_{t+h}, x_t - \hat{x}_t) = \rho(h) \quad , \quad h \geq 2 \tag{2.16}$$

where $\hat{x}_{t+h}$ and $\hat{x}_t$ are the linear combinations of $\{x_{t+h-1}, x_{t+h-2}, \ldots, x_{t+1}\}$ that minimise the mean squared errors of $\text{E}\left[x_{t+h} - \sum_{j=1}^{h-1} \alpha_j x_{t+j}\right]$ and $\text{E}\left[x_t - \sum_{j=1}^{h-1} \alpha_j x_{t+j}\right]$ respectively.

2.3 ARMA models

ARMA models are a type of time series models commonly used for modelling and forecasting time-dependent data. They are put together by the two components autoregressive (AR) and moving average (MA). The autoregressive component represents the relationship between the current value of a time series and its past values, hence the name "auto" for self regression. Specifically, it models the current value as a linear combination of its own lagged values.

Definition 11. *(Shumway and Stoffer 2017) An* autoregressive model *of order p, written* AR(p)*, is of the form:*

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + w_t, \tag{2.17}$$

where x_t is stationary and ϕ_i are parameters for $i = 1, \ldots, p$, with $\phi_p \neq 0$. w_t is white noise: $w_t \sim wn(0, \sigma_w^2)$. Often, w_t is also independent and identically distributed (iid) and usually also normally distributed. The mean of x_t is assumed to be zero in Equation (2.17)*, otherwise, replace x_t with $x_t - \mu$, where $\mu = \text{E}\,[x_t]$*

Indications of an AR process can be seen in plots of the ACF and the PACF. If the ACF plot shows a tailing off of the value and the PACF shows a sharp drop after lag p, this indicates that the AR(p) model might be a good choice of model. This lag p indicates the order of the AR model (Shumway and Stoffer 2017), due to the direct relationship it measures.

The moving average model on the other hand, models based on the error terms from the previous steps. It can only give predictions for q steps into the future, after that, it will only predict the expectation.

Definition 12. *(Shumway and Stoffer 2017) The* moving average *model of order q, written* MA(q)*, is of the form:*

$$x_t = w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \ldots + \theta_q w_{t-q} \tag{2.18}$$

with $w_t \sim wn(0, \sigma_w^2)$, and θ_i are parameters for $i = 1, \ldots, q$, with $\theta_q \neq 0$. As for the case in Definition 11, w_t is often normally distributed iid and the mean of x_t is assumed to be zero.

The ACF and the PACF plots can also imply a MA(q) model in much of the same way as for the AR model. However, in this case, the PACF should be tailing off and a sharp drop after lag k in the ACF plot to a near zero value indicates the value of q to be k.

A practical example can be seen in Figure 2.2, which shows the ACF and PACF of the AR(2) model: $x_t = 1.5x_{t-1} - 0.75x_{t-2} + w_t$. The sharp drop in the PACF after lag 2 indicates the order correctly and in combination with the tailing off of the ACF.

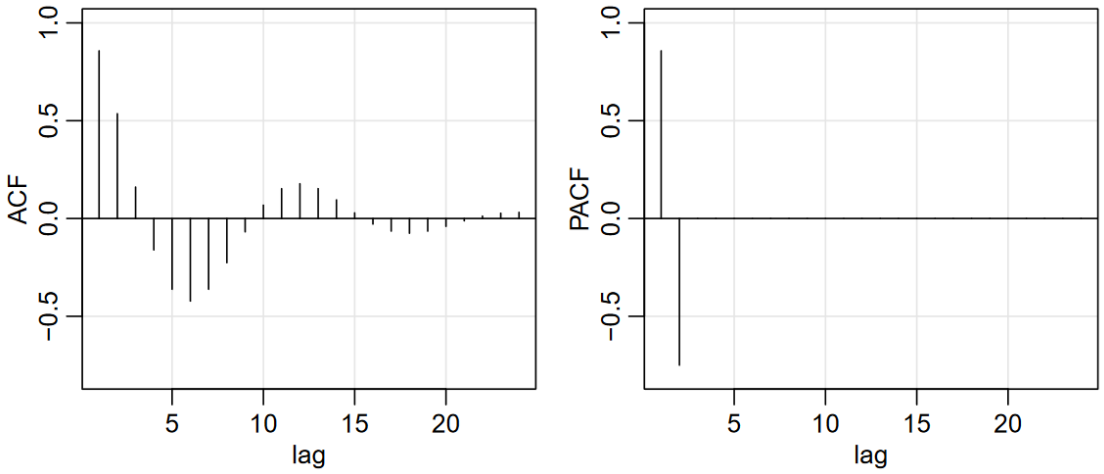


Figure 2.2: ACF and PACF plot for AR(2) model with $\phi_1 = 1.5$ and $\phi_2 = -0.75$. The PACF shows a sharp drop off after lag 2, while the ACF tails off, indicating the AR(2) structure correctly. (Figure taken from Shumway and Stoffer 2017)

The combination of the two models, AR and MA, make the ARMA(p,q) model, which captures the effects from both autoregressive and moving average characteristics present in time series data simultaneously. This makes it a versatile and widely-used tool for analysing and forecasting time series. It is defined as follows:

Definition 13. *The* ARMA(p,q) *model is on the form:*

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \theta_2 w_{t-2} + \ldots + \theta_q w_{t-q} \tag{2.19}$$

When using the ARMA model, the ACF and the PACF still give insightful information of the ARMA model, but it cannot be used to identify the order. This is because the indication of an ARMA model is that both the ACF and the PACF values are tailing off (Shumway and Stoffer 2017).

Often, trends are present in time series, which make them non-stationary, and therefore, need to be removed before further analysis. This is commonly achieved through differencing, a process that involves taking the differences between consecutive time steps. The simplest way of showing this is with the following model

$$x_t = \mu_t + y_t, \tag{2.20}$$

where x_t are the observations, μ_t is the trend and y_t is a stationary process. If then the trend is fixed $\mu_t = \beta_0 + \beta_1 t$, the differences between neighbouring data points:

$$\begin{aligned} x_t - x_{t-1} &= (\mu_t + y_t) - (\mu_{t-1} + y_{t-1}) = (\beta_0 + \beta_1 t + y_t) - (\beta_0 + \beta_1(t-1) + y_{t-1}) \\ &= \beta_1 + y_t - y_{t-1}, \end{aligned} \tag{2.21}$$

effectively make the series stationary (Shumway and Stoffer 2017). The extent of differencing, denoted by its degree, is aimed at eliminating trends of corresponding orders. This leads to the concept of the ARIMA model, where the I stands for integrated, and in this context integrated refers to the difference. Over-differencing, however, may introduce artificial dependencies that originally did not exist.

Specifically in financial time series, which frequently exhibit a general upward trend, log returns are often employed as a method of differencing, effectively normalising the data to account for this characteristic upward trajectory. However, this is not differencing of the raw data, but rather the differencing of the logarithm of the series.

There are many other expansions of the ARMA model handling different problems, many are however not applicable for this thesis and therefore not used. Some of these expansions are the generalisation of differencing by allowing for fractional differencing, which is used in the ARFIMA model. Its advantage is to capture long memory properties of time series, which is common for financial and economic time series, this thesis however, does not make use of this. Another common feature of times series are seasonal patterns which are found in data from temperatures to tourism. They can be handled by including seasonal components included with the SARIMA variant.

In conclusion, ARMA models are powerful tools for modelling, estimating and forecasting time series data by capturing the autoregressive and moving average components. However, they have limitations when dealing with heavy tailed distributions, asymmetry, volatility clusters, and non-linear serial dependencies. Such elements are crucial for financial and economic time series, where volatility or the dispersion of returns can change over time. This is where generalised autoregressive conditional heteroscedasticity (GARCH) models come into play.

2.4 GARCH

ARMA models usually assume that the residuals, w_t, are iid with constant variance. However, as can be seen from the returns of the Dow Jones Industrial Average Index (DJIA) in Figure 2.3, this assumption does not seem to hold. There are clearly periods with higher variance that occur in clusters, which indicate significant serial dependence.

Financial markets are self balancing by design, where any opportunity found will be exploited and therefore, vanish with time, making them impossible to predict with high certainty. However, the volatility in the markets does not vanish by the balancing effect of trades, making them a great asset to calculate risk.

This is where a tool for varying variance is needed. The GARCH model offers a framework to capture the evolving nature of market volatility. Unlike traditional models that assume constant variance over time, GARCH models allow variance to change, reflecting the real-world dynamics of financial markets. Although the GARCH model can be used by itself, it should be seen as a complementary to the ARMA model, supplementing each other by allowing a model to capture and forecast both the mean effect and the changing volatility patterns. These models are, therefore, used for addressing the dynamic nature of volatility in financial assets, like in Nguyen and Liu (2023).

As financial data has the self balancing characteristics, as mentioned above, the GARCH effect often dominates any ARMA structure. For the purpose of introducing the notion of the GARCH model itself, it is in this section demonstrated with the assumption of any mean structure to be an ARMA(0,0) process, meaning $x_t = w_t$. As

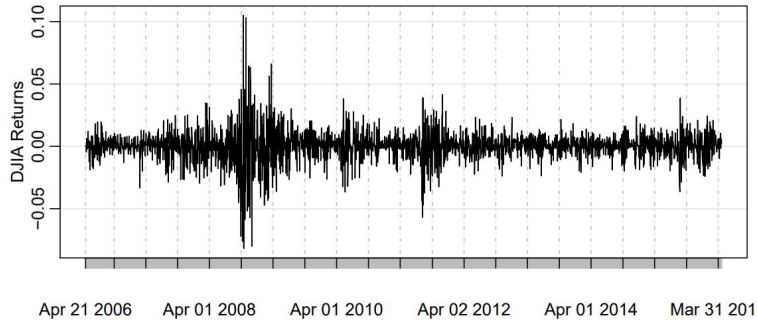


Figure 2.3: Returns on the DJIA index. The volatility does not look constant and there are clear clusters of increased volatility. The financial crisis of 2008 is naturally the biggest volatility cluster of this series. (Figure taken from Brockwell and Davis 2016)

the GARCH model is very complex, its concept will, therefore, be built up gradually in the following subsections.

2.4.1 ARCH

Intuitively, if a process has high volatility today, it is reasonable to believe it will be volatile tomorrow as well. For modelling this changing volatility, the ARCH model, which stands for *autoregressive conditional heteroscedasticity*, is a suitable choice. Conditional heteroscedasticity means the volatility is conditioned on time. The model is defined as follows:

Definition 14. *(Brockwell and Davis 2016) The* ARCH(p) *model is on the form:*

$$w_t = \sigma_t \epsilon_t \tag{2.22}$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 w_{t-1}^2 + \alpha_2 w_{t-2}^2 + \ldots + \alpha_p w_{t-p}^2 \tag{2.23}$$

where $\alpha_0 \geq 0$ *and* $\alpha_i \geq 0$ *for* $i = 1, \ldots, p$*. In addition* $\epsilon_t \sim iid(0,1)$*, but it is often assumed normally distributed as well.*

Here, α_0 can be seen as a representation of the minimum level of variance, where the other terms on the right hand side of Equation (2.23), increases the variance in an additive fashion. The α_i for $i = 1, \ldots, p$ are the scalars deciding how big these effect are for their respective days. By writing Equation (2.22) as $x_t = w_t = \sigma_t \epsilon_t$, it becomes clear that there is no ARMA model present, which is the same as saying it has ARMA(0,0) structure. If then rewriting the error term, w_t, of the ARMA model, to

$$w_t = \sigma_w \epsilon_t, \quad \epsilon_t \sim iid(0,1),$$

we also notice how the ARCH model simply replaces the standard deviation of the white noise, σ_w, with the time varying standard deviation σ_t.

The ARCH model can resemble the AR model as they both have an autoregressive structure. This becomes clearer if Equation (2.22) is squared and combined with Equation (2.23) to make

$$w_t^2 = (\alpha_0 + \alpha_1 w_{t-1}^2 + \alpha_2 w_{t-2}^2 + \ldots + \alpha_p w_{t-p}^2)\epsilon_t^2.$$

However, in contrast to the AR model, which is additive and linear, the ARCH model is autoregressive in the squared values and multiplicative with ϵ_t.

Intuitively, the ARCH model captures the idea that periods of high volatility tend to cluster in financial markets. For instance, large changes in prices are likely to be followed by more changes of big amplitude. The ARCH model quantifies this observation by allowing the current period's volatility to depend on the magnitudes of previous periods' shocks. An example of this is displayed through the ARCH(2):

$$w_t = \sigma_t \epsilon_t, \quad \sigma_t = \sqrt{\alpha_0 + \alpha_1 w_{t-1}^2 + \alpha_2 w_{t-2}^2}. \tag{2.24}$$

The variance here is dependent on the previous realisations of the series itself, and through the previous w_t's, dependent on previous ϵ_t's. If then the last two realisations were big in absolute value, the square root would also be big, which would indicate a high value in the current step. However, if then ϵ would by chance be very small two times in a row, the realisations of x_t will be small as well. Then the model will "forget" the high variance completely after only two steps. This is why ARCH models give spikes of variance. Therefore, in a high variance market with a few calm days, the ARCH model will quickly "forget" the high volatility. This shortcoming is what the extended version, the GARCH model, which will be presented later, is trying to handle.

To get a better understanding of the properties of the ARCH model, it is easiest to work with its simplest form, the ARCH(1) model:

$$w_t = \sigma_t \epsilon_t \tag{2.25}$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 w_{t-1}^2 \tag{2.26}$$

Recursively using Equation (2.25) and Equation (2.26) shows how each term progressively adds to the current volatility, reflecting the impact of past shocks:

$$\begin{aligned} w_t^2 &= \alpha_0 \epsilon_t^2 + \alpha_1 x_{t-1}^2 \epsilon_t^2 \\ &= \alpha_0 \epsilon_t^2 + \alpha_1 (\alpha_0 \epsilon_{t-1}^2 + \alpha_1 x_{t-2}^2 \epsilon_{t-1}^2) \epsilon_t^2 \\ &= \dots \\ &= \alpha_0 \sum_{j=0}^{n} \alpha_1^j \epsilon_t^2 \epsilon_{t-1}^2 \cdots \epsilon_{t-j}^2 + \alpha_1^{n+1} w_{t-n-1}^2 \epsilon_t^2 \epsilon_{t-1}^2 \cdots \epsilon_{t-n}^2. \end{aligned} \tag{2.27}$$

It is shown in Brockwell and Davis (2016) that w_t in the ARCH(1) process can be rewritten under some assumptions to:

$$w_t = \epsilon_t \sqrt{\alpha_0 (1 + \sum_{j=1}^{n} \alpha_1^j \epsilon_t^2 \epsilon_{t-1}^2 \cdots \epsilon_{t-j}^2)}. \tag{2.28}$$

Which make it clear that w_t is strictly stationary with the unconditional mean, variance and covariance as follows:

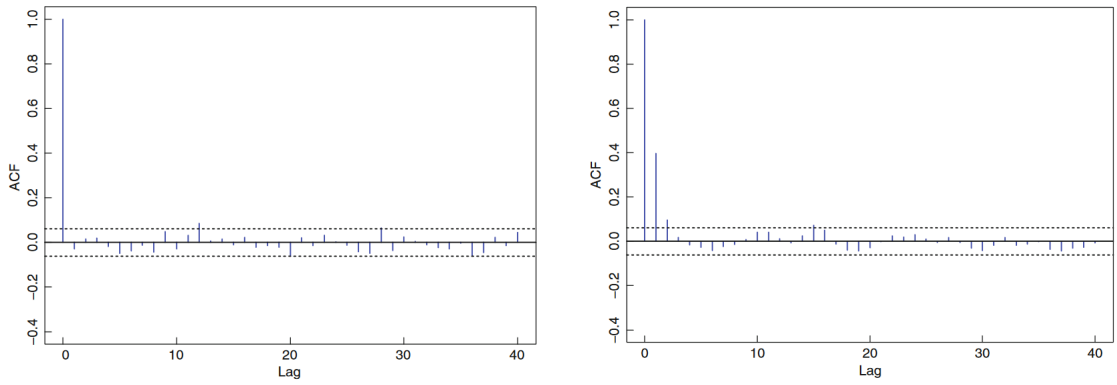
$$\mathrm{E}\left[w_t\right] = \mathrm{E}\left[\mathrm{E}\left[w_t | \epsilon_s, s < t\right]\right] = 0 \tag{2.29}$$

$$\mathrm{Var}\left(w_t\right) = \frac{\alpha_0}{1 - \alpha_1} \tag{2.30}$$

$$\mathrm{Cov}\left(w_{t+h}, w_t\right) = \mathrm{E}\left[w_{t+h} w_t\right] = \mathrm{E}\left[\mathrm{E}\left[w_{t+h} w_t | \epsilon_s, s < t + h\right]\right] = 0, \quad \text{for} \quad h > 0. \tag{2.31}$$

This shows that ARCH(1) with $\alpha_1 < 1$ is strictly stationary white noise, however, it is not independent, because:

$$\mathrm{E}\left[w_t^2 | w_{t-1}\right] = (\alpha_0 + \alpha_1 w_{t-1}^2)\, \mathrm{E}\left[\epsilon_t^2 | w_{t-1}\right] = \alpha_0 + \alpha_1 w_{t-1}^2 \tag{2.32}$$



a ACF plot on the realisations

b ACF plot on squared realisations

Figure 2.4: ACF plots for the process $w_t = \epsilon_t \sqrt{1 + 0.5 w_{t-1}^2}$. Figure 2.4a suggests a white noise process, Figure 2.4b, on the other hand, shows autocorrelation on the squared series. It is important to notice that this plot shows lag 0 which is obviously 1, contrary to similar plots in this thesis. (The figures are taken from Brockwell and Davis 2016)

This is illustrated in Figure 2.4, where the ACF plot shows no structure for the original series, but a structure for the series squared.

Equation (2.28) shows that w_t has a symmetric nature. It can also be shown that for every α_1 between 0 and 1, $\mathrm{E}\left[w^{2k}\right] = \infty$ for positive integers, k, indicating the heavy tailed nature of the marginal distribution of w_t (Brockwell and Davis 2016).

In general the ARCH(1) with the assumption that $\epsilon_t \sim \mathrm{N}(0, 1)$, is conditional Gaussian, $x_t | x_{t-1} \overset{iid}{\sim} \mathcal{N}(0, \alpha_0 + \alpha_1 x_{t-1}^2)$, making it easy to compute the conditional maximum likelihood estimates of the parameters. For this thesis however, Gaussian distributions are not assumed, but likelihood estimation is still applied. Many of the other properties of ARCH(1) explained here can be extended to ARCH(p). (Shumway and Stoffer 2017)

2.4.2 GARCH

The generalised autoregressive conditional heteroskedasticity (GARCH) model is an extension of the ARCH model, designed to better capture the persistence of volatility in financial markets over time. How this better capture persistence of volatility will be elaborated on further through a practical example later.

Definition 15. *A time series* $\{w_t\}$ *follows a* GARCH(p, q) *process if it can be modelled as:*

$$w_t = \sigma_t \epsilon_t, \tag{2.33}$$

where $\epsilon_t \sim iid(0, 1)$*, and the conditional variance* σ_t^2 *is given by:*

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i w_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2, \tag{2.34}$$

with $\alpha_0 > 0$ *and* $\alpha_j, \beta_j \geq 0$.

In this model, the variance equation of the ARCH model is expanded. It still includes the past squared realisations of w_t from the ARCH model, referred to as the ARCH part, as well as its past variances, referred to as the GARCH part. The parameters α_i and

β_j measure the impact of past squared returns and past variances, respectively, on the current variance. The term α_0 still represents a minimum level of volatility, while the condition $\alpha_1 + \beta_1 < 1$ ensures the model's stationarity and causal nature. Where causal in this sense means that it does not depend on future observation, a property if not satisfied, deems forecasting useless.

While it is normal to assume $\epsilon_t \sim \mathcal{N}(0, 1)$, analysis of empirical financial data usually show heavier tailed, zero-mean distributions, for w_t (Brockwell and Davis 2016). Another distribution that is also assumed for ϵ_t is the Student's t-distribution Definition 19, which then is defined like the following:

$$\sqrt{\frac{v}{v-2}}\epsilon_t \sim t_v, \quad v > 2, \tag{2.35}$$

where t_v denotes the Student's t-distribution and ν is its degree of freedom. The square root is simply a scaling factor that makes the variance of ϵ_t equal to 1. For this thesis however, a skewed version of the Student t-distribution is used, which is introduced in Section 2.5.3.

The GARCH model's ability to incorporate past variances makes it more flexible and realistic for financial time series analysis. It captures the phenomenon where high-volatility periods are likely to follow other high-volatility periods, thus providing a more comprehensive understanding of market dynamics.

For a better understanding of how the GARCH part of the GARCH model handles longer periods of volatility, an example of the GARCH(0,2) model is used:

$$w_t = \sigma_t \epsilon_t, \quad \sigma_t = \sqrt{\alpha_0 + \beta_1 \sigma_{t-1}^2 + \beta_2 \sigma_{t-2}^2}. \tag{2.36}$$

Here, the variance is independent of the previous realisations of ϵ, instead, only previous conditional variances matter. Therefore, if the two previous ϵ values are small, but the conditional variances are not, the variance for the current step will still remember this volatility, giving it its long term memory. However, as it is not reliant on ϵ, it is not able to react quickly enough to any shocks in the market, like the ARCH model could. Therefore, the great advantage of the GARCH model is that the ARCH part can quickly react to changes in volatility and that the GARCH part gives long term memory for volatility.

To sum up, the GARCH model has been found to provide a better fit for financial data, especially when non-Gaussian distributions, such as Student's t-distribution, are assumed for ϵ_t. This flexibility allows the GARCH model to effectively capture the persistence of volatility observed in financial markets, where large fluctuations tend to be followed by similar magnitude fluctuations (Brockwell and Davis 2016).

2.4.3 EGARCH

While the GARCH model can account for heavy tailed marginal distributions and persistent volatility, financial return series additionally frequently display a form of asymmetry related to disturbances, known as the leverage effect. This effect entails a scenario wherein negative shocks exert a more pronounced influence on future volatility compared to positive ones (Brockwell and Davis 2016). Capturing this nuanced behaviour, the exponential generalised autoregressive conditional heteroskedasticity (EGARCH) model introduces the flexibility to account for varied impacts on future volatility, based on whether the values for ϵ_t were positive or negative.

Definition 16. *The* EGARCH(1,1) *model is defined as follows:*

$$w_t = \sigma_t \epsilon_t, \quad \epsilon_t \sim iid(0,1), \tag{2.37}$$

$$\log \sigma_t^2 = \alpha_0 + \alpha_1 g(\epsilon_{t-1}) + \beta_1 \log \sigma_{t-1}^2 \tag{2.38}$$

where $\alpha_0, \alpha_1 \in \mathbb{R}$, $|\beta_1| < 1$ *and:*

$$g(\epsilon_t) = \epsilon_t + \lambda(|\epsilon_t| - \mathrm{E}\,[|\epsilon_t|]). \tag{2.39}$$

The difference to the GARCH model is that, instead of being additive on the original scale of σ_t^2, the EGARCH model is additive one the logarithmic scale $\log \sigma_t^2$, making it multiplicative on the original scale. This assures that $\sigma_t > 0$, since:

$$\sigma_t^2 = e^{\alpha_0 + \alpha_1 g(\epsilon_{t-1}) + \beta_1 \log \sigma_{t-1}^2} = e^{\alpha_0} (\sigma_{t-1}^2)^{\beta_1} e^{\alpha_1 g(\epsilon_{t-1})}, \tag{2.40}$$

where e^{α_0} is of course a constant since α_0 is.

In addition, the $g(\epsilon_t)$ function is defined to differentiate the impact of positive and negative shocks. By rewriting it to

$$g(\epsilon_t) = \begin{cases} (1+\lambda)\epsilon_t - \lambda\,\mathrm{E}\,[|\epsilon_t|]) & \text{if } \epsilon_t \geq 0 \\ (1-\lambda)\epsilon_t - \lambda\,\mathrm{E}\,[|\epsilon_t|]) & \text{if } \epsilon_t < 0, \end{cases} \tag{2.41}$$

it becomes clear how g is piece-wise linear with slope $(1+\lambda)$ if ϵ_t is positive and $(1-\lambda)$ otherwise. This allows the model to respond differently to the same magnitude of shocks dependent on it being positive or negative. λ will typically be estimated to be negative, which causes a steeper slope for negative values of ϵ_y, aligning with financial data, since it typically reacts more strongly to negative shocks (Embrechts, Lindskog and McNeil 2001).

The EGARCH(p, q) model generalises the structure of $\log \sigma_t^2$ to include higher-order terms, ensuring stationarity and causality under specific conditions.

One should not confuse the parameters in the EGARCH model presented here, with the output from the rugarch R-package. For this thesis any output from that package is transformed to follow the definition and naming convention given here. However, for clarity, their implementation deviates in the following way by implementing the function $g(\epsilon_t)$ for each order of the ARCH(p) as:

$$g_i(\epsilon_t) = \epsilon_t + \gamma_i(|\epsilon_t| - \mathrm{E}\,[|\epsilon_t|]), \tag{2.42}$$

where $\gamma_i = \lambda/\alpha_i$ for $i = 1, \ldots, p$. To put it simply, the single parameter λ in Definition 16, is replaced with a parameter γ_i for each of the the p orders of the ARCH(p) part, by dividing by the respective α_i value.

2.5 Skew and heavy tailed distributions

Financial time series data often deviate from the symmetrical patterns assumed in normal distributions (Brockwell and Davis 2016). These deviations, characterised by skew and heavy tails, can impact the performance and accuracy of statistical models used in financial analysis. Recognising and appropriately modelling these characteristics is crucial for financial forecasting and risk assessment, which is relevant for the risk and portfolio prediction employed in this thesis using ARMA-EGARCH and copula models.

2.5.1 Skewness and kurtosis

Skewness measures the asymmetry of the probability distribution of a real-valued random variable about its mean. Skewness is a measure of the shape of the data and makes the distribution easier to understand. For unimodal distributions a positive skewness indicates that the distribution has a longer right tail, while a negative value indicates a longer left tail. However, this is solely an indicative measure, as distributions with varying tail lengths can exhibit similar skewness values. For instance, if the tails balance each other out, despite their different lengths.

Definition 17. *The* skewness *of a random variable X is the third standardised moment, defined as:*

$$\mu_3 = \mathrm{E}\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] \tag{2.43}$$

where μ is the mean and σ is the standard deviation of X.

Heavy-tailed distributions signify a higher probability of extreme events or outliers than what is predicted by a normal distribution. Kurtosis is a statistical measure for quantifying the shape of probability distributions and for indicating heavy tails or not. A higher value of kurtosis indicates heavier tails, and a low value lighter tails. If the normal distribution is defined to have a kurtosis of 0, heavy tailed distributions have values higher than this.

Definition 18. *The* kurtosis *of a random variable X is the forth standardised moment, defined as:*

$$\mu_4 = \mathrm{E}\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] - 3 \tag{2.44}$$

2.5.2 Student's t-distribution

One of the distributions that align well with financial returns is the student's t-distribution. It is similar to the normal distribution with a symmetric, bell-shaped distribution centred around zero. It distinguishes itself by its heavier tails, indicated by the degrees of freedom parameter ν. This allows a more accurate representation of the likelihood of extreme market movements, with lower values of ν indicating heavier tails. By varying the value for ν, the distribution can become the standard normal distribution by setting $\nu \to \infty$ and also the Cauchy distribution if $\nu = 1$.

Definition 19. *The probability density function of the* Student's t-distribution *for the random variable x is:*

$$f(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})}\left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}} \tag{2.45}$$

where Γ is the gamma function and ν is the degree of freedom.

The standardised Student's t-distribution in Definition 19 can be described only using the shape parameter ν for the degree of freedom. However, for this thesis the Student's t-distribution is used through the implementations of the R-package rugarch, which has implemented a standardising process making use of 3 parameters as follows (Ghalanos 2023):

$$f(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\beta\nu\pi}\Gamma(\frac{\nu}{2})}(1+\frac{(x-\alpha)^2}{\beta\nu})^{-\frac{\nu+1}{2}}. \tag{2.46}$$

Here α and β are the location and scale parameters respectively.

2.5.3 Skewed student's t-distribution

Addressing the limitations of both the normal and standard Student's t-distributions, the skewed Student's t-distribution offers an asymmetric and skewed probability density. This accommodates nuances in financial data, as illustrated by the heavy tailed and left-skewed residuals of the ARMA-GARCH models, used in the empirical study of this thesis, shown in the histograms in Figure 2.5.

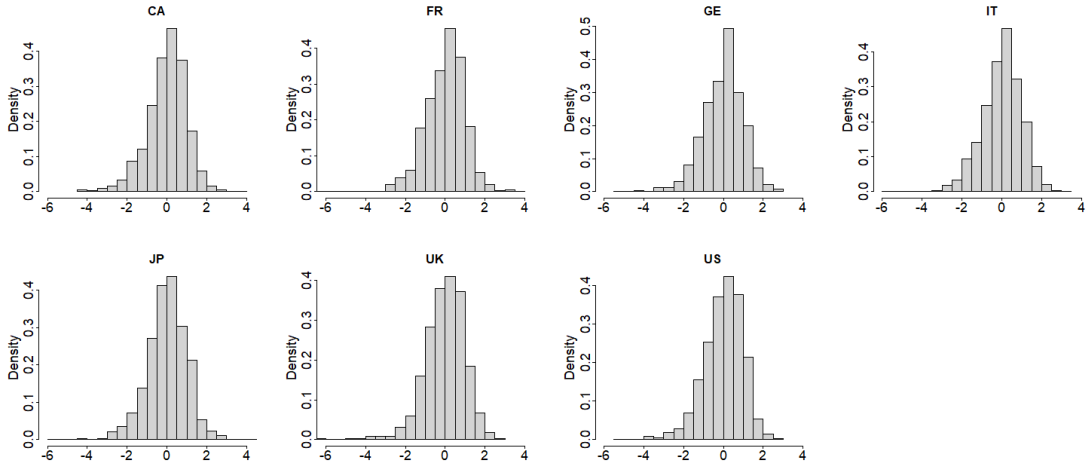


Figure 2.5: Histograms of standardised residuals from the ARMA-EGARCH models for the full series from 1999-2019.

There are several ways to skew the Student's t-distribution. In the rugarch package mentioned above, it is implemented by using the transformation methods described in Fernandez and Steel (1998), which introduces inverse scale factors. The density for a random variable, x, given a skew parameter, ξ, can be represented as:

$$f(x|\xi) = \frac{2}{\xi + \xi^{-1}}(f(\xi x)H(-x) + f(\xi^{-1}x)H(x)), \tag{2.47}$$

where $\xi \in \mathbb{R}^+$ and a value of 1 means the distribution is symmetric. $H(\cdot)$ is the Heaviside function defined as:

$$H(x) := \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0. \end{cases} \tag{2.48}$$

The skew Student's t-distribution used in this thesis is therefore, obtained by applying Equation (2.47) on Equation (2.46).

2.6 Tests used for evaluation of models

The Ljung-Box test is a statistical test used to determine whether there are significant autocorrelations in a time series. It's commonly employed in the analysis of residuals in ARIMA models to check the adequacy of the model fit. The test examines a null hypothesis that the data are independently distributed, with no autocorrelation up to a specified lag. A significant test result suggests the presence of autocorrelation, indicating that the model may need to be re-specified to better capture the underlying data structure.

2.7 Copulas

Copulas are multivariate distribution functions that describe the dependence structure between random variables. The primary benefit of using copulas is that they provide a way to decouple the joint distribution of variables from their marginal distributions. This makes it easier to study and model the dependency between variables independently of how they behave individually, allowing analysts to focus on the way variables co-move with each other. This is particularly well-suited for financial applications, where for example, the continuous prices for individual stocks can give the impression of moving independently of one another, yet underlying intricate dependence structures exist.

Copulas in a time series context mainly work when the time series are stationary and the variables are continuous. When time series are auto-correlated, they may generate a non-existent dependence between sets of variables and result in incorrect copula dependence structure. Thus, a very important pre-processing step is to check for auto-correlation, trends and seasonality within time series.

In copula applications, a key step involves standardising the marginal distributions, effectively transforming each of them to conform to a uniform distribution over the interval [0,1]. This standardisation process offers multiple advantages. Firstly, it ensures that the copula captures only the dependence structure and not the specifics of the marginal distributions. Secondly, it enables the analysis of the relationships between variables without the need for specific assumptions about their individual probability distributions. This means that one can employ a copula to model dependence between variables such as stock returns, without making any particular assumptions regarding the probability distribution of each stock's returns.

2.7.1 Mathematical introduction to copulas

As mentioned above, all margins must be uniform to construct a copula. If they are not uniform already, they can be transformed using the probability integral transformation. The transformation states that given a random continuous variable X with the cumulative distribution function F_X, then if U is defined as $U = F_X(X)$, U has a standard uniform distribution. For d variables in a vector $\mathbf{X}$ this gives:

$$(U_1, \ldots, U_d) = (F_1(X_1), \ldots, F_d(X_d)). \tag{2.49}$$

From this point on, the distributions are assumed to be standard uniform, unless otherwise stated.

Before we are able to fully understand and explore the functions and possibilities of copulas, it is necessary to define

Definition 20 (Embrechts, Lindskog and McNeil 2001)**.** *Let $S_1, \ldots, S_d$ be nonempty subsets of $\overline{\mathbb{R}}$, where $\overline{\mathbb{R}}$ denotes the extended real line $[-\infty, \infty]$. Let F be a real function of d variables such that $Domain(F) = S_1 \times \ldots \times S_d$ and for $a \leq b$ $(a_k \leq b_k \forall k)$, let $B = [a, b] (= [a_1, b_1] \times \ldots \times [a_d, b_d])$ be a d-box whose vertices are in $Domain(F)$. Then the F-volume of an d-box B is given by the d-th order difference F on B:*

$$V_F(B) = \Delta_{\mathbf{a}}^{\mathbf{b}} F(\mathbf{t}) = \Delta_{a_d}^{b_d} \ldots \Delta_{a_1}^{b_1} F(\mathbf{t})$$

where

$$\Delta_{a_k}^{b_k} F(\mathbf{t}) = F(t_1, \ldots, t_{k-1}, b_k, t_{k+1}, \ldots, t_d) - F(t_1, \ldots, t_{k-1}, a_k, t_{k+1}, \ldots, t_d)$$

With this in mind, a copula can now be defined in the following way.

Definition 21. *A copula C of dimension d is a function that transforms a d-dimensional input u on the unit cube* $[0,1]^d$ *to [0,1] with the following properties (Embrechts, Lindskog and McNeil 2001, p. 3):*

- *For every* $\mathbf{u}$ *in* $[0,1]^d$, $C(\mathbf{u}) = 0$ *if at least one coordinate of* $\mathbf{u}$ *is 0*
- $C(\mathbf{u}) = u_k$ *if all coordinates of* $\mathbf{u}$ *equal 1 except* u_k
- $V_C([\mathbf{a}, \mathbf{b}]) \geq 0$, *for every* $\mathbf{a}$ *and* $\mathbf{b}$ *in* $[0,1]^d$ *such that* $a_i \leq b_i$ *for all* i.

These points are fulfilled for distribution functions with uniform marginal distributions, which will be used in this thesis.

Sklar's theorem

The fundamental theorem of copulas is Sklar's theorem, which states the following.

Theorem 1 (Sklar 1959)**.** *If F is a d-dimensional distribution function with margins* $F_1, \ldots, F_d$. *Then there exists a copula C such that,*

$$F(x_1, \ldots, x_d) = C[F_1(x_1), \ldots, F_d(x_d)], \tag{2.50}$$

for all x_i *in* $[-\infty, \infty]$, $i = 1, \ldots, d$. *If* F_i *is continuous for all i, C is unique, otherwise C is uniquely determined only on* $Range(F_1) \times \ldots \times Range(F_d)$. *On the other hand, if C is a copula and* $F_1, \ldots, F_d$ *are distribution functions, then the function F, defined above, is a multivariate distribution function with marginals* $F_1, \ldots, F_d$

This states the core idea of copulas: namely, that any multivariate joint distribution can be written in terms of univariate marginals distribution functions and a copula which describes their dependence structure. Equation (2.50) shows the cumulative distribution function for a copula, however, if the marginal distributions are differentiable, the probability density function of a copula is

$$c[F_1(x_1), \ldots, F_d(x_d)] = \frac{\partial^d C[F_1(x_1), \ldots, F_d(x_d)]}{\partial F_1(x_1) \ldots \partial F_d(x_d)}. \tag{2.51}$$

Using the chain rule, it follows from Equation (2.50), that the probability density function for Sklar is

$$f(x_1, \ldots, x_d) = c[F_1(x_1), \ldots, F_d(x_d)] \cdot f_1(x_1) \ldots f_d(x_d). \tag{2.52}$$

Bounds

Consider the following functions defined on $[0,1]^d$:

$$M^d(\mathbf{u}) = \min(u_1, \ldots, u_d) \tag{2.53}$$

$$W^d(\mathbf{u}) = \max(u_1 + \ldots + u_d - n + 1, 0), \tag{2.54}$$

which are denoted the Fréchet-Hoeffding bounds:

Theorem 2 (Fréchet 1957)**.** *If C is any copula, then for every* $\mathbf{u}$ *in* $[0,1]^d$

$$W^d(\mathbf{u}) \leq C(\mathbf{u}) \leq M^d(\mathbf{u}) \tag{2.55}$$

This theorem shows that any copula is bounded within these two functions. The function M^d is a d-copula itself for $d \geq 2$, while W^d is not. In fact, even though W^d in not a copula for any $d \geq 3$, it is the best possible lower bound. For better intuition, Figure 2.6 illustrates the possible values between the upper and lower bounds.

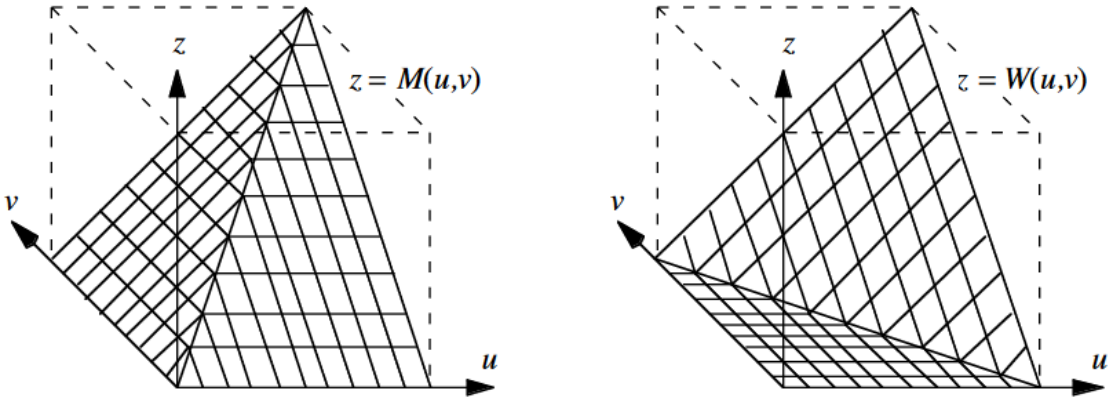


Figure 2.6: Example of Fréchet-Hoeffding bounds for the bivariate case. The lower bound is made up by the bottom of the pyramid and the back (from the dashed line up to the point (1,1)). The upper bound is made up of the two front facing surfaces. (Figure taken from Nelson 2006).

2.7.2 Vines

Multivariate dependent data can be complex to model, therefore, using a cascade of simpler building blocks can be a powerful tool for model construction (Aas et al. 2009). In this case the building blocks will be pair-copulas, meaning a bivariate copula. A consequence of this approach is that the number of possible pair-copula constructions quickly becomes huge, which makes it difficult to keep track of all the configurations. To organise them, Bedford and R. M. Cooke 2001 have introduced the *the regular vine*. For a better understanding, pair-copulas constructions will be introduced before delving further into the details of vines.

Multidimensional decomposition with pair-copulas

Now we are taking a closer look at the factorisation of a joint density function of a d-dimensional continuous random vector $\mathbf{X} = (X_1, \ldots, X_d)$. Its uniqueness is preserved under relabelling and is

$$f(x_1, \ldots, x_d) = f_d(x_d) \cdot f(x_{d-1}|x_d) \cdot f(x_{d-2}|x_{d-1}, x_d) \cdots f(x_1|x_2, \ldots, x_d). \tag{2.56}$$

This makes it clear that every joint distribution function implicitly contains the description of the marginals of individual variables and a description of their dependence structure (Aas et al. 2009). With copulas providing a way of isolating the dependence structure, the bivariate case of a joint distribution can be written as

$$f(x_1, x_2) = c_{1,2}[F_1(x_1), F_2(x_2)] \cdot f_1(x_1) \cdot f_2(x_2), \tag{2.57}$$

where $c_{1,2}(\cdot, \cdot)$ is the pair-copula density for the pair of transformed variables $F_1(x_1)$ and $F_2(x_2)$. Be aware that notation ordering is inconsequential, meaning: $c_{1,2}[F_1(x_1), F_2(x_2)] = c_{2,1}[F_2(x_2), F_1(x_1)]$.

The conditional density can also be represented with the same pair-copula, but with only one marginal instead of two, such as:

$$\begin{aligned} f(x_1|x_2) = \frac{f(x_1, x_2)}{f(x_2)} &= \frac{c_{1,2}[F_1(x_1), F_2(x_2)] \cdot f_1(x_1) \cdot f_2(x_2)}{f(x_2)} \\ &= c_{1,2}[F_1(x_1), F_2(x_2)] \cdot f_1(x_1), \end{aligned} \tag{2.58}$$

which can be generalised and shuffled around to define the copula as:

$$c_{j,k}[F_j(x_j), F_k(x_k)] = \frac{f(x_j|x_k)}{f_j(x_j)}. \tag{2.59}$$

When increasing the dimensions up to three, the density can be written as:

$$\begin{aligned} f(x_1|x_2,x_3) &= \frac{f(x_1,x_2,x_3)}{f(x_2,x_3)} = \frac{f(x_1,x_2,x_3)/f(x_3)}{f(x_2,x_3)/f(x_3)} = \frac{f(x_1,x_2|x_3)}{f(x_2|x_3)} \\ &= \frac{f(x_1,x_2|x_3)\cdot f(x_1|x_3)}{f(x_2|x_3)\cdot f(x_1|x_3)} = c_{1,2|3}[F(x_1|x_3), F(x_2|x_3)]\cdot f(x_1|x_3) \\ &= c_{1,2|3}[F(x_1|x_3), F(x_2|x_3)]\cdot c_{1,3}[F_1(x_1), F_3(x_3)]\cdot f_1(x_1), \end{aligned} \tag{2.60}$$

However, this is not the only description of this conditional density. Using the pair-copula $c_{1,3|2}$ instead of $c_{1,2|3}$, gives

$$\begin{aligned} f(x_1|x_2,x_3) &= c_{1,3|2}[F(x_1|x_2), F(x_3|x_2)]\cdot f(x_1|x_2) \\ &= c_{1,3|2}[F(x_1|x_2), F(x_3|x_2)]\cdot c_{1,2}[F(x_1), F(x_2)\cdot f_1(x_1). \end{aligned} \tag{2.61}$$

This shows that $f(x_1|x_2,x_3)$ has two pair-copulas representations, explaining how the number of decompositions increase with the dimension.

In the general case, the terms in Equation (2.56) can be decomposed into pair-copulas multiplied by a conditional marginal density, using the formula

$$f(x|\mathbf{v}) = c_{x,v_j|\mathbf{v}_{-j}}[F(x|\mathbf{v}_{-j}), F(v_j|\mathbf{v}_{-j})]\cdot f(x|\mathbf{v}_{-j}), \tag{2.62}$$

where v_j is an arbitrary component of $\mathbf{v}$ and $\mathbf{v}_{-j}$ denotes the remaining components of the vector. In summary, with certain regularity conditions met, it is possible to represent a multivariate density as a combination of pair-copulas applied to various conditional probability distributions. Furthermore, this construction process is inherently iterative, and for a given factorisation, multiple re-parameterisations can exist.

Vine copula

A vine is a tool for describing constraints on multivariate distributions (Bedford and R. M. Cooke 2002). In this thesis it is used for describing a nested set of connected trees, where the edges represent a pair-copula between two nodes. By using vines, the logic of the building complexity is kept while using simpler blocks, which enables flexibility and a high level of intuition. The regular vine, also called R-vine, is a very general form of vines, however, the special cases of vines, *canonical vines* (also called C-vine) and *D-vine* (Kurowicka and R. Cooke 2005) are subsets of R-vines with simpler structures and will therefore be introduced first.

D-vines

The D-vine might be the simplest vine to understand due to the nodes having a maximum of two edges and therefore, making it possible to be illustrated with flat layers of trees stacked on top of each other. An example of this is shown for the five dimensional case in Figure 2.7, which has four trees denoted T_j with $j = 1, \ldots, 4$. In general, all trees in a vine have $6 - j$ nodes and one less edge. The whole construction, called a pair copula construction, is defined by the $d(d-1)/2$ edges built up from the marginal densities of

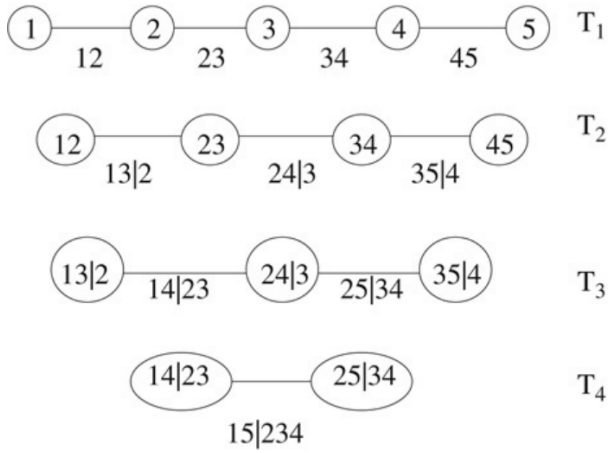


Figure 2.7: Example of D-vine with 5 variables. (The figures are taken from Aas et al. 2009)

each variable. Mathematically, the density of any D-vine can be represented as (Aas et al. 2009)

$$f(x_1,\dots,x_d) = \prod_{k=1}^{d} f(x_k) \prod_{j=1}^{d-1} \prod_{i=1}^{d-j} c_{i,i+j|i+1,\dots,i+j-1} \\ [F(x_i|x_{i+1},\dots,x_{i+j-1}), F(x_{i+j}|x_{i+1},\dots,x_{i+j-1})], \quad (2.63)$$

where index j identifies the tree and i represents that tree's edges.

Figure 2.7 represents the D-vine shown in Equation (2.64), where the marginal distributions are the nodes in the first tree. Notice how the edges represent the pair-copulas, like how the edge labelled 13|2 in T_2 corresponds to the pair-copula density with the same subscript $c_{1,3|2}(\cdot)$. Also notice how each tree T_{j+1} has nodes that correspond to the edges of T_j and that the edges in T_{j+1} are conditioned on their common variable, like how edges 12 and 23 in T_1 become nodes in T_2 with the edge 13|2 between themselves.

As D-vines are limited to being flat trees stacked on top of each other, and each layer is dependent on the one before, they are in fact completely defined by the ordering in the first tree. This is due to how the pair copulas has to be conditioned on a common parameter. As the nodes in a D-vine are connected to at most two other nodes, the D-vine could be advantageous when there is a natural order among the variables. Since the end nodes are connected to only one other node directly, it is reasonable to assume these are among the least influential variables.

$$
\begin{aligned}
f(x_1, x_2, x_3, x_4, x_5) = &\prod_{k=1}^{5} f(x_k) \prod_{j=1}^{5-1} \prod_{i=1}^{5-j} c_{i,i+j|i+1,\ldots,i+j-1} \\
&[F(x_i|x_{i+1}, \ldots, x_{i+j-1}), F(x_{i+j}|x_{i+1}, \ldots, x_{i+j-1})] \\
&= f_1(x_1) \cdot f_2(x_2) \cdot f_3(x_3) \cdot f_4(x_4) \cdot f_5(x_5) \\
&\cdot c_{1,2}[F_1(x_1), F_2(x_2)] \cdot c_{2,3}[F_2(x_2), F_3(x_3)] \\
&\cdot c_{3,4}[F_3(x_3), F_4(x_4)] \cdot c_{4,5}[F_4(x_4), F_5(x_5)] \\
&\cdot c_{1,3|2}[F(x_1|x_2), F(x_3|x_2)] \cdot c_{2,4|3}[F(x_2|x_3), F(x_4|x_3)] \\
&\cdot c_{3,5|4}[F(x_3|x_4), F(x_5|x_4)] \\
&\cdot c_{1,4|2,3}[F(x_1|x_2, x_3), F(x_4|x_2, x_3)] \\
&\cdot c_{2,5|3,4}[F(x_2|x_3, x_4), F(x_4|x_3, x_4)] \\
&\cdot c_{1,5|2,3,4}[(F(x_1|x_2, x_3, x_4), F(x_5|x_2, x_3, x_4)]
\end{aligned} \tag{2.64}
$$

C-vines

Canonical vines are not as visually pleasing as D-vines when graphed, however, their structure is still very simple, as can be seen in Figure 2.8. The reason for this is that each tree T_j has one node in the centre that is connected to all the $d-j$ other nodes. This structure can therefore be a good fit if one variable has a particularly strong interaction with the others. If located, it can then be chosen as the centre of that tree, like the leftmost variables in each tree in Figure 2.8. However, unlike the D-vine, each new layer of the tree is not determined by the previous, but the next centre node can be chosen among the copulas from the previous layer. It is therefore, natural to choose the centre node that gives the best fit. However, the centre chosen will influence the possibilities for the next, which can lead to a suboptimal structure for the whole tree, in case a greedy procedure is used. The density corresponding to a canonical vine is as follows:

$$
\begin{aligned}
f(x_1, \ldots, x_d) = &\prod_{k=1}^{d} f(x_k) \prod_{j=1}^{d-1} \prod_{i=1}^{d-j} c_{j,j+i|1,\ldots,j-1} \\
&[F(x_j|x_1, \ldots, x_{j-1}), F(x_{j+i}|x_1, \ldots, x_{j-1})].
\end{aligned} \tag{2.65}
$$

and for the example for the C-vine in Figure 2.8, its distribution is given by:

$$
\begin{aligned}
f(x_1, x_2, x_3, x_4, x_5) = &\prod_{k=1}^{5} f(x_k) \prod_{j=1}^{5-1} \prod_{i=1}^{5-j} c_{j,j+i|1,\ldots,j-1} \\
&[F(x_j|x_1, \ldots, x_{j-1}), F(x_{j+i}|x_1, \ldots, x_{j-1})] \\
&= f_1(x_1) \cdot f_2(x_2) \cdot f_3(x_3) \cdot f_4(x_4) \cdot f_5(x_5) \\
&\cdot c_{1,2}(F_1(x_1), F_2(x_2)) \cdot c_{1,3}(F_1(x_1), F_3(x_3)) \\
&\cdot c_{1,4}(F_1(x_1), F_4(x_4)) \cdot c_{1,5}(F_1(x_1), F_5(x_5)) \\
&\cdot c_{2,3|1}(F(x_2|x_1), F(x_3|x_1)) \cdot c_{2,4|1}(F(x_2|x_1), F(x_4|x_1)) \\
&\cdot c_{2,5|1}(F(x_2|x_1), F(x_5|x_1)) \\
&\cdot c_{3,4|1,2}(F(x_3|x_1, x_2), F(x_4|x_1, x_2)) \\
&\cdot c_{4,5|1,2}(F(x_4|x_1, x_2), F(x_5|x_1, x_2)) \\
&\cdot c_{4,5|1,2,3}(F(x_4|x_1, x_2, x_3), F(x_5|x_1, x_2, x_3))
\end{aligned} \tag{2.66}
$$

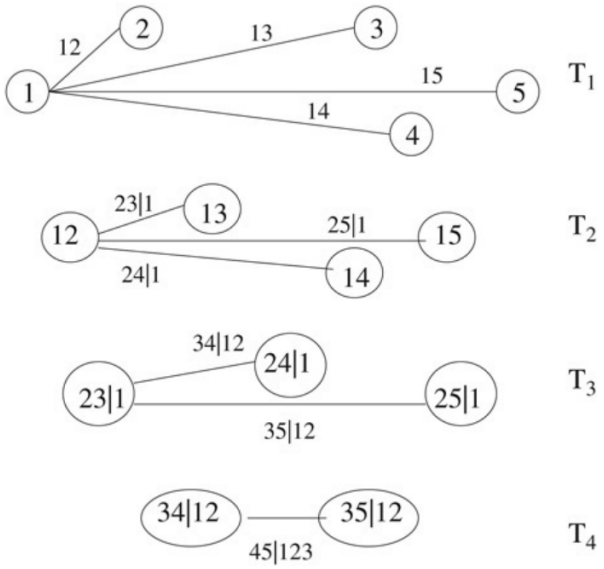


Figure 2.8: Example of Canonical vine (C-vine) with 5 variables. (The figures are taken from Aas et al. 2009)

To choose the optimal D-vine, the first tree is determined by the order that maximises the sum of the Kendall's tau value. And as stated before, this defines the whole vine.

R-vines

The regular vine is a special case of a vine that embraces many pair-copula decomopositions (Aas et al. 2009). It does not have the additional restrictions of the C- and D-vine, making Figure 2.8 and Figure 2.7 also R-vines. Regular vines can however, take many more shapes as shown in Figure 2.9. This added flexibility is advantageous as it makes it possible to build more advanced models at the cost of more complexity; a cost that becomes increasingly large with the number of variables, which will be looked into deeper in the next section. The R-vine definition is as follows:

Definition 22 (Dissmann et al. 2013)**.** *A vine $\mathcal{V}$ with $d-1$ trees is a* regular vine *on d elements if the following conditions are met:*

- *T_1 is a tree with nodes $N_1 = \{1, \ldots, d\}$ and a set of edges denoted E_1.*
- *For $i = 2, \ldots, d-1$, T_i is a tree with nodes $N_i = E_{i-1}$ and a set of edges E_i.*
- *For $i = 2, \ldots, d-1$, $\{a, b\} \in E_i$ with $a = \{a_1, a_2\}$ and $b = \{b_1, b_2\}$, it must hold that $\#(a \cap b) = 1$, where $\#$ denotes the cardinality (the number of nodes) of a set.*

Configurations

Pair copulas can be configured in many different ways, however for the minimal case with two dimensions, there is only one configuration, the pair copula itself. When the number of dimension increase the number of configurations increase exponentially. For three-dimensional case, there are 3 and the four-dimensional case has 24.

Luckily, C-vines and D-vines are the majority of the vines for lower dimensions. In fact, for the three-dimensional case, all the configurations are both C- and D-vines.

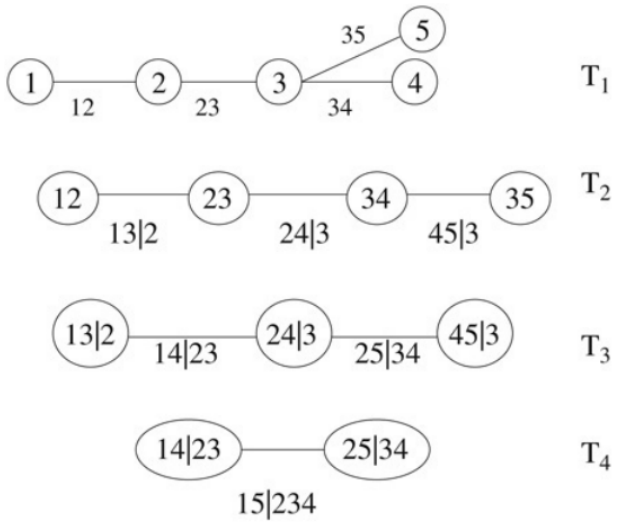


Figure 2.9: Example of a Regular vine (r-vine) with 5 variables. (The figures are taken from Aas et al. 2009)

When there are 4 variables, there are 12 different decompositions of both D-vines and C-vines and none of them are overlapping, and there are no other R-vine decompositions.

In the five-dimensional case there are 240 vine copula decompositions where half are either D- or C-vines. In fact, for the general d-dimensional case there are $d!/2$ decompositions of both C- and D-vines Aas et al. (2009). As this thesis will use seven dimensions for the empirical study, there is 2520 configurations of both C- and D-vines, all nonoverlapping, in addition to the rest of the R-vines.

Conditional independence

As seen above, pair copula becomes large quite fast. As each tree is dependent on the previous, the notion of conditional independence can sometimes simplify some of these pair vine constructions.

If X_1 and X_3 are independent given X_2, the copula conditioning on X_2 is just equal to 1:

$$\begin{aligned} f(x_1, x_2, x_3) &= f_1(x_1) \cdot f_2(x_2) \cdot f_3(x_3) \cdot c_{1,2}[F_1(x_1), F_2(x_2)] \cdot c_{2,3}[F_2(x_2), F_3(x_3)] \\ &\quad \cdot c_{1,3|2}[F_1(x_1|x_2), F_3(x_3|x_2)] \\ &= f_1(x_1) \cdot f_2(x_2) \cdot f_3(x_3) \cdot c_{1,2}[F_1(x_1), F_2(x_2)] \cdot c_{2,3}[F_2(x_2), F_3(x_3)] \end{aligned} \tag{2.67}$$

This notion can of course be generalised to higher orders:

$$c_{x_i,x_j|\mathbf{v}}[F(x_i|\mathbf{v}), F(x_j|\mathbf{v})] = 1, \tag{2.68}$$

where $\mathbf{V}$ is a vector of variables and both X and Y are single variables.

This can be applied on the pair copula construction to simplify it. This, however is dependent on the joint densities taking advantage of this conditioning, otherwise all pair-copulas are needed. As seen if Equation (2.67), under the same conditions, is instead written:

$$\begin{aligned} f(x_1, x_2, x_3) &= f(x_2|x_1, x_3) \cdot f(x_1|x_3) \cdot f_3(x_3) \\ &= c_{1,2|3}[F(x_1|x_3), F(x_2|x_3)] \cdot c_{2,3}[F_2(x_2), F_3(x_3)] \cdot f_2(x_2) \\ &\quad \cdot c_{1,3}[F_1(x_1), F_3(x_3)] \cdot f_1(x_1) \\ &\quad \cdot f_3(x_3), \end{aligned} \tag{2.69}$$

where none of the copulas are conditional independent.

Let some variables have some arbitrary dependence between each other. Then, let the first tree in a vine be created by connecting variables that have high dependence between each other. Then, for the second tree, the dependence between two nodes are conditioned on another node, which can only decrease the dependence between the two. This makes it reasonable to assume that the dependence between nodes on average decreases for each level of the vine. If so, the chance of getting independent nodes increase and can therefore simplify the construction. However, copulas can also be assumed independent for simplification reasons, in that case an approximation of the error is $c_{x_i,x_j|\mathbf{v}}[F(x_i|\mathbf{v}, F(x_j|\mathbf{v})] - 1$. Meaning if some pair copulas are close to independent, making this assumption can simplify the vine without adding to much error.

2.7.3 Dependence

Linear correlation is the most used measure for dependence. However, linear correlation is not a copula-based measure of dependence and can therefore, be misleading. Kendall's tau is a good alternative for a measure of dependence that measures the strength and direction of association between two variables based on the ranks of data, rather than their raw values. One of its advantages is that it is a non-parametric statistic, which means it does not assume that the data follows a specific distribution. Due to it being a rank-based measure, it is less sensitive to outliers compared to parametric dependence coefficients like Pearson's correlation. To define Kendall's tau, it is necessary to first introduce the following two concepts, concordance and discordance.

Definition 23. *(Embrechts, Lindskog and McNeil 2001)* Concordance and discordance *are as follows:*

Let $(x_1, y_1)^T$ *and* $(x_2, y_2)^T$ *be two observations from a continuous random vector* $(X, Y)^T$*. Then two observations are concordant if* $(x_1 - x_2)(y_1 - y_2) > 0$ *and discordant if* $(x_1 - x_2)(y_1 - y_2) < 0$

With this defined, Kendall's tau is a dependence measure that uses the difference in probability for concordant and discordant pairs. More formally, Kendall's tau has the following definition:

Definition 24. *(Embrechts, Lindskog and McNeil 2001)* Kendall's tau *for the random vector* $(X, Y)^T$ *is defined as*

$$\tau(X, Y) = Pr[(X_1 - X_2)(Y_1 - Y_2) > 0] - Pr[(X_1 - X_2)(Y_1 - Y_2) < 0] \tag{2.70}$$

However, if $(X, Y)^T$ also has the copula $C(U, V)$, where $U, V \sim Unif(0, 1)$, it can be shown that (Embrechts, Lindskog and McNeil 2001)

$$\tau(X, Y) = 4\,\mathrm{E}\,[C(U, V)] - 1. \tag{2.71}$$

As Kendall's tau is based on a difference in probability, it can take any value in $[-1, 1]$. This value can give insights about a copula, where, if the value is the lower limit, the copula becomes the Fréchet-Hoeffding lower bound, and the upper bound if Kendall's tau is 1 (Embrechts, Lindskog and McNeil 2001):

$$\tau(X, Y) = 1 \quad \iff C = M \tag{2.72}$$

$$\tau(X, Y) = -1 \quad \iff C = W \tag{2.73}$$

Tail dependence is a concept that relates to the dependence between values in the tails of a bivariate distribution. This concept is relevant for understanding the relationship between extremes in continuous random variables X and Y, and it is a fundamental property of copulas. The tail dependencies does not have to be the same in the two tails however. For the upper tail the dependence is:

Definition 25. *(Embrechts, Lindskog and McNeil 2001) The* upper tail dependence *of a vector of continuous random variables $(X, Y)^T$ with the marginal distribution functions F and G is:*

$$\lambda_U = \lim_{u \nearrow 1} (P(Y > G^{-1}(u)) | X > F^{-1}(u)), \tag{2.74}$$

given that $\lambda_U \in [0, 1]$.

If $\lambda_U = 0$, X and Y are asymptotically independent in the upper tail, otherwise they are asymptotically dependent. The lower tail dependence λ_L can be defined in a similar way:

Definition 26. *The* lower tail dependence *of a vector of continuous random variables $(X, Y)^T$ with the marginal distribution functions F and G is:*

$$\lambda_U = \lim_{u \searrow 1} (P(Y \leq G^{-1}(u)) | X \leq F^{-1}(u)), \tag{2.75}$$

given that $\lambda_U \in [0, 1]$.

2.7.4 Copula families

When constructing a copula, there are many types to choose from. Most of them belong to a specific family of copulas with up to two parameters to control its properties. Due to the extent of the mathematical characteristics of different copulas, this thesis will waive explaining the mathematics in detail, but rather focus on the properties of copula themselves. Some of the properties of copulas that make them different are symmetry, tail dependencies and rotations.

Elliptical copulas

Elliptical copulas are simply the class of copulas directly derived from their elliptical distributions, which include the multivariate Gaussian and the multivariate Student's t-distributions. As elliptical copulas are symmetric, they have the same tail distribution for the lower and upper tail (Embrechts, Lindskog and McNeil 2001). Elliptical distributions can be described, among others, by the R matrix, called the linear correlation matrix of $\mathbf{X}$. The R matrix is defined as

$$R_{i,j} = \frac{\Sigma_{i,j}}{\sqrt{\Sigma_{i,i}\Sigma_{j,j}}}, \tag{2.76}$$

Where Σ is a $d \times d$ non-negative definite, symmetric matrix.

The Gaussian copula is the copula of the d-variate normal distribution. It is mathematically convenient, but has however no tail dependence, which can make it underestimate tail risk. Simulations from the Gaussian copula in Figure 2.10 show how it has an almost perfect elliptic shape with no sign of tail dependencies. In the bivariate case using the linear correlation matrix R, the Gaussian copula is

$$\begin{aligned} C_R^{Ga}[u,v] &= \Phi_R^2(\Phi^{-1}(u), \Phi^{-1}(v)) \\ &= \int_{-\infty}^{\Phi^{-1}(u)} \int_{-\infty}^{\Phi^{-1}(v)} \frac{1}{2\pi(1-R_{1,2}^2)^{\frac{1}{2}}} \exp\Big(- \frac{s^2 - 2R_{1,2}st + t^2}{2(1-R_{1,2}^2)} \Big) \, \mathrm{d}s \, \mathrm{d}t \end{aligned} \tag{2.77}$$

Here, Φ_R^2 is the joint distribution function for the bivariate standard normal distribution function with linear correlation matrix R. Φ^{-1} is the inverse function of the univariate standard normal distribution (Embrechts, Lindskog and McNeil 2001).

The t-copula is for the bivariate Student's t-distribution and has the advantage of having tail dependence:

$$\lambda_U = \lambda_L = 2 \lim_{x \to \infty} P(X_2 > x | X_1 = x) = 2\bar{t}_{\nu+1}\Big(\frac{\sqrt{\nu+1}\sqrt{1-R_{1,2}}}{\sqrt{1+R_{1,2}}} \Big), \tag{2.78}$$

where $\bar{t}$ is the survival Student's t-distribution introduced in Section 2.7.5. The tail is of course equal for both tails because of symmetry. Its dependence increases with $R_{1,2}$ and decreases in ν. The tail dependence is visible on simulations from the t-copula in Figure 2.10 by having several realisations in the tails, especially in the lower one.

With shape parameter ν, the t-copula can be written as:

$$\begin{aligned} C_{\nu,R}^t[u,v] &= t_{\nu,R}^2(t_\nu^{-1}(u), t_\nu^{-1}(v)) \\ &= \int_{-\infty}^{t_\nu^{-1}(u)} \int_{-\infty}^{t_\nu^{-1}(v)} \frac{1}{2\pi(1-R_{1,2}^2)^{\frac{1}{2}}} \Big(1 + \frac{s^2 - 2R_{1,2}st + t^2}{\nu(1-R_{1,2}^2)} \Big)^{-\frac{\nu+2}{2}} \, \mathrm{d}s \, \mathrm{d}t. \end{aligned} \tag{2.79}$$

Where $t_{\nu,R}^2$ is the distribution function for $\sqrt{\nu}\mathbf{Y}/\sqrt{S}$, where $S \sim \chi_\nu^2$ and $\mathbf{Y} \sim \mathcal{N}_2(\mathbf{0}, R)$ are independent. Here, $\mathbf{0}$ is a column vector of zeroes and t_ν are the margins of $t_{\nu,R}^2$ (Embrechts, Lindskog and McNeil 2001).

Elliptical copulas has the advantage of being closely connected to multivariate distribution functions and their radial symmetry simplifying them. The latter, however, is also a drawback as it restricts the shapes.

Archimedean copulas

Archimedean copulas are a popular family of copulas known for their flexibility and ease of use in modelling complex dependency structures. It is specifically relevant for financial data, since several of the Archimedean copulas can model asymmetries, that elliptical copulas cannot. All commonly encountered Archimedean copulas also have closed form expression, contrary to the elliptical family (Embrechts, Lindskog and McNeil 2001). There are some drawbacks when using Archimedean copulas for multidimensional distributions, however, as this thesis only makes use of pair copulas, they are not of any hindrance.

Archimedean copulas are constructed using a generator function ϕ, which is a continuous, strictly decreasing, and convex function that maps the interval $[0,1]$ onto $[0,\infty]$. This construction allows for a wide variety of dependence structures. More formally:

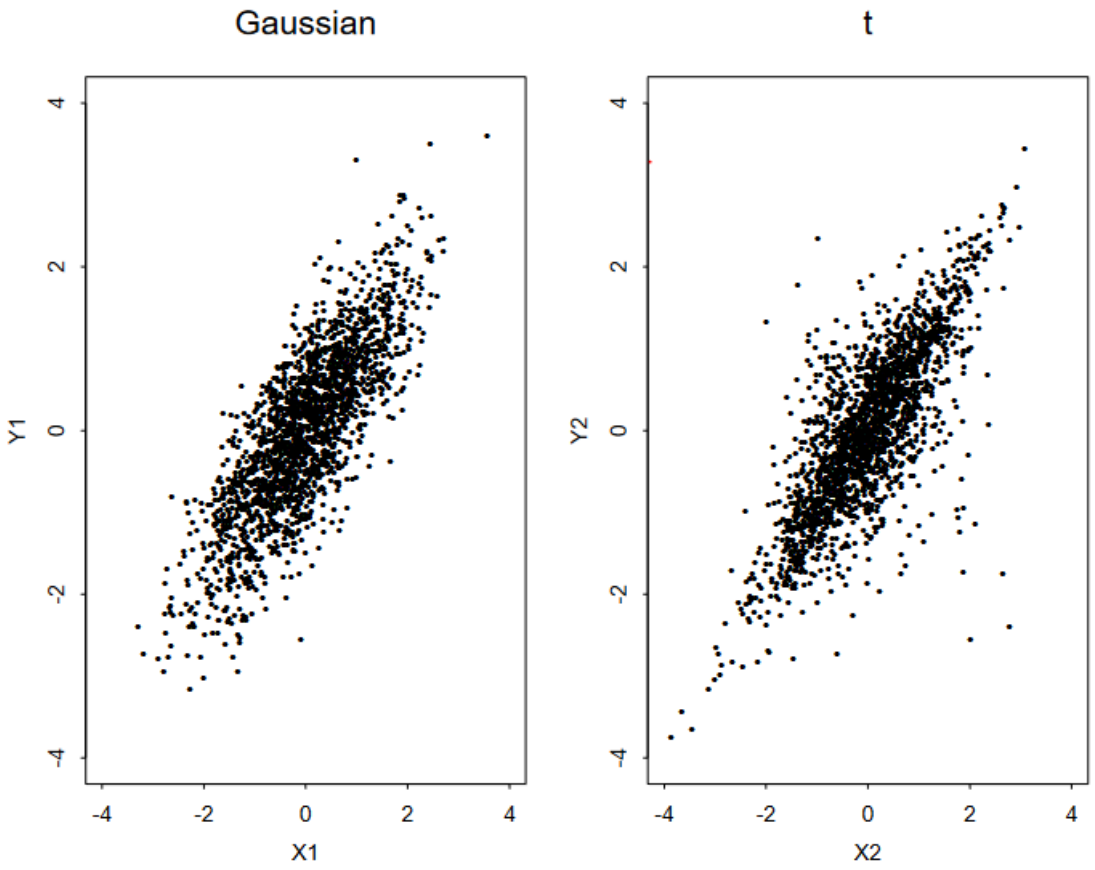


Figure 2.10: Samples from the Gaussian and t-copula. Draw attention on the heavier tails of the t-copula. (Figure taken from Embrechts, Lindskog and McNeil 2001)

Definition 27. *(Embrechts, Lindskog and McNeil 2001) An* Archimedean copula *has the form of eq.* (2.80)*, if the following is assumed:*

Let ϕ *be a continuous, strictly decreasing function from* $[0,1]$ *to* $[0,\infty]$ *such that* $\phi(1)=0$*, and let* ϕ^{-1} *be the pseudo-inverse of* ψ*. Let* C *be the function from* $[0,1]^2$ *to* $[0,1]$ *by:*

$$C(u,v)=\phi^{-1}(\phi(u)+\phi(v)), \tag{2.80}$$

Then C *is a copula if, and only if,* ϕ *is convex.*

It can also be shown that two random variables X and Y with an Archimedean copula C generated by ϕ gives the Kendall's tau of X and Y as:

$$\tau_C = 1 + 4\int_0^1 \frac{\phi(t)}{\phi'(t)}\,\mathrm{d}t \tag{2.81}$$

Tail dependence for Archimedean copulas can be expressed using their generator functions ϕ. For the Archimedean copulas, that has upper tail dependence, it is given by (Embrechts, Lindskog and McNeil 2001):

$$\lambda_U = 2 - 2\lim_{s\searrow 0}\left(\frac{\phi^{-1\prime}(2s)}{\phi^{-1\prime}(s)}\right). \tag{2.82}$$

While for the lower tail dependence it is:

$$\lambda_L = 2 \lim_{s \to 0} \left(\frac{\phi^{-1'}(2s)}{\phi^{-1'}(s)} \right). \tag{2.83}$$

Different Archimedean copula families have different characteristics and some of them are listen in Table 2.1. One of them are the Gumbel copula that has the advantage of being able to model upper tail dependencies, contrary to most other Archimedean copulas.

The Clayton copula is a copula that only has lower tail dependence for $\theta > 0$. This makes it suitable when modelling events that are likely to occur together in the lower end of their distribution, like financial markets. If θ is its lower limit of -1, the Clayton copula becomes the Fréchet-Hoeffding lower bound: $C_{-1} = W$, while if θ goes towards ∞ it becomes the upper bound: $C_\infty = M$. If $\theta > 0$, the copula simplifies to

$$C_\theta(u, v) = [u^{-\theta} + v^{-\theta} - 1]^{11/\theta} \tag{2.84}$$

The Frank copula family is the only Archimedean copula that is symmetric (Embrechts, Lindskog and McNeil 2001) and does not exhibit tail dependence. This copula family also becomes the Fréchet-Hoeffding bounds in the following way: $\lim_{\theta \to -\infty} C_\theta = W$ and $\lim_{\theta \to \infty} C_\theta = M$

Family	Copula $C_\theta(u, v)$	Generator $\phi(t)$	Limits θ
Gumbel	$\exp(-[(-\log u)^\theta + (-\log v)^\theta]^{1/\theta})$	$(-\log t)^\theta$	$[1, \infty)$
Clayton	$\max([u^{-\theta} + v^{-\theta} - 1]^{-1/\theta}, 0)$	$(t^\theta - 1)/\theta$	$[-1, \infty) \setminus \{0\}$
Frank	$-\frac{1}{\theta} \log \left(10 \frac{(e^{-\theta u} - 1)(e^{-\theta v} - 1)}{e^\theta - 1} \right)$	$-\log \frac{e^{-\theta t} - 1}{e^{-\theta} - 1}$	$\mathbb{R} \setminus \{0\}$
Joe*	$1 - [(1-u)^\theta + (1-v)^\theta - (1-u)^\theta (1-v)^\theta]^{1/\theta}$	$-\log(1 - (1-t)^\theta)$	$(0, 1]$

Table 2.1: Some of the copula families used in this thesis. Showing their copulas, generator function and limits for θ. * The Joe copula is from Joe (1997).

2.7.5 Copula modifications

Copulas can also be modified, giving them more flexibility. One such way is the notion of a survival copulas. They are just when the cumulative distribution functions are defined for values above the threshold instead of below. Formally, this can be defined for any function:

Definition 28. *For a continuous random variable x with the cumulative distribution function $F(x)$, its* survival function *$S(x)$ is:*

$$S(x) = 1 - F(x) \tag{2.85}$$

Copulas can also be rotated by 90, 180 or 270 degrees. In essence, rotating a copula changes its orientation and thereby altering the way it captures dependence, which grants flexibility. A 180 degree rotation is the same as the survival copula, simply meaning it is mirrored about both axes. This can for instance be done for the following joint distribution described by a copula,

$$C(u, v) = F(x_1, x_2) \quad , \quad u_1 = F_1(x_1), \quad u_2 = F_2(x_2), \tag{2.86}$$

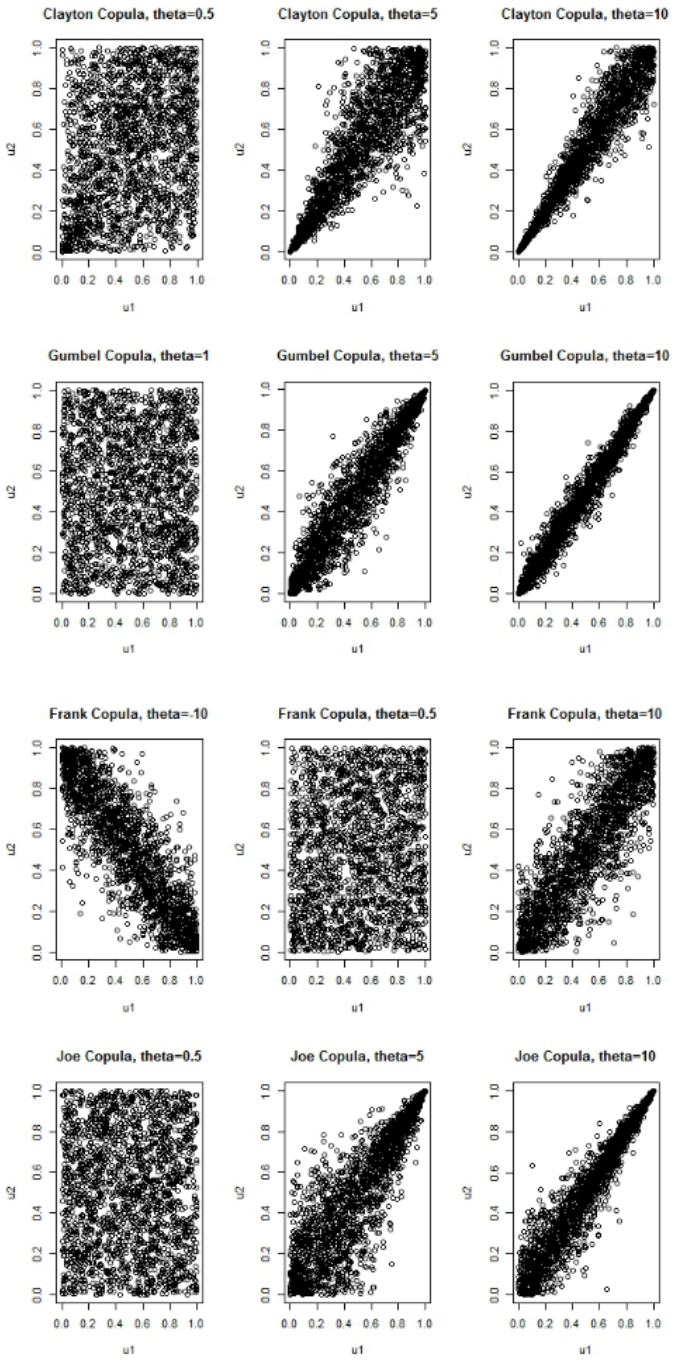


Figure 2.11: Displays the patterns of different bivariate copulas. (Figure taken from Krouthén 2015)

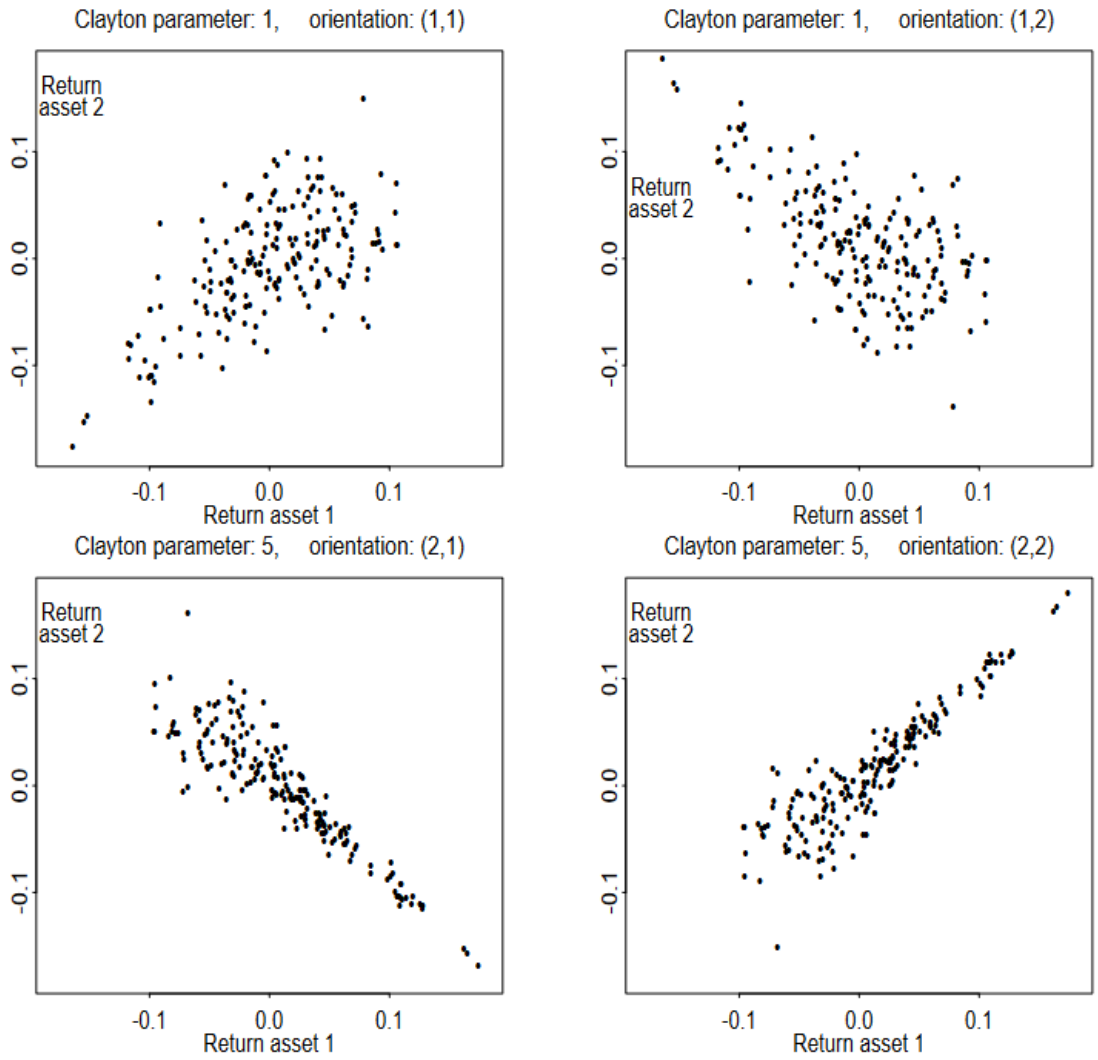


Figure 2.12: (Figure taken from 2023)

by instead using any of the following (2023):

$$u_1 = F_1(x_1), \quad 1 - u_2 = F_2(x_2) \quad \text{Orientation (1,2)} \tag{2.87}$$

$$1 - u_1 = F_1(x_1), \quad u_2 = F_2(x_2) \quad \text{Orientation (2,1)} \tag{2.88}$$

$$1 - u_1 = F_1(x_1), \quad 1 - u_2 = F_2(x_2) \quad \text{Orientation (2,2) - 180°.} \tag{2.89}$$

How the different orientations can look is shown in fig. 2.12.

Copulas can also be two parametric, which makes them more flexible. They then use two different generator functions, giving them the properties of different copulas simultaneously. This makes it possible to have both upper and lower tail dependence, without them being the same and also asymmetry and tail dependence at the same time. Some of the different copulas go under several names, some of the most common ones are listen in Table 2.2.

2.7.6 Copula simulation

Sampling from the D and C vine can be done with simple algorithms. Aas et al. (2009) suggest the following approach. Sample $w_1, \ldots, w_d$ independently from the uniform [0,1].

Name	Copula mix
BB1	Clayton-Gumbel
BB6	Joe-Gumbel
BB7	Joe-Clayton
BB8	Joe-Frank

Table 2.2: Different names for some common copulas

Then recursively set the x-values

- $x_1 = (w_1)$
- $x_2 = F^{-1}(w_2|x_1)$
- $\vdots$
- $x_d = F^{-1}(w_d|x_1, \ldots, x_{d-1})$,

where $F(x_j|x_1, \ldots, x_j - 1)$ should be chosen differently for the C- and D-vine. For the C-vine it should be chosen to be:

$$F(x_j|x_1, \ldots, x_{j-1}) = \frac{\partial C_{j,j-1|1,\ldots,j-2}[F(x_j|x_1, \ldots, x_{j-2}), F(x_{j-1}|x_1, \ldots, x_{j-2})]}{\partial F(x_{j-1}|x_1, \ldots, x_{j-2})}, \quad (2.90)$$

and for the D-vine:

$$F(x_j|x_1, \ldots, x_{j-1}) = \frac{\partial C_{j,1|2,\ldots,j-1}[F(x_j|x_2, \ldots, x_{j-1}), F(x_1|x_2, \ldots, x_{j-1})]}{\partial F(x_1|x_2, \ldots, x_{j-1})}, \quad (2.91)$$

This makes the x's dependently uniformly distributed. Then, the x's can be transformed to the desired distribution using the the inverse transform of the marginals Section 2.7.1

2.8 Robust implementation

Due to how many steps the data has to move, from raw data, to the simulations used to make efficiency frontiers in Chapter 3, mistakes can cascade through the system. To assure the results from these thesis are correct, measures has been taken to make to make each step of the process as robust and agile for changes as possible. These measures are implemented to reduce the risk of human error and traceability and have partially been implemented as a reactive measure. These modelling choices are therefore not seen fit to have in Chapter 3.

First, the model selection criteria AIC and BIC should be defined as follows:

Definition 29. *The* AIC *is an estimator for the prediction error of a statistical model for a given set of data. Let d be the amount of parameters to estimate and $\hat{L}$ the maximum likelihood function for the model. Then AIC is defined as:*

$$AIC = 2d - 2\ln(\hat{L}) \quad (2.92)$$

Definition 30. *The* BIC *is an estimator for the prediction error of a statistical model for a given set of data. Let d be the amount of parameters to estimate, $\hat{L}$ the maximum likelihood function for the model and n is the sample size. Then BIC is defined as:*

$$AIC = \ln(n)d - 2\ln(\hat{L}) \quad (2.93)$$

BIC typically prefers smaller models as the penalty for the amount of parameters is larger than 2 for $n > e^2$.

2.8.1 Grid search for ARMA-EGARCH orders

The schemes for finding the optimal orders for the ARMA-EGARCH was performed in a grid like fashion, by searching all combinations from ARMA(0,0)-EGARCH(0,0) to ARMA(1,1)-EGARCH(2,2), with a few exceptions, due to the program crashing. Both AIC and BIC was considered as a selection criterion and the optimal orders when using AIC are found in Table 2.3. However, due to AIC preferring bigger models than BIC, many of the models chosen have the same orders as the maximum in the grid search. This would call for increasing the grid search with higher orders for ARMA and EGARCH. The grid search for ARMA(1,1)-EGARCH(2,2) checks roughly $2 \times 2 \times 3 \times 3 = 36$ combinations per market. Increasing it to check up to ARMA(2,2)-EGARCH(3,3) would increase this number to $3 \times 3 \times 4 \times 4 = 144$. However, since there are many more aspects of this analysis, increasing the run time four-fold for this step was not deemed productive, arguing for the use of BIC instead of AIC.

Inspecting the order found on the basis of BIC Table 3.1 shows only one market with any ARMA order, ARMA(0,1) for the US. There is also only one market that hits the highest order for the EGARCH order in JP with EGARCH(1,2). This supports the arguments that the upper boundaries of the search are sufficient for this study.

The best models found with BIC are shown in Table 3.1, showing simpler models, not pushing the upper limit of the orders tested, giving a more robust selection criterion in these cases. The fact that the ARMA orders are much lower than the EGARCH orders aligns well with the initial assumptions for financial data stated in Chapter 2.

Country	CA	FR	GE	IT	JP	UK	US
ARMA(p,q)	(1,0)	(1,1)	(1,1)	(1,1)	(0,1)	(0,1)	(1,1)
EGARCH(p,q)	(2,1)	(2,2)	(2,2)	(1,2)	(2,1)	(2,2)	(2,1)

Table 2.3: The ARMA and EGARCH parameter for the whole time series for AIC. This is not used in this study. (1999-2019)

2.8.2 Unstable Skewed Student's t-distribution

One major problem with using the skewed Student's t-distribution for the residuals was how unstable the results was. This was not a big issue for the full time period, but affected the four shorter time series, most likely due to the fact that they have fewer data points. The fitting function would often converge with a set of shape and skew parameters much closer to the normal distribution than what plots of the series would indicate. Fitting a skewed Student's t-distribution on the residuals alone, not simultaneously with the ARMA-EGARCH, gave more reasonable values, with parameters in line with the simultaneous fit for whole series. As such, all series were compared to a standalone distribution fit to assure they were found to have a reasonable fit.

2.8.3 Simulation amount

The number of simulations that are sufficient is not an easy question to answer. However, as the complete model in the end will be evaluated by the CVaR on 1%, the number of simulations below 1% should not be so small that it becomes unstable. The value of 20000 simulations were therefore decided for, not choosing a higher value due to having to do this process for 3 vines and 5 time frames.

2.8.4 Optimising for the efficiency frontier

To find the optimal weights, an objective function is constructed, which is minimised using the numerical optimisation algorithm implemented as nlmin in R. However, the weights should sum to 1, so a penalty is given for weights that deviate from this, to stabilise the optimisation. However, this only grants the single point that minimises the CVaR, and not the whole frontier. Therefore, a another penalty term is set for the deviation from an expected return. It is then possible to perform the optimisation for a sequence of expected returns to create the frontier. An example of the final results of the frontiers are displayed in Figure 3.12a. The optimising procedure is found in Appendix B.4 where the CVaR_func function is the objective function that is optimised, showing how the penalties are implemented.

Chapter 3

Empirical study

The presented measures and methods from Chapter 2 can now be applied to the empirical study of this thesis for financial data of the G7 countries. As already mentioned above, financial assets have serial dependence, heavy tails and skewed distributions. This thesis then tries to model these properties, and uses that to predict the next times step, which is a week in this thesis. Then, each of the stock indexes are put together in a portfolio, with the goal of finding the optimal weights for the portfolio.

Nguyen and Liu (2023) is the inspiration for the approach of this thesis. Their approach is to make a model, applying ARMA-GARCH models for each country, with skewed Student's t-distributions and copulas. Then simulating from that model and find the optimal weights for a portfolio containing the stock indexes of the G7-countries. This thesis aims to use their work as a baseline and build up on it further by using the more flexible R-vine. This process requires multiple layers of models, calculated in sequences, to be able to obtain the best final result. Therefore, inspecting and tuning each part of the final model is of the most interest, as it gives insights into each part of the model and how the parts interact. In the course of the presentation of this thesis' progress, deviations from Nguyen and Liu 2023 work, will be highlighted to enable comparison of results in a relevant manner.

3.1 Overview of the analysis

This empirical analysis makes use of the techniques from Chapter 2 to tackle the issues with portfolio prediction for G7-countries. It includes many steps performed in an order, where the output of one step is the input to the next. This can lead to the overview of the process itself becoming slightly unclear. To facilitate the understanding of the process, all of the conducted steps are listed and briefly introduced below.

- Inspect and clean data - Transform to data to log returns
- Fit ARMA-EGARCH with skewed Student's t-distributions
- Transform residuals to uniform
- Fit vine copulas to uniform residuals
- Simulate new uniform residuals from vine copula
- Transform back to skewed Student's t-distributions

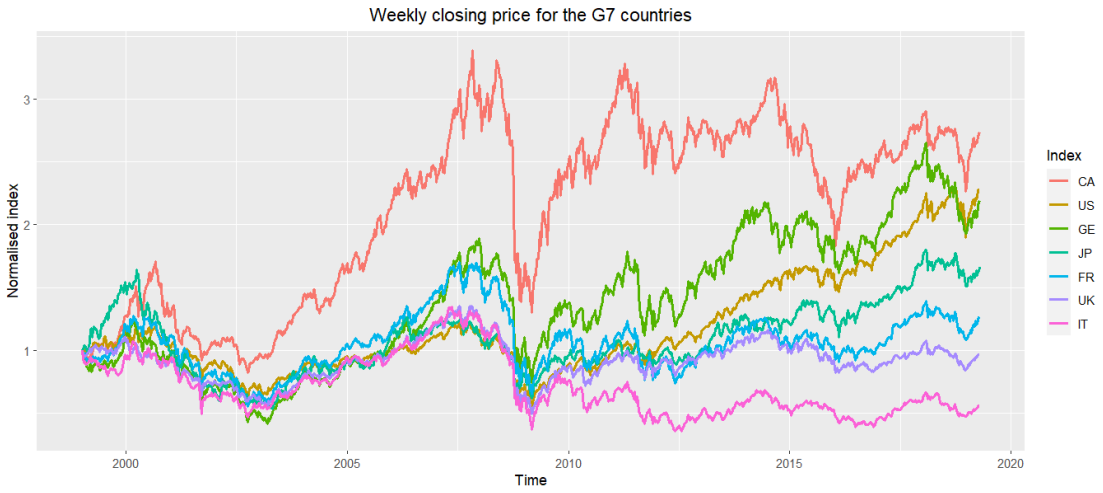


Figure 3.1: Indexes of the G7 countries. The indexes are normalised by dividing by the first value of each series. (1999-2019)

- Predict one step ahead with ARMA-EGARCH models using simulated residuals
- Select weight for portfolio

After cleaning the data and turning it into log returns, the first step of the process is to fit an ARMA-EGARCH model to each one of the seven index series. This step takes out any time dependence component, leaving the residuals to be independent and identically distributed for each respective index series. Since the residuals are skewed and heavy tailed, the skewed Student's t-distribution is used instead of the typical Gaussian assumption. As the fitting of the optimal ARMA-EGARCH models is done simultaneously with the parameters for the skewed distribution, a tuning scheme is used in a grid like fashion. The residuals from the optimal ARMA-EGARCH models then need to be transformed into a uniform distribution. These uniform residuals are then used to find the optimal copulas for the C-, D- and R-vines.

To then be able to generate samples for one step ahead for the original data, the process needs to be done in reverse like so:

- Simulate new uniform residuals with dependencies from the vine copulas
- Transform the residuals back into their skewed Student's t-distributions
- Use as input for their respective ARMA-EGARCH models to generate 7-dimensional log returns
- Given some weights, the portfolio has a univariate distribution

However, this distribution is conditioned on the set of weights for the different indexes, which as a last step requires the weights to be optimised. This leads to the creation of efficiency frontiers, curves showing the expected returns and risk for the frontier of optimal portfolios.

This process will first be presented for the period from 1999 until 2019, which will be sometimes referred to as the "full period". However, this time frame is also split into four disjoint subsections, which will receive the same analysis to better capture the differences for each time period.

3.2 The data

The data used for this empirical study are historical stock index values for the G7 countries. These are the S&P/TSX Composite Index for Canada (CA), the CAC 40 for France (FR), the DAX40 (also known as DAX30) for Germany (GE), the MSCI Italy Index for Italy (IT), Nikkei Stock Average for Japan (JP), the FTSE 100 Index for the United Kingdom (UK) and the S&P 500 Stock Composite Index for the United States of America (US). The data is retrieved from Yahoo Finance, except for the FTSE 100, which is from MarketWatch. To have the data in a preferable and common format, the data was first cleaned, as described below.

As the markets are listed in their local currencies, the indexes cannot be compared one to one. To remove any effects from the currencies themselves, all series needed to be converted to a common currency. Even though the Euro is the local currency in three out of the seven markets, the US dollar is the worlds most used currency and was consequently used for this purpose. The historical currency data used to convert to the US dollar was gathered from the Norwegian Central bank, and the Norwegian NOK was, therefore, used as an intermediate currency in this conversion.

To enable the comparison of the results from this thesis to the work of Nguyen and Liu 2023, the usage of the same time frame, from January 1998 to May 2019, was preferred. However, the Euro was first introduced on the 1st of January 1999, which makes currency prices and conversions more complicated for this period. To simplify tedious conversions, this thesis uses the introduction of the Euro as the starting date. Since some markets were closed around May 2019, this thesis uses data until the 18th of April 2019 to have the same amount of data points for all series, making the data series end roughly a month earlier than Nguyen and Liu 2023.

After currency conversions, the different index series can be seen in Figure 3.1. The same plot presented in Nguyen and Liu (2023) looks slightly different however, even if accounting for the change in start point. This can be caused by the change of currencies and could be a source for any differences from Nguyen and Liu (2023).

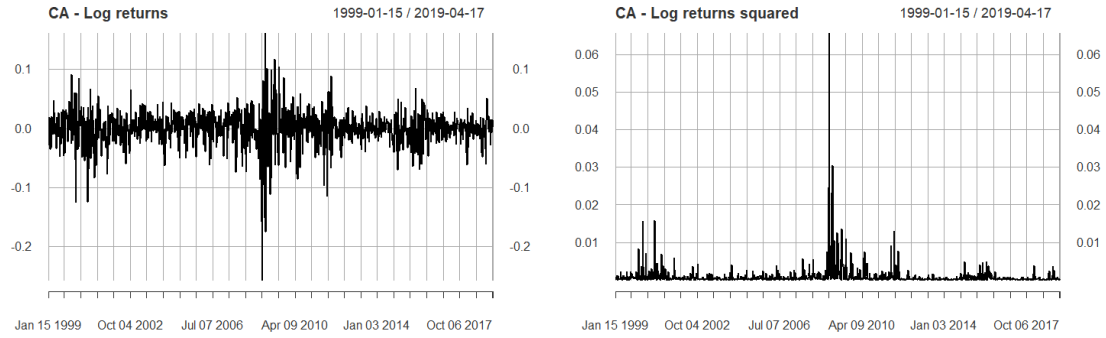
The prices used are the closing prices for each week, as daily data series are too noisy for efficient estimation and testing, and quarterly data are insufficient. This choice also helps with reducing the day-of-the-week effects, as well as from the trading hours of the different markets being different (Nguyen and Liu 2023).

Then the data were turned into log returns, with the addition that the means of each log return series were subtracted. This made it possible to fit the ARMA-EGARCH models without a mean component, which was necessary for stable convergence. After the index data has been transformed into log returns, the series look a lot more stable, as shown for the Canadian index in Figure 3.2a.

3.3 ARMA-EGARCH

While the log return series seems to be uncorrelated, the squared series is clearly not, as shown in Figure 3.2, indicating that a GARCH model could find dependencies in the series. Looking at the ACF and PACF plots in Figure 3.3 supports this claim, showing a clear structure for the squared series, while the log returns are more unclear. This indicates significant correlations on the first lags for the squared log returns, however, not for the log returns themselves.

As argued in Chapter 2 the ARMA-EGARCH model with the skewed Student's t-distribution for the residuals is a reasonable model assumption for financial data, which



a Log returns for CA

b Squared log returns for CA

Figure 3.2: The log returns and squared log returns series for CA. (1999-2019)

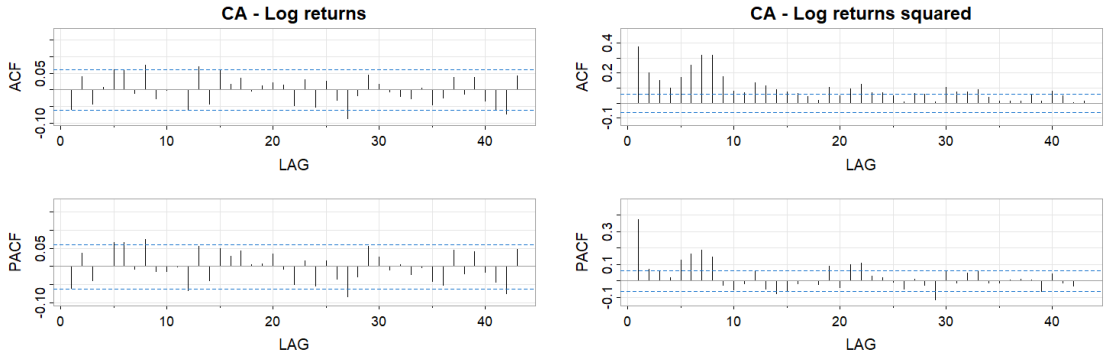


Figure 3.3: Log returns are squared log returns for CA. (1999-2019)

is why it is used in this thesis. Estimating the parameters for this model should be done simultaneously, since sequential modelling can lead to suboptimal results, due to ignoring interdependencies between the mean, volatility, and the distributional assumptions for the residuals. The rugarch-package in R makes this simultaneous fitting possible, however, the fitting process can be quite unstable and computationally demanding. There is no straightforward way of finding the optimal orders for the ARMA-EGARCH models, since the effect from some of the parameters can be drowned by the effect of others. Therefore, the BIC was chosen as the model selection criteria, as argued for in Section 2.8, and the resulting ARMA-EGARCH orders are found in Table 3.1.

Country	CA	FR	GE	IT	JP	UK	US
ARMA(p,q)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,1)
EGARCH(p,q)	(1,1)	(1,1)	(1,1)	(1,1)	(1,2)	(1,1)	(1,1)

Table 3.1: The optimal ARMA and EGARCH orders using BIC. (1999-2019)

3.3.1 Model evaluation

Further inspection of the fitted models reveal the difficulties of finding the optimal models. A closer look at different test statistics in some cases show signs of suboptimal fitted models. Given the models are fit with ARMA-EGARCH and skewed Student's t-distributions simultaneously while the tests are used to evaluate them individually, it is however, not unlikely that the test results can vary. For instance, one test can indicate

a bad fit for model 1 and a good fit for model 2, while another test could indicate the opposite. Is it therefore, still deemed more appropriate to use the BIC consequently for the model selection and instead highlight the weaknesses found by these tests on some of the fitted models.

Inspecting the p-values for the Ljung-Box tests, shown for lag 1 in Table 3.2, indicates that the ARMA structure in the models were adequately fit. Keeping the null hypothesis of the residuals having no serial correlation. For the squared residuals, however, the null hypothesis is rejected at 5% confidence level for the US and IT. This indicates that there is some lack in adequacy for the EGARCH-parts of the models to pick up on the variance structure for these market-series. Especially the fit for IT does not look too good, since the p-value displayed is significant on even 1%.

Country	CA	FR	GE	IT	JP	UK	US
Stand. residuals	0.920	0.361	0.961	0.096	0.945	0.608	0.230
Stand. squared residuals	0.113	0.474	0.114	0.001	0.398	0.856	0.0115

Table 3.2: p-values for lag 1 of weighted Ljung-Box Test. Both standardised residuals and standardised residuals squared. Null hypothesis: No serial correlation. (1999-2019)

The residuals from the fitted models are visualised with histograms in Figure 3.4, and the skew and shape parameters for the fitted models are given in Table 3.3. The histograms indicate that the assumed skewed Student's t-distributions used when fitting may have been appropriate. As expected for financial series, when looking at the tails it can be seen that the lower tails are the heavier ones. This also aligns with the values of the skew parameters given in Equation (2.47). Remember, however, that a skew parameter lower than one indicates asymmetry with negative skew (longer lower tail) in the rugarch implementation, explained in Section 2.5.3. The US has the lowest value for the skew parameter, and JP the value closest to one, which is backed up by their shapes in the histograms, where they seem to be among the most skewed and most symmetric, respectively.

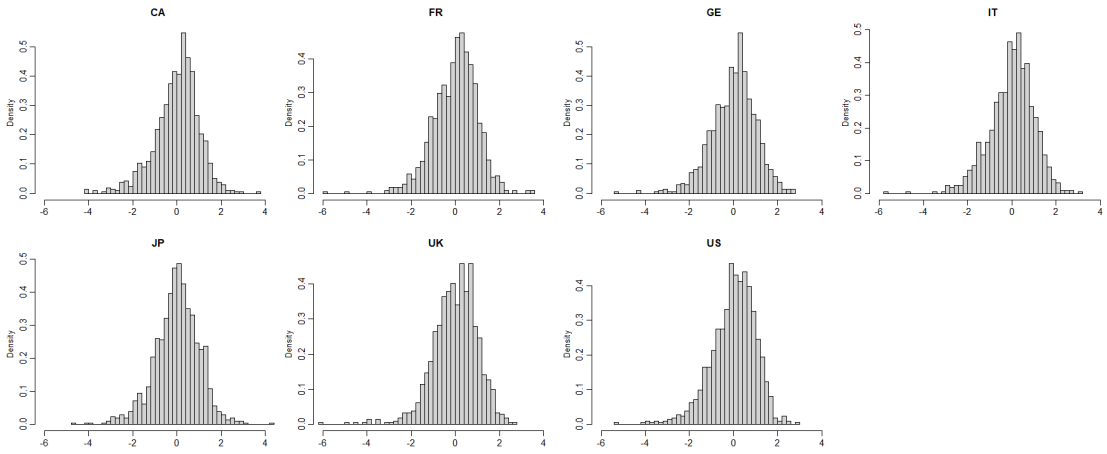


Figure 3.4: Histograms of the residual from ARMA-EGARCH fit. (1999-2019)

3.3.2 Model validation

The orders of the fitted models in Table 3.1 show that the ARMA orders are much lower than the EGARCH orders. This supports the idea that financial opportunities

Country	CA	FR	GE	IT	JP	UK	US
Skew	0.760	0.792	0.825	0.802	0.879	0.772	0.724
Shape	6.755	7.950	10.251	11.242	7.791	9.596	10.298

Table 3.3: The skew and shape parameters for the skewed Student's t-distributed residuals. (1999-2019)

are exploited, removing the mean structure, while the volatility will not disappear, as argued in Section 2.4.

With the parameters of the ARMA-EGARCH models presented in Table 3.4, taking a look at some of the specific models can give insights into how the log return series behave. The model for the US market will be used as an example, since it is the only one with any ARMA structure. It is as follows

$$(x_t - \mu) = \phi_1 x_{t-1} + w_t = \phi_1 x_{t-1} + \sigma_t \epsilon_t \tag{3.1}$$

$$\log(\sigma_t^2) = \alpha_0 + \alpha_1 g(\epsilon_{t-1}) + \beta_1 \log(\sigma_{t-1}^2) \quad \Leftrightarrow \quad \sigma_t^2 = e^{\alpha_0} (\sigma_{t-1}^2)^{\beta_1} e^{\alpha_1 g(\epsilon_{t-1})} \tag{3.2}$$

$$g(\epsilon_{t-1}) = \epsilon_{t-1} + \lambda(|\epsilon_{t-1}| - \mathrm{E}\left[|\epsilon_{t-1}|\right]), \tag{3.3}$$

where ϵ_t are iid from the skewed Student's t-distributions described above and μ is the mean subtracted before fitting. Looking at the value for the autoregressive parameter ϕ, for the US market, shows that the current log return value is slightly negatively related to last weeks log returns. However, no such effect is found for any of the other markets.

Without listing any other specific models, the parameters in the Table 3.4 can give indications for the structures found. For instance, JP has a β_1-value less than half of the others, which can be explained by it having two β's, splitting the effect over more parameters.

Country	CA	FR	GE	IT	JP	UK	US
μ	0.00095	0.00022	0.00074	-0.00054	0.00047	-0000.3	0.0007
ϕ_1	-	-	-	-	-	-	-0.119
α_0	-0.237	-0.265	-0.469	-0.180	-0.528	-0.604	-0.658
α_1	-0.077	-0.117	-0.139	-0.092	-0.126	-0.146	-0.217
β_1	0.967	0.962	0.933	0.974	0.357	0.919	0.915
β_2	-	-	-	-	0.570	-	-
λ	-2.623	-0.974	-1.215	-0.9347	-1.809	-1.349	-0.930

Table 3.4: Fitted parameters for each of the markets, including the mean value μ, subtracted from the non-centred log returns series. (1999-2019)

3.4 Copulas

Before the copulas can be fitted, the residuals need to be transformed to the uniform distribution, using the probability integral transformation introduced in Section 2.7.1. After uniformly mapping them to their probabilistic value between zero and one, the transformation of the distributions deviates some from a perfect uniform distribution, however, it is deemed sufficient for further analysis, as seen in the histograms in Figure 3.5. These uniform transformations of the residuals are then used to fit a C- and D-vine, in addition to the more flexible R-vine.

The Kendall's tau values between the pairs of marginal distributions, representing each stock market index, are shown in shown in Table 3.5. This is of particular importance for the D-vine, as explained in Section 2.7.2, due to the whole pair copula construction being defined by the ordering in the first tree, and that ordering is found by minimising the sum of Kendall's tau.

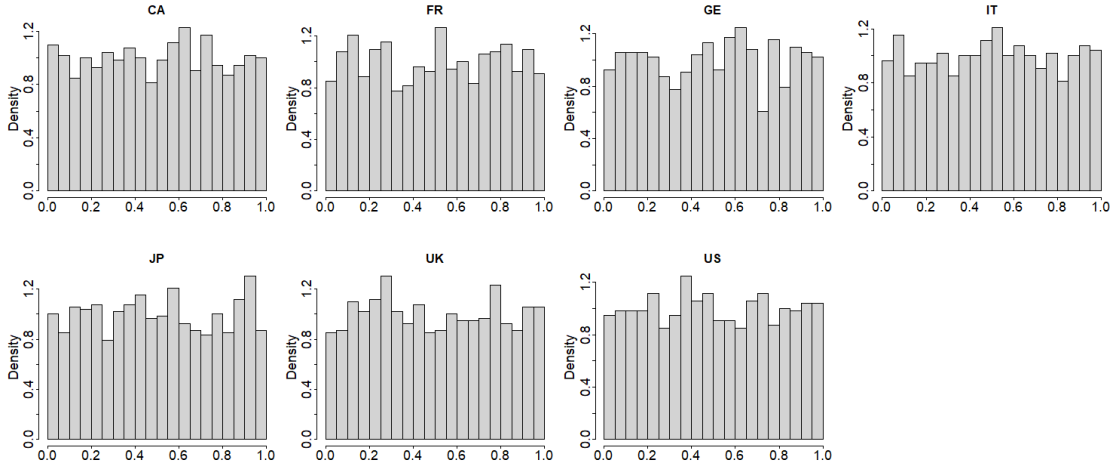


Figure 3.5: Histogram of the residuals transformed to uniform from their respective skewed Student's t-distributions. (1999-2019)

3.4.1 The overall patterns

The first tree of the different vines are illustrated in Figure 3.6, Figure 3.7 and Figure 3.8, respectively. As they show the pair copulas below the lines and Kendall's tau values above, they give a good overview and indications for each of the vine copula constructions. The complete constructions of these vine copulas can be seen in Table 3.6, Table 3.7 and Table 3.8.

The structure for the first trees for the different vines are supported by the average values for Kendall's tau in Table 3.5. The markets with the highest dependencies are for the most part in the central positions and the markets with the lowest dependencies are placed in the outer locations.

A geographical effect seems to be present for multiple reasons. Firstly, because none of the D-vine pairs are far away from each other. Secondly, when there are fewer restrictions with the R-vine, FR is still the most central node. And thirdly, four of the seven markets are European, partially explaining why the C-vine has FR as the centre. Initially, one might assume US would be in the centre, however, with its size and impact

in the world, a change in the US market is likely to lead in market movement, giving it a lower value for Kendall's tau with the other markets.

Furthermore, there are significant non-linear dependencies, as most copulas are Archimedean or Student-t copulas. Additionally, several market pairs have higher tail dependencies in the lower tail, while the rest of the market pairs have equal or no tail dependencies. This indicates higher chances of extreme falls in value transmitting to other markets, than the other way around.

Comparing the results in this thesis to those of Nguyen and Liu (2023) shows that the values for Kendall's tau are very similar, demonstrating that the models up to this point are not too far apart. The types of the copulas and the level of tail dependency, however, deviate more. This thesis is dominated by a high amount of t-copulas and lower tail dependencies compared to that of Nguyen and Liu (2023). This can indicate that the ARMA-EGARCH models, fitted in this thesis, have unified the residuals more, possibly due to removing the effects from currencies.

Contour plots for the pair copulas in the different vines are shown in Figure 3.9. Focusing on the first tree shows how the dependence structure looks very similar. Looking at the contours of the one pair copula that differs between the C- and R-vine, shows that it is very similar in structure. This makes the two models even more similar than what the already similar tree structure suggests, making it even more reasonable to expect similar results for the two vines.

	CA	FR	GE	IT	JP	UK	US
CA	1.00	0.48	0.46	0.42	0.32	0.51	0.47
FR	0.48	1.00	0.74	0.69	0.33	0.64	0.53
GE	0.46	0.74	1.00	0.64	0.33	0.59	0.52
IT	0.42	0.69	0.64	1.00	0.29	0.54	0.44
JP	0.32	0.33	0.33	0.29	1.00	0.32	0.32
UK	0.51	0.64	0.59	0.54	0.32	1.00	0.51
US	0.47	0.53	0.52	0.44	0.32	0.51	1.00
Average:	0.52	0.63	0.61	0.57	0.41	0.59	0.54

Table 3.5: Kendall's tau. (1999-2019)

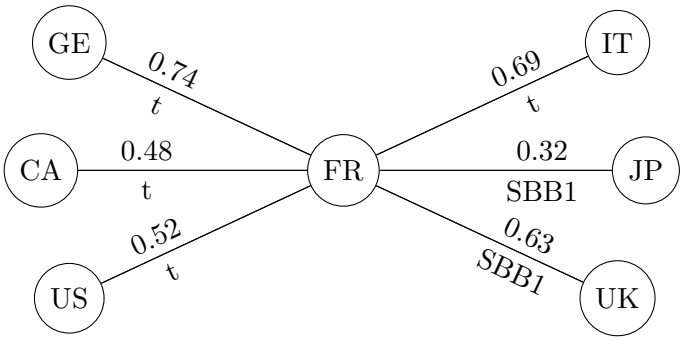


Figure 3.6: First tree for the C-vine. Kendall's tau is the value above the lines and the type of pair copula is below. (1999-2019)

Figure 3.7: First tree for the D-vine. (1999-2019)

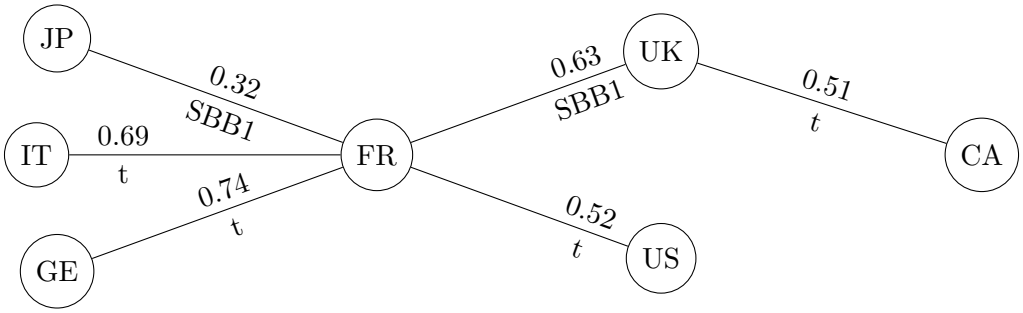


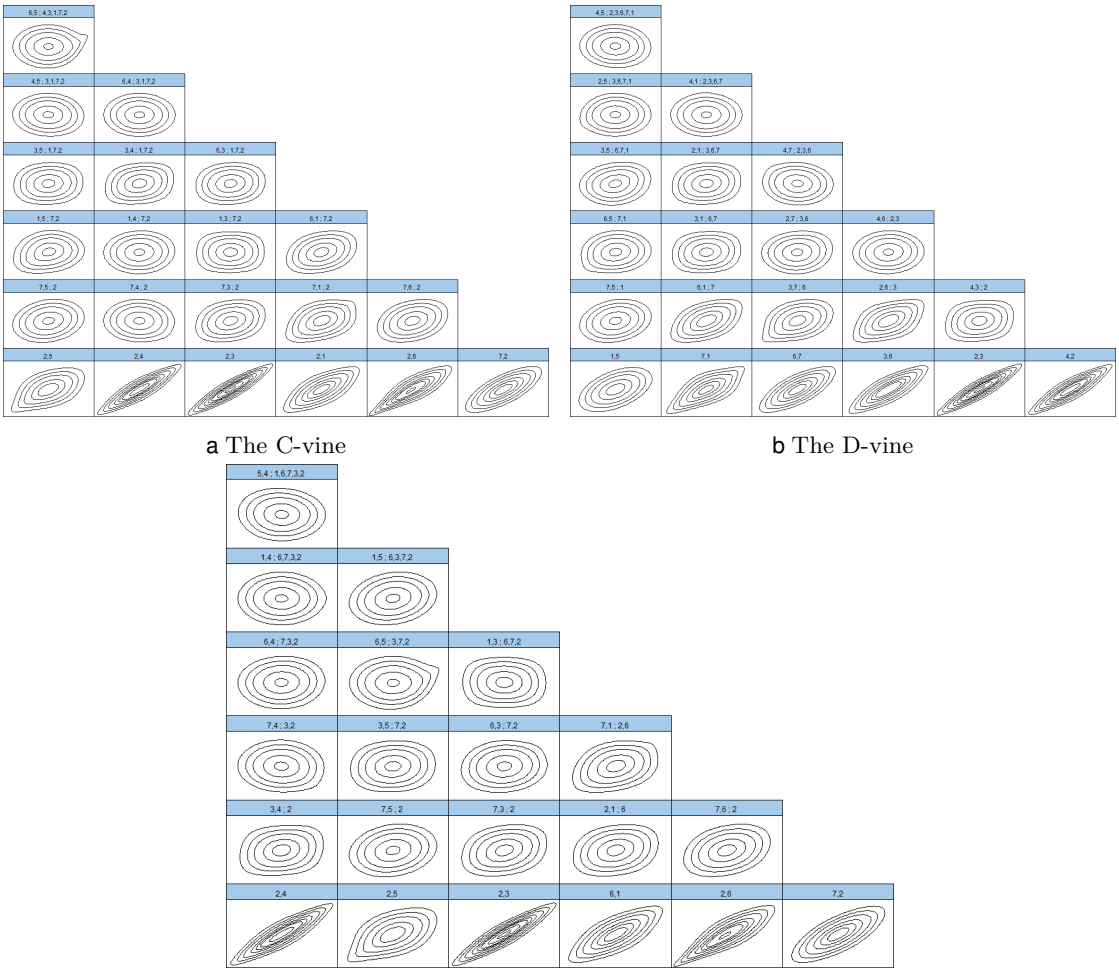
Figure 3.8: First tree for the R-vine. (1999-2019)

3.4.2 Vine specifics

Looking closer at the C-vine, the first thing to notice is how the French market is located in the middle, indicating its high dependence with the other markets. Table 3.6 lists the US as the centre market for the following tree. However, more interestingly, CA is the centre node for the third tree, while the UK is never the centre node at all, the opposite of what is the case for Nguyen and Liu (2023). As the dependencies within each tree in general decreases with the tree lever shown in Tables 3.6 to 3.8, the structures on higher levels are less impactful. Which could make the fitting less robust for the higher levels, possibly explaining the difference in positioning CA and UK.

For the D-vine case, the first tree shown in Figure 3.7 is of extra importance, as it determines the structure for the whole vine. Looking at the first tree, it seems like a very reasonable fit, with having two of the markets with least dependence, IT and JP, in each end. This implicitly makes them less influential by limiting them to only one other market to interact with, instead of two. This, however, differs from Nguyen and Liu who have FR and JP on each end with IT in between US and CA. As FR is the centre of the C-vine it would be reasonable to not have it at the end. Looking at the second tree Table 3.7 gives even more confidence in the vine being suitable, since the lowest conditioned dependencies are for the end node pairs. Since the second tree is predetermined by the first, it is not given that this will be the case.

The R-vine, with its more loose restrictions, has chosen a structure very close to the C-vine, with only CA deviating and being connected to UK. This could indicate that the optimal vine structure is very close to the C-vine found in this instance. However, this is mostly relevant for the first tree, since the next trees are conditioned, they do not have the same possible combinations anymore.



a The C-vine

b The D-vine

c The R-vine

Figure 3.9: Contour plots for the different vine copulas. The pairs in the pair copulas are listed above each window in the figures. The countries are in alphabetic order with: 1=CA, 2=FR, 3=GE, 4=IT, 5=JP, 6=UK,7=US. (1999-2019)

Tree	Pair	Copula	Param. 1	Param. 2	Kendall's tau	Tail dep.
1	FR-JP	Rot. BB1 180 deg.	0.17 (0.06)	1.37 (0.05)	0.32	0.05 0.34
1	FR-IT	t	0.88 (0.01)	6.15 (1.42)	0.69	0.53 0.53
1	FR-GE	t	0.92 (0.00)	8.00 (1.96)	0.74	0.55 0.55
1	FR-CA	t	0.69 (0.02)	7.63 (1.96)	0.48	0.24 0.24
1	FR-UK	Rot. BB1 180 deg.	0.24 (0.06)	2.43 (0.09)	0.63	0.31 0.67
1	US-FR	t	0.73 (0.01)	26.53 (18.36)	0.52	0.05 0.05
2	US-JP|FR	Gaussian	0.18 (0.03)	- -	0.11	- -
2	US-IT|FR	Rot. Joe 270 deg.	-1.05 (0.02)	- -	−0.03	- -
2	US-GE|FR	t	0.22 (0.03)	11.82 (4.39)	0.14	0.01 0.01
2	US-CA|FR	BB1	0.30 (0.06)	1.14 (0.03)	0.24	0.16 0.13
2	US-UK|FR	t	0.29 (0.03)	20.46 (13.64)	0.19	0.00 0.00
3	CA-JP|US,FR	Rot. BB8 180 deg.	1.39 (0.22)	0.89 (0.12)	0.11	- -
3	CA-IT|US,FR	Rot. Clayton 180 deg.	0.05 (0.03)	- -	0.02	0.00 -
3	CA-GE|US,FR	t	0.02 (0.03)	7.64 (1.94)	0.01	0.02 0.02
3	UK-CA|US,FR	t	0.28 (0.03)	14.04 (5.97)	0.18	0.01 0.01
4	GE-JP|CA,US,FR	t	0.07 (0.03)	13.21 (5.18)	0.05	0.00 0.00
4	GE-IT|CA,US,FR	t	0.16 (0.03)	8.27 (2.26)	0.10	0.03 0.03
4	UK-GE|CA,US,FR	t	0.09 (0.03)	13.70 (6.27)	0.06	0.00 0.00
5	IT-JP|GE,CA,US,FR	Rot. Clayton 180 deg.	-0.05 (0.04)	- -	−0.02	- -
5	UK-IT|GE,CA,US,FR	Indep.	- -	- -	0.00	- -
6	UK-JP|IT,GE,CA,US,FR	Tawn type 2	1.95 (0.36)	0.04 (0.01)	0.03	0.04 -

Table 3.6: C-vine. Tail dependence column contains upper and lower tail respectively. (1999-2019)

Tree	Pair	Copula	Param. 1	Param. 2	Kendall's tau	Tail dep.
1	CA-JP	Gaussian	0.48 (0.02)	- -	0.32	- -
1	US-CA	BB1	0.66 (0.08)	1.42 (0.05)	0.47	0.37 0.48
1	UK-US	t	0.72 (0.01)	15.25 (7.04)	0.51	0.12 0.12
1	GE-UK	t	0.81 (0.01)	8.27 (2.20)	0.60	0.34 0.34
1	FR-GE	t	0.92 (0.00)	8.00 (1.96)	0.74	0.55 0.55
1	IT-FR	t	0.88 (0.01)	6.15 (1.42)	0.69	0.53 0.53
2	US-JP|CA	Gaussian	0.21 (0.03)	- -	0.13	- -
2	UK-CA|US	t	0.44 (0.03)	9.21 (2.51)	0.29	0.08 0.08
2	GE-US|UK	Rot. BB1 180 deg.	0.10 (0.05)	1.23 (0.04)	0.23	0.00 0.24
2	FR-UK|GE	t	0.43 (0.03)	5.59 (1.11)	0.29	0.15 0.15
2	IT-GE|FR	t	0.15 (0.03)	7.28 (1.79)	0.10	0.04 0.04
3	UK-JP|US,CA	Rot. Gumbel 180 deg.	1.09 (0.02)	- -	0.08	- 0.11
3	GE-CA|UK,US	t	0.09 (0.03)	9.24 (2.69)	0.06	0.01 0.01
3	FR-US|GE,UK	Rot. Clayton 180 deg.	0.11 (0.03)	- -	0.05	0.00 -
3	IT-UK|FR,GE	Frank	0.27 (0.19)	- -	0.03	- -
4	GE-JP|UK,US,CA	BB8	1.50 (0.31)	0.78 (0.17)	0.10	- -
4	FR-CA|GE,UK,US	t	0.10 (0.03)	10.23 (3.50)	0.06	0.01 0.01
4	IT-US|FR,GE,UK	Rot. Gumbel 90 deg.	-1.04 (0.02)	- -	−0.04	- -
5	FR-JP|GE,UK,US,CA	Rot. Tawn type 2 180 deg.	1.16 (0.12)	0.08 (0.07)	0.03	- 0.03
5	IT-CA|FR,GE,UK,US	Tawn type 1	2.57 (0.94)	0.01 (0.00)	0.01	0.01 -
6	IT-JP|FR,GE,UK,US,CA	Rot. Clayton 180 deg.	-0.05 (0.03)	- -	−0.02	- -

Table 3.7: D-vine. Tail dependence column contains upper and lower tail respectively. (1999-2019)

Tree	Pair	Copula	Param. 1	Param. 2	Kendall's tau	Tail dep.
1	FR-IT	t	0.88 (0.01)	6.15 (1.42)	0.69	0.53 0.53
1	FR-JP	Rot. BB1 180 deg.	0.17 (0.06)	1.37 (0.05)	0.32	0.05 0.34
1	FR-GE	t	0.92 (0.00)	8.00 (1.96)	0.74	0.55 0.55
1	UK-CA	t	0.72 (0.01)	8.19 (2.06)	0.51	0.25 0.25
1	FR-UK	Rot. BB1 180 deg.	0.24 (0.06)	2.43 (0.09)	0.63	0.31 0.67
1	US-FR	t	0.73 (0.01)	26.53 (18.36)	0.52	0.05 0.05
2	GE-IT|FR	t	0.15 (0.03)	7.28 (1.79)	0.10	0.04 0.04
2	US-JP|FR	Gaussian	0.18 (0.03)	- -	0.11	- -
2	US-GE|FR	t	0.22 (0.03)	11.82 (4.39)	0.14	0.01 0.01
2	FR-CA|UK	t	0.21 (0.03)	12.41 (4.38)	0.14	0.01 0.01
2	US-UK|FR	t	0.29 (0.03)	20.46 (13.64)	0.19	0.00 0.00
3	US-IT|GE,FR	Rot. Joe 270 deg.	-1.06 (0.02)	- -	−0.03	- -
3	GE-JP|US,FR	t	0.07 (0.03)	13.28 (5.24)	0.05	0.00 0.00
3	UK-GE|US,FR	BB7	1.05 (0.03)	0.08 (0.03)	0.07	0.06 0.00
3	US-CA|FR,UK	BB1	0.23 (0.05)	1.11 (0.03)	0.19	0.13 0.07
4	UK-IT|US,GE,FR	Rot. Clayton 180 deg.	0.05 (0.03)	- -	0.03	0.00 -
4	UK-JP|GE,US,FR	Tawn type 2	1.56 (0.25)	0.06 (0.02)	0.05	0.05 -
4	CA-GE|UK,US,FR	t	-0.01 (0.03)	7.84 (2.02)	−0.00	0.02 0.02
5	CA-IT|UK,US,GE,FR	Tawn type 2	2.04 (0.78)	0.01 (0.00)	0.01	0.01 -
5	CA-JP|UK,GE,US,FR	Gaussian	0.17 (0.03)	- -	0.11	- -
6	JP-IT|CA,UK,US,GE,FR	Rot. Clayton 270 deg.	-0.05 (0.03)	- -	−0.03	- -

Table 3.8: R-vine. Tail dependence column contains upper and lower tail respectively. (1999-2019)

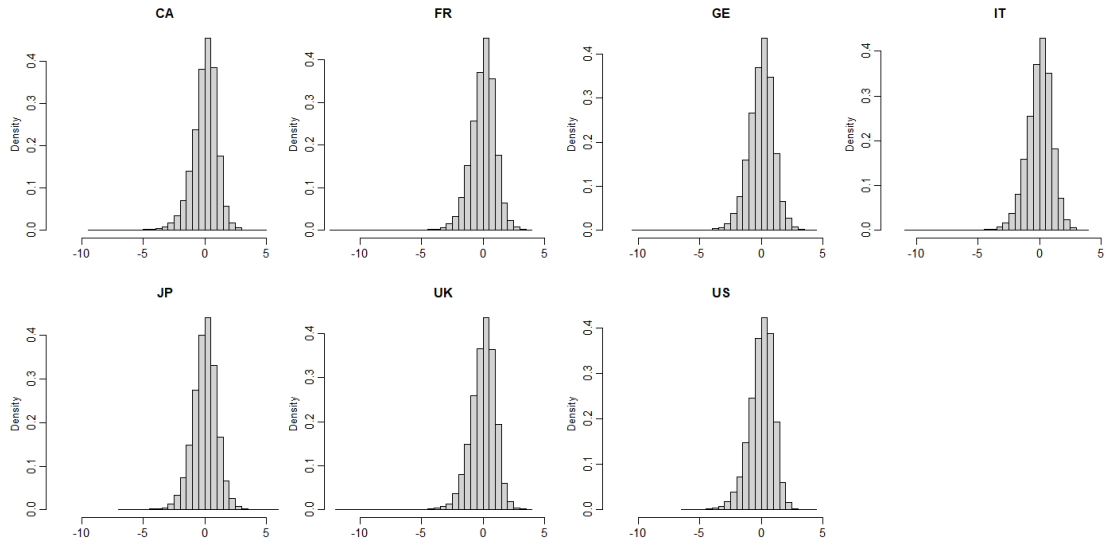


Figure 3.10: Histograms of the simulated skewed Student's t-distributed residuals from the C-vine. (1999-2019)

3.5 The other time series

The question arises, if the models are applicable for the whole series? This thesis follows the argument of Nguyen and Liu (2023), who use a non-parametric approach to find structural break points in the comovement of the log return series and split the series into four, based on their finding for break points. The break points are the 7th of April 2003, the 21st of January 2008 and the 5th of November 2012.

With now four new subseries, the same process was repeated. The most interesting results from the inspection of the models is presented in this chapter, while the full sett of results can be found in Appendix A.

First off, FR is the centre of the C-vine for all the time frames, showing how the dependencies with the other markets are strong, not only over the whole series, but for all the subseries.

For the D-vine, the pattern is mostly the same as for the whole series, with JP always being an end node. However, it is worth noting that the IT, which was an end node for the whole series, is never an end node in any of the subseries. With how high the average value for Kendall's tau was for the whole series, seen in Table 3.5, this might not be so surprising, but rather a coincident.

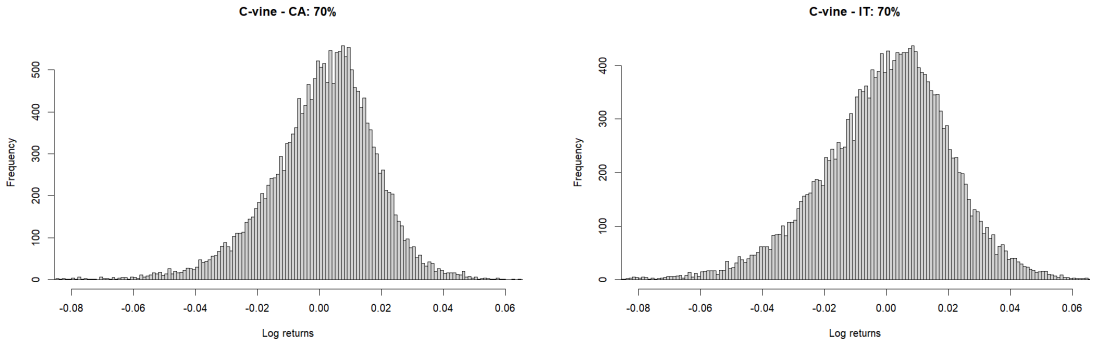
The R-vine changes between looking close to a D-vine and a C-vine for the first tree, showing how it is more flexible. This could indicate that the R-vine is more robust when fit on different series.

Looking closer at the pair copulas used for the first tree does not show drastically different results either.

3.6 Optimising portfolios

3.6.1 Simulations

When the vine copulas are fit, it is possible to simulate realisations from the 7-dimensional dependence structure. As vines simulate uniform marginal distributions, the



a CA is weighted with 70%, the other markets are 5%

b IT is weighted with 70%, the other markets are 5%

Figure 3.11: Example of the histogram of the returns for the C-vine simulations with two different set of weights. (1999-2019)

series needs to be transformed back to their respective skewed Student's t-distributions, before using them on the ARMA-EGARCH models.

Inspections of the simulated skewed Student's t-distributed residuals show that the simulated residuals distributions seem to have roughly the same shape as the original ones, as seen in the histograms in Figure 3.10. However, they seem to have sampled some much lower values than the real residual, as the x-axis goes much lower. This is not surprising, as 20000 simulations are magnitudes bigger than the original data set.

To generate simulations for the next week, the 7-dimensional residuals are used as input for their respective ARMA-EGARCH models, inserting the time dependence again, for the example in the US model in Equations (3.1) to (3.3). Their mean values μ is also added again, giving non-centred log returns.

At this point, another model should be fitted and simulated from to use as a benchmark. For the traditional mean variance approach, the dependence would be assumed to be 7-dimensional normally distributed. This would call for fitting models without any other effect than the ARMA model. However, to focus on the effect of the copula, the same residuals are used. These are used to fit a 7-dimensional normal distribution, which is simulated from and turned into log return series in the same way as for the vines. This model will from now on be referenced to as the reference model.

3.6.2 Efficiency frontiers

At this stage, we have 20000 simulations for the 7-dimensional log returns next week. However, it is not the individual series that are of interest, but a weighted portfolio of them. This then transforms the 7-dimensional log returns into a univariate distribution for the portfolio, however, this is dependent on a given set of weights. This is shown in Figure 3.11, which shows histograms of the log returns for two portfolios where the weight is 5% for all series but one, which has the remaining 70%. When putting 70% in IT, shown in Figure 3.11b, the distribution is clearly less centred around zero and has more weight longer out in the tails, than when CA has a 70% weight as shown Figure 3.11a.

To find the optimal weights of the portfolio is then the next step. Here, the idea of Markowitz (1952) comes in, that an investor should try to maximise the return for a given risk. This leads to the optimisation problem of finding an efficiency frontier, the

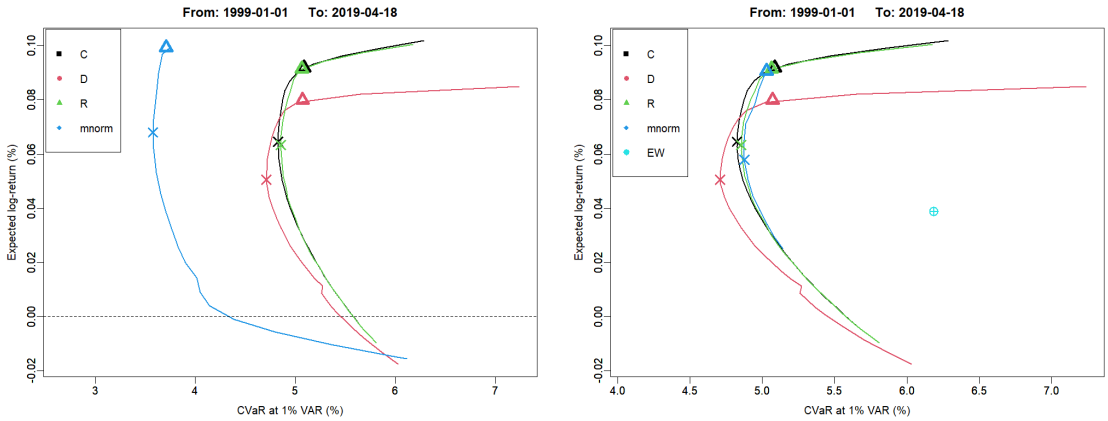
set of weights that maximises the returns for a given risk. Efficiency frontiers often plot the expected log returns against the risk measure VaR, in this paper, however, CVaR is used, as it better captures the heavy tail, as argued in Section 2.1.3.

In Figure 3.12a, the efficiency frontiers for the different vines and the benchmark model is shown. The crosses indicate where the minimum CVaR value is for each of them and the triangles, the maximum return-to-CVaR. These frontiers then show the optimal portfolios for the estimated CVaR for a range of estimated expected returns. Important to notice is how the benchmark model estimates the risk to be way lower than for the vines.

However, when comparing models for finding the optimal weights, the frontier should be compared to the same data. For this thesis, the simulations from the R-vine will be used. This was chosen to elaborate on the work by Nguyen and Liu (2023), by using vine they do not test. The weights found for the benchmark model is, therefore, used to calculate the corresponding efficiency frontier for the R-vine simulations. When the benchmark model is fit on the simulations for the R-vine, the R-vine is assumed to be the true model, meaning the benchmark is bounded by the same efficiency frontier as the R-vine. This does, however, show how the simplified benchmark performs different from the R-vine. For the rest of the efficiency frontiers in this thesis, the benchmark model is shown for the simulations from the R-vine.

Figure 3.12b shows the now adjusted efficiency frontier for the benchmark. To better put the optimal portfolios in perspective, an equally weighted portfolio on the R-vine simulations is also added.

Let us first compare the different efficiency frontiers for the vines in Figure 3.12b. It is noticeable how the R-vine is almost the same as the curve for the C-vine. This is not too strange, since the R-vine was very similar to the C-vine for the first trees in the pair copula construction. That the D-vine shows a different shape of the curve reveals how the non-linear dependencies are modelled differently. When comparing the benchmark model to the R-vine it finds almost the same optimal weights. However, before making any remarks on the benchmark model it should be inspected further through the subseries.



a mnorm on 7-dimensional multinormal simulations

b mnorm on R-vine simulations

Figure 3.12: Efficiency frontiers. The crosses are indicating minimum CVaR and the triangles the maximum of return-to-CVaR. mnorm is the the model fitted on 7-dimensional normally distributed dependence. The frontiers are all calculated for their own simulated data, except for mnorm in (b). (1999-2019)

Looking at the efficiency frontiers from the subperiods, shown in Figure 3.13, do not show any obvious trends in how the efficiency frontiers are shaped or placed in relation to one another. However, there might be slight indications that the R-vine resembles the C-vine curve slightly more than the D-vine. This could hint to the C-vine being better at modelling the dependencies than the D-vine, due to how the flexibility of the R-vine still chooses a structure close to that of the C-vine. This potential finding aligns with what Nguyen and Liu (2023) states, that the C-vine "is statistically superior" to the D-vine.

The optimal weights for the portfolios that minimise the CVaR for the different models are shown in Table 3.9. The weights found are very similar and explains why the frontiers are so close to each other.

Seeing the results as a whole, the frontiers show very similar behaviour for all the vines. A reason for this is the way this study is conceived. The simulations of the one step ahead, in this thesis, are the next step at the end of the time frame. This makes the state in that singe time step very important. If the variance was high for some markets and low for others, the EGARCH model would scale the variance of the marginal distributions differently. This can lead to some markets being very unfavourable in that exact time step and others not. It is then a lot more favourable to have weights in some markets than others, making the distribution of the portfolio, very close to some few individual market distributions. When some weights in the portfolios are exactly zero, the efficiency frontier has a much smaller space to search for solutions. This then leads to the different vines and the benchmark model to have the optimal weights very close to another.

Dependence model	CA	FR	GE	IT	JP	UK	US
C	6.3	0.0	0.0	8.8	23.5	29.4	31.4
D	0.0	0.0	0.0	11.9	28.6	32.1	27.1
R	0.0	0.0	0.0	6.49	21.7	27.0	44.6
Benchmark	7.8	0.0	0.0	2.9	19.1	24.6	45.3

Table 3.9: Weights for the portfolios that minimise CVaR- (1999-2003)

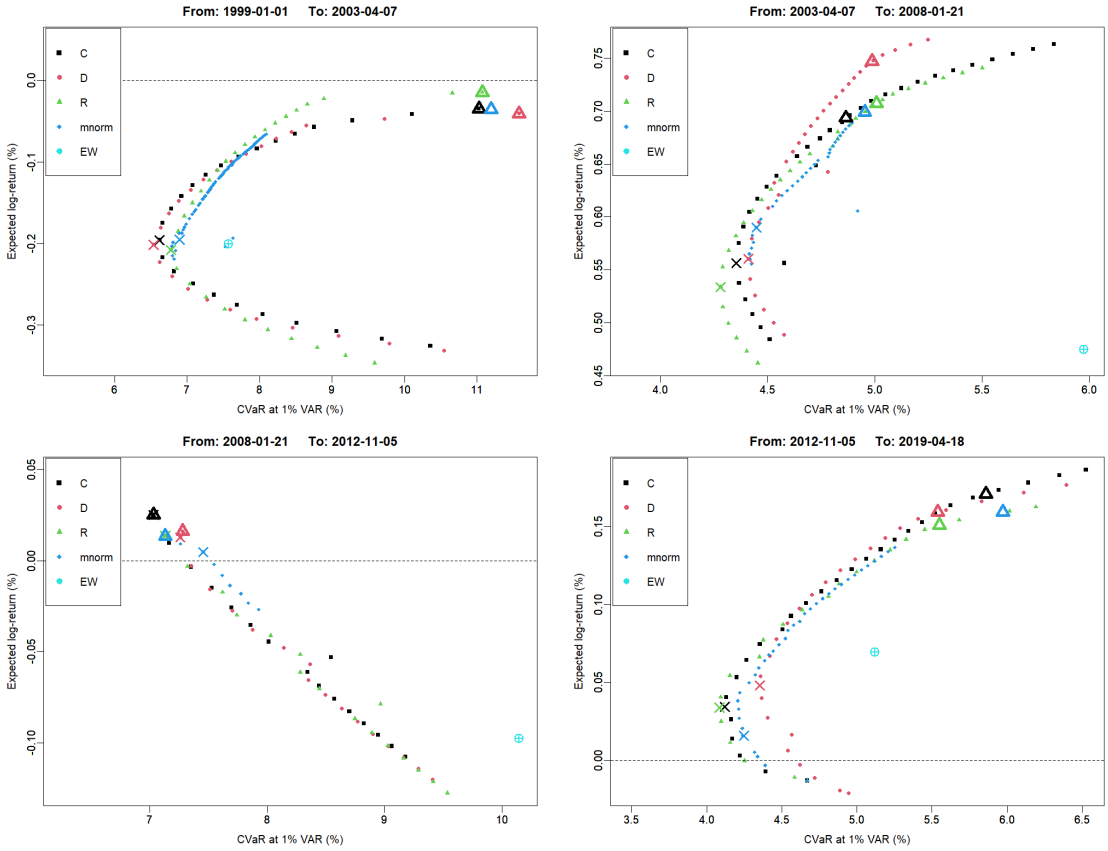


Figure 3.13: Efficiency frontiers. The crosses are indicating minimum CVaR and the triangles the maximum of return-to-CVaR. An equally weighted portfolio of the R-vine sims is indicated by the singular dot. (1999-2019)

Chapter 4

Summary and concluding notes

The goal of this thesis was to fit and compare the different vines. Especially the R-vine was suggested to be more flexible and robust.

4.1 Summary and conclusion

Fitting of an ARMA-EGARCH model with skewed Student's t-distributed residuals is found to be a god fit with most its components being significant for all the log return series inspected for the different time periods. The ARMA order is the part of the different models that most often is found to not have a significant effect, while the other effects are found to be significant. The changing volatility captured by the GARCH found that the best fits have orders around GARCH(1,1) and that all models have at least one component. The stronger reaction to negative market chocks depicted in the literature is also found to be true for this empirical study. Furthermore, the heavy tailed and skewed nature of the residuals is found to be significant as well when fitting, even though this was very unstable.

The three different pair copula constructions shows that they model the dependencies different from each other, but there is no clear pattern in the way they are different. However, if more precisely looking at the R-vine, it shows its flexibility having a structure close to a the-vine for some subperiods, and to C-vine for other periods. Comparing the use of vines with the multidimensional normal distribution for dependencies also does not give vastly different weights for the portfolio displayed in the plots of the efficiency frontiers. However, looking at the efficiency frontier on its own simulated data shows that it displays the risk as being a lot lower. This shows how the vines better capture the dependencies between log return series of stock market indexes. The different dependence structures were found to be very similar to each other, most likely due to how the simulations were done for one step ahead making them dependent on the current time step. This lead to a lot of influence from the state at these individual time steps on the optimal weights for the portfolio. Therefore, it is important to remember that some markets become overwhelmingly favourable and make the distribution of the portfolio very similar to a few of its markets. This leads to similar optimal weights of the different vines and the benchmark model.

4.2 Limitations and further work

The time constraint of this thesis unfortunately created many further tests and simulations to be neglected. The tuning of the parameters, as well as the fitting of

the model and evaluation of the time frames appeared to be very time consuming. To be able to assure a correct model, an implementation without mistakes was important and therefore traded with trying more tests and approaches. This lead to a big part of the work being building a framework for a robust simulation study. If given more time, in addition to what has been compared in this thesis, it would have been interesting as well to see how an ARMA model with normally distributed residuals would give different results.

One of the relatively simple alternatives that should be further researched would be experimenting with different GARCH variations, which might be able to better fit the time-varying variance. For the dependence structure, we recommend future research on this topic to focus on looking into incorporating time dependence, as in Nagler, Krüger and Min (2022) and Brenchmann and Czado (2015).

Appendix A

Appendix A - Additional results

Here are additional results.

A.1 ARMA-EGARCH orders

Country	CA	FR	GE	IT	JP	UK	US
ARMA(p,q)	(0,0)	(0,0)	(0,0)	(0,0)	(1,0)	(0,0)	(0,1)
EGARCH(p,q)	(1,1)	(1,2)	(1,1)	(1,0)	(1,2)	(1,1)	(2,1)

Table A.1: The ARMA and EGARCH orders for first period. for BIC.

Country	CA	FR	GE	IT	JP	UK	US
ARMA(p,q)	(0,0)	(0,0)	(0,1)	(0,1)	(0,0)	(0,1)	(0,0)
EGARCH(p,q)	(1,0)	(1,0)	(2,1)	(0,1)	(0,1)	(1,0)	(1,2)

Table A.2: The ARMA and EGARCH orders for second period. for BIC.

Country	CA	FR	GE	IT	JP	UK	US
ARMA(p,q)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)
EGARCH(p,q)	(1,1)	(1,1)	(1,1)	(1,1)	(0,1)	(1,1)	(1,1)

Table A.3: The ARMA and EGARCH orders for third period. for BIC.

Country	CA	FR	GE	IT	JP	UK	US
ARMA(p,q)	(0,0)	(0,0)	(1,0)	(0,0)	(0,0)	(0,0)	(1,1)
EGARCH(p,q)	(1,2)	(1,1)	(1,0)	(0,1)	(1,0)	(1,2)	(1,1)

Table A.4: The ARMA and EGARCH orders for forth period. for BIC.

A.2 Vines

A.2.1 Period 1: 1999-2003

A.2.2 Period 2: 2003-2008

A.2.3 Period 3: 2008-2012

A.2.4 Period 4: 2012-2019

Tree	Pair	Copula	Param. 1	Param. 2	Kendall's tau	Tail dep.
1	FR-IT	Gaussian	0.83 (0.02)	- -	0.62	- -
1	FR-JP	Gaussian	0.30 (0.06)	- -	0.19	- -
1	FR-CA	Gumbel	1.66 (0.09)	- -	0.40	0.48 -
1	FR-GE	t	0.85 (0.02)	12.80 (8.86)	0.65	0.31 0.31
1	FR-UK	t	0.78 (0.03)	6.22 (2.58)	0.57	0.38 0.38
1	US-FR	Gaussian	0.70 (0.03)	- -	0.49	- -
2	US-IT|FR	Rot. Joe 180 deg.	1.17 (0.07)	- -	0.09	- 0.19
2	US-JP|FR	Clayton	0.17 (0.09)	- -	0.08	- 0.02
2	US-CA|FR	t	0.51 (0.05)	9.44 (6.84)	0.34	0.09 0.09
2	US-GE|FR	Gumbel	1.16 (0.05)	- -	0.14	0.18 -
2	US-UK|FR	Gaussian	0.30 (0.06)	- -	0.19	- -
3	GE-IT|US,FR	Gaussian	0.41 (0.05)	- -	0.27	- -
3	GE-JP|US,FR	Indep.	- -	- -	0.00	- -
3	GE-CA|US,FR	Rot. Joe 180 deg.	1.12 (0.07)	- -	0.07	- 0.15
3	UK-GE|US,FR	Joe	1.12 (0.07)	- -	0.07	0.15 -
4	CA-IT|GE,US,FR	Rot. Tawn type 2 270 deg.	-8.67 (5.01)	0.01 (0.00)	−0.01	- -
4	CA-JP|GE,US,FR	Gaussian	0.20 (0.06)	- -	0.13	- -
4	UK-CA|GE,US,FR	t	0.05 (0.07)	7.60 (4.17)	0.03	0.02 0.02
5	UK-IT|CA,GE,US,FR	t	-0.03 (0.08)	5.44 (2.40)	−0.02	0.04 0.04
5	UK-JP|CA,GE,US,FR	Rot. Tawn type 2 270 deg.	20.00 (10.04)	0.02 (0.00)	−0.02	- -
6	JP-IT|UK,CA,GE,US,FR	Rot. Tawn type 1 270 deg.	-10.27 (6.78)	0.01 (0.00)	−0.01	- -

Figure A.1: C-vine (1999-2003)

Tree	Pair	Copula	Param. 1	Param. 2	Kendall's tau	Tail dep.
1	CA-JP	Gaussian	0.35 (0.05)	- -	0.23	- -
1	US-CA	Rot. BB1 180 deg.	0.57 (0.18)	1.54 (0.13)	0.49	0.45 0.43
1	GE-US	Gaussian	0.68 (0.03)	- -	0.48	- -
1	IT-GE	Gaussian	0.82 (0.02)	- -	0.62	- -
1	FR-IT	Gaussian	0.83 (0.02)	- -	0.62	- -
1	UK-FR	t	0.78 (0.03)	6.22 (2.58)	0.57	0.38 0.38
2	US-JP|CA	Rot. Joe 180 deg.	1.09 (0.06)	- -	0.05	- 0.11
2	GE-CA|US	Rot. Gumbel 180 deg.	1.17 (0.05)	- -	0.14	- 0.19
2	IT-US|GE	Clayton	0.19 (0.09)	- -	0.09	- 0.03
2	FR-GE|IT	t	0.52 (0.05)	8.36 (4.79)	0.35	0.12 0.12
2	UK-IT|FR	Rot. Tawn type 1 180 deg.	2.38 (1.33)	0.01 (0.01)	0.01	- 0.01
3	GE-JP|US,CA	Indep.	- -	- -	0.00	- -
3	IT-CA|GE,US	Indep.	- -	- -	0.00	- -
3	FR-US|IT,GE	Frank	1.74 (0.42)	- -	0.19	- -
3	UK-GE|FR,IT	Joe	1.18 (0.07)	- -	0.09	0.20 -
4	IT-JP|GE,US,CA	Rot. Tawn type 1 90 deg.	-19.6 5 (NaN)	0.0 0 (NaN)	−0.00	- -
4	FR-CA|IT,GE,US	Gumbel	1.08 (0.04)	- -	0.07	0.10 -
4	UK-US|FR,IT,GE	Clayton	0.29 (0.09)	- -	0.13	- 0.09
5	FR-JP|IT,GE,US,CA	Rot. Tawn type 2 180 deg.	1.37 (0.33)	0.07 (0.06)	0.04	- 0.05
5	UK-CA|FR,IT,GE,US	t	0.05 (0.08)	5.62 (2.51)	0.03	0.05 0.05
6	UK-JP|FR,IT,GE,US,CA	Rot. Tawn type 2 270 deg.	-15.65 (5.98)	0.02 (0.00)	−0.02	- -

Figure A.2: D-vine (1999-2003)

Tree	Pair	Copula	Param. 1	Param. 2	Kendall's tau	Tail dep.
1	GE-IT	Gaussian	0.82 (0.02)	- -	0.62	- -
1	FR-UK	t	0.78 (0.03)	6.22 (2.58)	0.57	0.38 0.38
1	FR-GE	t	0.85 (0.02)	12.80 (8.86)	0.65	0.31 0.31
1	CA-JP	Gaussian	0.35 (0.05)	- -	0.23	- -
1	US-CA	Rot. BB1 180 deg.	0.57 (0.18)	1.54 (0.13)	0.49	0.45 0.43
1	US-FR	Gaussian	0.70 (0.03)	- -	0.49	- -
2	FR-IT|GE	Rot. BB8 180 deg.	3.42 (1.85)	0.66 (0.25)	0.30	- -
2	US-UK|FR	Gaussian	0.30 (0.06)	- -	0.19	- -
2	US-GE|FR	Gumbel	1.16 (0.05)	- -	0.14	0.18 -
2	US-JP|CA	Rot. Joe 180 deg.	1.09 (0.06)	- -	0.05	- 0.11
2	FR-CA|US	Gumbel	1.14 (0.05)	- -	0.12	0.17 -
3	US-IT|FR,GE	Rot. Joe 180 deg.	1.08 (0.06)	- -	0.04	- 0.10
3	GE-UK|US,FR	Joe	1.12 (0.07)	- -	0.07	0.15 -
3	CA-GE|US,FR	Rot. Joe 180 deg.	1.13 (0.07)	- -	0.07	- 0.15
3	FR-JP|US,CA	Rot. Tawn type 2 180 deg.	1.33 (0.26)	0.13 (0.12)	0.06	- 0.08
4	CA-IT|US,FR,GE	Rot. Tawn type 2 270 deg.	-8.94 (3.29)	0.01 (0.00)	−0.01	- -
4	CA-UK|GE,US,FR	t	0.04 (0.07)	8.06 (4.57)	0.03	0.02 0.02
4	JP-GE|CA,US,FR	Indep.	- -	- -	0.00	- -
5	JP-IT|CA,US,FR,GE	Rot. Tawn type 2 180 deg.	8.36 (4.01)	0.01 (0.00)	0.01	- 0.01
5	JP-UK|CA,GE,US,FR	Rot. Tawn type 1 90 deg.	-10.95 (3.82)	0.02 (0.00)	−0.02	- -
6	UK-IT|JP,CA,US,FR,GE	t	-0.00 (0.08)	5.02 (2.10)	−0.00	0.05 0.05

Figure A.3: R-vine (1999-2003)

Tree	Pair	Copula	Param. 1	Param. 2	Kendall's tau	Tail dep.
1	FR-IT	Gaussian	0.90 (0.01)	- -	0.71	- -
1	FR-JP	Tawn type 2	1.92 (0.19)	0.56 (0.10)	0.32	0.40 -
1	FR-UK	Gaussian	0.86 (0.01)	- -	0.66	- -
1	FR-GE	Gaussian	0.90 (0.01)	- -	0.72	- -
1	FR-CA	Gaussian	0.68 (0.03)	- -	0.47	- -
1	US-FR	Gaussian	0.73 (0.02)	- -	0.52	- -
2	CA-IT|FR	Frank	0.97 (0.39)	- -	0.11	- -
2	CA-JP|FR	Frank	1.42 (0.39)	- -	0.15	- -
2	CA-UK|FR	Rot. Gumbel 180 deg.	1.28 (0.06)	- -	0.22	- 0.28
2	CA-GE|FR	Rot. Joe 180 deg.	1.05 (0.05)	- -	0.03	- 0.06
2	US-CA|FR	Gaussian	0.23 (0.06)	- -	0.14	- -
3	GE-IT|CA,FR	Tawn type 1	6.55 (4.15)	0.01 (0.00)	0.01	0.01 -
3	GE-JP|CA,FR	Frank	1.06 (0.40)	- -	0.12	- -
3	GE-UK|CA,FR	Indep.	- -	- -	0.00	- -
3	US-GE|CA,FR	Gaussian	0.24 (0.06)	- -	0.15	- -
4	UK-IT|GE,CA,FR	Frank	0.89 (0.39)	- -	0.10	- -
4	UK-JP|GE,CA,FR	Indep.	- -	- -	0.00	- -
4	US-UK|GE,CA,FR	Rot. Joe 270 deg.	-1.10 (0.06)	- -	−0.05	- -
5	JP-IT|UK,GE,CA,FR	Rot. Tawn type 2 90 deg.	-20.0 0 (NA)	0.00 (0.00)	−0.00	- -
5	US-JP|UK,GE,CA,FR	Indep.	- -	- -	0.00	- -
6	US-IT|JP,UK,GE,CA,FR	Indep.	- -	- -	0.00	- -

Figure A.4: C-vine (2003-2008)

Tree	Pair	Copula	Param. 1	Param. 2	Kendall's tau	Tail dep.
1	CA-JP	Rot. BB8 180 deg.	2.62 (0.67)	0.86 (0.11)	0.34	- -
1	US-CA	Gaussian	0.60 (0.03)	- -	0.41	- -
1	GE-US	Gaussian	0.72 (0.02)	- -	0.52	- -
1	IT-GE	Frank	9.04 (0.63)	- -	0.64	- -
1	FR-IT	Gaussian	0.90 (0.01)	- -	0.71	- -
1	UK-FR	Gaussian	0.86 (0.01)	- -	0.66	- -
2	US-JP|CA	Frank	1.37 (0.38)	- -	0.15	- -
2	GE-CA|US	Gaussian	0.31 (0.05)	- -	0.20	- -
2	IT-US|GE	Rot. Gumbel 180 deg.	1.12 (0.04)	- -	0.10	- 0.14
2	FR-GE|IT	Gaussian	0.67 (0.03)	- -	0.47	- -
2	UK-IT|FR	Frank	1.19 (0.40)	- -	0.13	- -
3	GE-JP|US,CA	Clayton	0.30 (0.09)	- -	0.13	- 0.10
3	IT-CA|GE,US	Frank	1.84 (0.39)	- -	0.20	- -
3	FR-US|IT,GE	Frank	0.93 (0.38)	- -	0.10	- -
3	UK-GE|FR,IT	Indep.	- -	- -	0.00	- -
4	IT-JP|GE,US,CA	Tawn type 2	20.0 0 (NaN)	0.0 0 (NaN)	0.00	0.00 -
4	FR-CA|IT,GE,US	Rot. Tawn type 2 180 deg.	1.79 (0.44)	0.10 (0.05)	0.08	- 0.09
4	UK-US|FR,IT,GE	Rot. Tawn type 1 180 deg.	1.37 (0.24)	0.11 (0.08)	0.06	- 0.08
5	FR-JP|IT,GE,US,CA	Indep.	- -	- -	0.00	- -
5	UK-CA|FR,IT,GE,US	Rot. Gumbel 180 deg.	1.24 (0.06)	- -	0.19	- 0.25
6	UK-JP|FR,IT,GE,US,CA	Rot. Tawn type 1 180 deg.	20.00 (14.38)	0.01 (0.00)	0.00	- 0.01

Figure A.5: D-vine (2003-2008)

Tree	Pair	Copula	Param. 1	Param. 2	Kendall's tau	Tail dep.
1	GE-JP	Frank	3.75 (0.43)	- -	0.37	- -
1	GE-US	Gaussian	0.72 (0.02)	- -	0.52	- -
1	FR-GE	Gaussian	0.90 (0.01)	- -	0.72	- -
1	FR-IT	Gaussian	0.90 (0.01)	- -	0.71	- -
1	UK-CA	Gaussian	0.71 (0.03)	- -	0.50	- -
1	UK-FR	Gaussian	0.86 (0.01)	- -	0.66	- -
2	US-JP|GE	Frank	0.96 (0.38)	- -	0.11	- -
2	FR-US|GE	Frank	1.55 (0.39)	- -	0.17	- -
2	IT-GE|FR	Tawn type 2	13.77 (11.12)	0.01 (0.00)	0.01	0.01 -
2	UK-IT|FR	Frank	1.19 (0.40)	- -	0.13	- -
2	FR-CA|UK	Gaussian	0.18 (0.06)	- -	0.12	- -
3	FR-JP|US,GE	Tawn type 2	3.32 (1.31)	0.03 (0.01)	0.03	0.03 -
3	IT-US|FR,GE	Rot. Joe 180 deg.	1.10 (0.05)	- -	0.05	- 0.12
3	UK-GE|IT,FR	Rot. Joe 180 deg.	1.08 (0.05)	- -	0.04	- 0.10
3	CA-IT|UK,FR	Rot. Joe 180 deg.	1.09 (0.06)	- -	0.05	- 0.11
4	IT-JP|FR,US,GE	Indep.	- -	- -	0.00	- -
4	UK-US|IT,FR,GE	Rot. Tawn type 1 180 deg.	1.40 (0.25)	0.13 (0.08)	0.07	- 0.09
4	CA-GE|UK,IT,FR	Indep.	- -	- -	0.00	- -
5	UK-JP|IT,FR,US,GE	Rot. Tawn type 1 180 deg.	9.70 (9.18)	0.01 (0.00)	0.01	- 0.01
5	CA-US|UK,IT,FR,GE	Gumbel	1.14 (0.05)	- -	0.13	0.17 -
6	CA-JP|UK,IT,FR,US,GE	Frank	1.29 (0.38)	- -	0.14	- -

Figure A.6: R-vine (2003-2008)

Tree	Pair	Copula	Param. 1	Param. 2	Kendall's tau	Tail dep.
1	FR-JP	BB7	1.47 (0.12)	1.02 (0.16)	0.43	0.40 0.51
1	FR-IT	t	0.94 (0.01)	12.67 (8.85)	0.78	0.53 0.53
1	FR-GE	Gaussian	0.96 (0.00)	- -	0.82	- -
1	FR-UK	Gaussian	0.90 (0.01)	- -	0.71	- -
1	FR-CA	Gaussian	0.79 (0.02)	- -	0.58	- -
1	US-FR	t	0.82 (0.02)	11.53 (8.32)	0.61	0.28 0.28
2	CA-JP|FR	Clayton	0.33 (0.09)	- -	0.14	- 0.12
2	CA-IT|FR	t	-0.00 (0.07)	5.03 (1.90)	−0.00	0.05 0.05
2	CA-GE|FR	Gaussian	0.15 (0.06)	- -	0.10	- -
2	CA-UK|FR	Gaussian	0.41 (0.05)	- -	0.27	- -
2	US-CA|FR	t	0.41 (0.06)	4.49 (1.50)	0.27	0.19 0.19
3	US-JP|CA,FR	t	-0.13 (0.07)	8.00 (4.06)	−0.08	0.01 0.01
3	US-IT|CA,FR	Indep.	- -	- -	0.00	- -
3	US-GE|CA,FR	t	0.16 (0.07)	4.12 (1.41)	0.10	0.11 0.11
3	US-UK|CA,FR	Gaussian	0.20 (0.06)	- -	0.13	- -
4	GE-JP|US,CA,FR	Frank	0.73 (0.38)	- -	0.08	- -
4	GE-IT|US,CA,FR	Indep.	- -	- -	0.00	- -
4	UK-GE|US,CA,FR	BB7	1.04 (0.04)	0.15 (0.09)	0.09	0.05 0.01
5	IT-JP|GE,US,CA,FR	t	0.10 (0.07)	8.15 (4.65)	0.06	0.02 0.02
5	UK-IT|GE,US,CA,FR	Indep.	- -	- -	0.00	- -
6	UK-JP|IT,GE,US,CA,FR	Indep.	- -	- -	0.00	- -

Figure A.7: C-vine (2008-2012)

Tree	Pair	Copula	Param. 1	Param. 2	Kendall's tau	Tail dep.
1	US-CA	BB1	0.66 (0.17)	1.79 (0.14)	0.58	0.53 0.56
1	UK-US	t	0.82 (0.02)	10.02 (7.09)	0.61	0.32 0.32
1	GE-UK	Gaussian	0.89 (0.01)	- -	0.69	- -
1	FR-GE	Gaussian	0.96 (0.00)	- -	0.82	- -
1	IT-FR	t	0.94 (0.01)	12.67 (8.85)	0.78	0.53 0.53
1	JP-IT	Rot. Gumbel 180 deg.	1.73 (0.09)	- -	0.42	- 0.51
2	UK-CA|US	Gaussian	0.48 (0.04)	- -	0.32	- -
2	GE-US|UK	Rot. Gumbel 180 deg.	1.27 (0.06)	- -	0.21	- 0.28
2	FR-UK|GE	Frank	2.32 (0.39)	- -	0.24	- -
2	IT-GE|FR	Indep.	- -	- -	0.00	- -
2	JP-FR|IT	t	0.17 (0.07)	4.18 (1.63)	0.11	0.11 0.11
3	GE-CA|UK,US	Gaussian	0.09 (0.06)	- -	0.06	- -
3	FR-US|GE,UK	t	0.08 (0.07)	4.66 (1.73)	0.05	0.07 0.07
3	IT-UK|FR,GE	t	-0.05 (0.07)	5.12 (1.96)	−0.03	0.04 0.04
3	JP-GE|IT,FR	Clayton	0.17 (0.08)	- -	0.08	- 0.02
4	FR-CA|GE,UK,US	Indep.	- -	- -	0.00	- -
4	IT-US|FR,GE,UK	Indep.	- -	- -	0.00	- -
4	JP-UK|IT,FR,GE	Rot. Tawn type 2 270 deg.	-3.59 (1.41)	0.03 (0.01)	−0.03	- -
5	IT-CA|FR,GE,UK,US	Indep.	- -	- -	0.00	- -
5	JP-US|IT,FR,GE,UK	Rot. Tawn type 1 270 deg.	-6.90 (3.53)	0.03 (0.00)	−0.03	- -
6	JP-CA|IT,FR,GE,UK,US	Frank	1.52 (0.38)	- -	0.16	- -

Figure A.8: D-vine (2008-20128)

Tree	Pair	Copula	Param. 1	Param. 2	Kendall's tau	Tail dep.
1	GE-JP	Rot. Gumbel 180 deg.	1.76 (0.09)	- -	0.43	- 0.52
1	FR-IT	t	0.94 (0.01)	12.67 (8.85)	0.78	0.53 0.53
1	GE-US	Rot. BB1 180 deg.	0.18 (0.13)	2.30 (0.17)	0.60	0.19 0.65
1	FR-GE	Gaussian	0.96 (0.00)	- -	0.82	- -
1	UK-CA	Gaussian	0.82 (0.02)	- -	0.61	- -
1	UK-FR	Gaussian	0.90 (0.01)	- -	0.71	- -
2	FR-JP|GE	Frank	0.63 (0.37)	- -	0.07	- -
2	GE-IT|FR	Indep.	- -	- -	0.00	- -
2	FR-US|GE	Gumbel	1.16 (0.05)	- -	0.14	0.19 -
2	UK-GE|FR	Rot. Gumbel 180 deg.	1.15 (0.05)	- -	0.13	- 0.18
2	FR-CA|UK	Frank	1.52 (0.40)	- -	0.17	- -
3	IT-JP|FR,GE	t	0.09 (0.07)	4.97 (2.06)	0.06	0.07 0.07
3	US-IT|GE,FR	t	-0.08 (0.07)	8.13 (4.45)	−0.05	0.01 0.01
3	UK-US|FR,GE	BB7	1.13 (0.07)	0.31 (0.10)	0.19	0.16 0.11
3	CA-GE|UK,FR	Rot. Gumbel 180 deg.	1.05 (0.04)	- -	0.05	- 0.07
4	US-JP|IT,FR,GE	Indep.	- -	- -	0.00	- -
4	UK-IT|US,GE,FR	Rot. Joe 90 deg.	-1.09 (0.06)	- -	−0.05	- -
4	CA-US|UK,FR,GE	Gumbel	1.24 (0.06)	- -	0.20	0.25 -
5	UK-JP|US,IT,FR,GE	Tawn type 1	20.00 (8.49)	0.02 (0.00)	0.02	0.02 -
5	CA-IT|UK,US,GE,FR	Tawn type 1	3.71 (1.71)	0.01 (0.00)	0.01	0.01 -
6	CA-JP|UK,US,IT,FR,GE	Rot. BB8 180 deg.	1.86 (0.70)	0.74 (0.24)	0.15	- -

Figure A.9: R-vine (2008-2012)

Tree	Pair	Copula	Param. 1	Param. 2	Kendall's tau	Tail dep.
1	FR-IT	t	0.84 (0.01)	9.20 (5.00)	0.63	0.37 0.37
1	FR-CA	t	0.63 (0.03)	5.88 (2.09)	0.43	0.25 0.25
1	FR-GE	t	0.91 (0.01)	16.61 (11.85)	0.72	0.36 0.36
1	FR-JP	Gaussian	0.52 (0.03)	- -	0.35	- -
1	FR-UK	Rot. BB1 180 deg.	0.26 (0.11)	2.19 (0.14)	0.60	0.30 0.63
1	US-FR	Gaussian	0.66 (0.03)	- -	0.46	- -
2	US-IT|FR	Indep.	- -	- -	0.00	- -
2	US-CA|FR	Clayton	0.47 (0.09)	- -	0.19	- 0.23
2	US-GE|FR	Frank	1.27 (0.34)	- -	0.14	- -
2	US-JP|FR	Gaussian	0.27 (0.05)	- -	0.17	- -
2	US-UK|FR	Gaussian	0.35 (0.05)	- -	0.22	- -
3	UK-IT|US,FR	Indep.	- -	- -	0.00	- -
3	UK-CA|US,FR	Gaussian	0.40 (0.04)	- -	0.26	- -
3	UK-GE|US,FR	Frank	0.79 (0.33)	- -	0.09	- -
3	UK-JP|US,FR	Tawn type 1	2.00 (0.51)	0.05 (0.02)	0.04	0.05 -
4	GE-IT|UK,US,FR	Indep.	- -	- -	0.00	- -
4	GE-CA|UK,US,FR	t	-0.06 (0.06)	6.33 (2.50)	−0.04	0.02 0.02
4	JP-GE|UK,US,FR	Gumbel	1.06 (0.04)	- -	0.06	0.08 -
5	CA-IT|GE,UK,US,FR	Indep.	- -	- -	0.00	- -
5	JP-CA|GE,UK,US,FR	Frank	0.53 (0.33)	- -	0.06	- -
6	JP-IT|CA,GE,UK,US,FR	Rot. Tawn type 1 270 deg.	-4.60 (2.38)	0.01 (0.00)	−0.01	- -

Figure A.10: C-vine (2012-2019)

Tree	Pair	Copula	Param. 1	Param. 2	Kendall's tau	Tail dep.
1	US-CA	BB1	0.66 (0.17)	1.79 (0.14)	0.58	0.53 0.56
1	UK-US	t	0.82 (0.02)	10.02 (7.09)	0.61	0.32 0.32
1	GE-UK	Gaussian	0.89 (0.01)	- -	0.69	- -
1	FR-GE	Gaussian	0.96 (0.00)	- -	0.82	- -
1	IT-FR	t	0.94 (0.01)	12.67 (8.85)	0.78	0.53 0.53
1	JP-IT	Rot. Gumbel 180 deg.	1.73 (0.09)	- -	0.42	- 0.51
2	UK-CA|US	Gaussian	0.48 (0.04)	- -	0.32	- -
2	GE-US|UK	Rot. Gumbel 180 deg.	1.27 (0.06)	- -	0.21	- 0.28
2	FR-UK|GE	Frank	2.32 (0.39)	- -	0.24	- -
2	IT-GE|FR	Indep.	- -	- -	0.00	- -
2	JP-FR|IT	t	0.17 (0.07)	4.18 (1.63)	0.11	0.11 0.11
3	GE-CA|UK,US	Gaussian	0.09 (0.06)	- -	0.06	- -
3	FR-US|GE,UK	t	0.08 (0.07)	4.66 (1.73)	0.05	0.07 0.07
3	IT-UK|FR,GE	t	-0.05 (0.07)	5.12 (1.96)	−0.03	0.04 0.04
3	JP-GE|IT,FR	Clayton	0.17 (0.08)	- -	0.08	- 0.02
4	FR-CA|GE,UK,US	Indep.	- -	- -	0.00	- -
4	IT-US|FR,GE,UK	Indep.	- -	- -	0.00	- -
4	JP-UK|IT,FR,GE	Rot. Tawn type 2 270 deg.	-3.59 (1.41)	0.03 (0.01)	−0.03	- -
5	IT-CA|FR,GE,UK,US	Indep.	- -	- -	0.00	- -
5	JP-US|IT,FR,GE,UK	Rot. Tawn type 1 270 deg.	-6.90 (3.53)	0.03 (0.00)	−0.03	- -
6	JP-CA|IT,FR,GE,UK,US	Frank	1.52 (0.38)	- -	0.16	- -

Figure A.11: D-vine (2012-2019)

Tree	Pair	Copula	Param. 1	Param. 2	Kendall's tau	Tail dep.
1	FR-IT	t	0.84 (0.01)	9.20 (5.00)	0.63	0.37 0.37
1	FR-GE	t	0.91 (0.01)	16.61 (11.85)	0.72	0.36 0.36
1	FR-JP	Gaussian	0.52 (0.03)	- -	0.35	- -
1	UK-CA	BB1	0.64 (0.14)	1.52 (0.10)	0.50	0.42 0.49
1	UK-FR	Rot. BB1 180 deg.	0.26 (0.11)	2.19 (0.14)	0.60	0.30 0.63
1	US-UK	Gaussian	0.69 (0.02)	- -	0.48	- -
2	GE-IT|FR	Indep.	- -	- -	0.00	- -
2	UK-GE|FR	Gumbel	1.12 (0.04)	- -	0.11	0.15 -
2	UK-JP|FR	Gaussian	0.17 (0.05)	- -	0.11	- -
2	US-CA|UK	Rot. Gumbel 180 deg.	1.14 (0.04)	- -	0.12	- 0.16
2	US-FR|UK	BB7	1.09 (0.05)	0.21 (0.08)	0.14	0.11 0.04
3	UK-IT|GE,FR	Indep.	- -	- -	0.00	- -
3	US-GE|UK,FR	Clayton	0.19 (0.07)	- -	0.09	- 0.02
3	US-JP|UK,FR	Rot. BB7 180 deg.	1.09 (0.06)	0.22 (0.08)	0.14	0.04 0.11
3	FR-CA|US,UK	Frank	0.57 (0.34)	- -	0.06	- -
4	US-IT|UK,GE,FR	Frank	-0.48 (0.34)	- -	−0.05	- -
4	JP-GE|US,UK,FR	Joe	1.09 (0.05)	- -	0.05	0.11 -
4	CA-JP|US,UK,FR	Frank	0.55 (0.34)	- -	0.06	- -
5	JP-IT|US,UK,GE,FR	Indep.	- -	- -	0.00	- -
5	CA-GE|JP,US,UK,FR	t	-0.09 (0.06)	6.24 (2.32)	−0.06	0.02 0.02
6	CA-IT|JP,US,UK,GE,FR	Indep.	- -	- -	0.00	- -

Figure A.12: R-vine (2012-2019)

Appendix A. Appendix A - Additional results

Appendix B

Appendix B - R code

Here is the R-code used for this thesis.

B.1 Parameter file

```
    # Parameters ----

## Time cut points ----
start_point <- "1999-01-01"
break1 <- "2003-04-07"
break2 <- "2008-01-21"
break3 <- "2012-11-05"
end_point <- "2019-04-18"

## Optimal ARMA-GARCH orders - sstd ----
arma_garch_orders_p1 <-
  list(dist = "sstd",
       ARMA = list(c(0,0), c(0,0), c(0,0), c(0,0), c(1,0), c(0,0), c(0,1)),
       GARCH = list(c(1,1), c(1,2), c(1,1), c(1,0), c(1,2), c(1,1), c(2,1)))
arma_garch_orders_p2 <-
  list(dist = "sstd",
       ARMA = list(c(0,0), c(0,0), c(0,1), c(0,1), c(0,0), c(0,1), c(0,0)),
       GARCH = list(c(1,0), c(1,0), c(2,1), c(0,1), c(0,1), c(1,0), c(1,2)))
arma_garch_orders_p3 <-
  list(dist = "sstd",
       ARMA = list(c(0,0), c(0,0), c(0,0), c(0,0), c(0,0), c(0,0), c(0,0)),
       GARCH = list(c(1,1), c(1,1), c(1,1), c(1,1), c(0,1), c(1,1), c(1,1)))
arma_garch_orders_p4 <-
  list(dist = "sstd",
       ARMA = list(c(0,0), c(0,0), c(1,0), c(0,0), c(0,0), c(0,0), c(1,1)),
       GARCH = list(c(1,2), c(1,1), c(1,0), c(0,1), c(1,0), c(1,2), c(1,1)))
arma_garch_orders_all<-
  list(dist = "sstd",
       ARMA = list(c(0,0), c(0,0), c(0,0), c(0,0), c(0,0), c(0,0), c(0,1)),
       GARCH = list(c(1,1), c(1,1), c(1,1), c(1,1), c(1,2), c(1,1), c(1,1)))

## D-vine roots ----
```

```
d_vine_root_p1 <- "5173426"
d_vine_root_p2 <- "5173426"
d_vine_root_p3 <- "1763245"
d_vine_root_p4 <- "5716324"
d_vine_root_all <- "5176324"

## Parameters per series ----
params_p1 <- list(desc = "p1",
                  start_point = start_point,
                  end_point = break1,
                  arma_garch_orders = arma_garch_orders_p1,
                  d_vine_root = d_vine_root_p1)

params_p2 <- list(desc = "p2",
                  start_point = break1,
                  end_point = break2,
                  arma_garch_orders = arma_garch_orders_p2,
                  d_vine_root = d_vine_root_p2)

params_p3 <- list(desc = "p3",
                  start_point = break2,
                  end_point = break3,
                  arma_garch_orders = arma_garch_orders_p3,
                  d_vine_root = d_vine_root_p3)

params_p4 <- list(desc = "p4",
                  start_point = break3,
                  end_point = end_point,
                  arma_garch_orders = arma_garch_orders_p4,
                  d_vine_root = d_vine_root_p4)

params_all <- list(desc = "all",
                   start_point = start_point,
                   end_point = end_point,
                   arma_garch_orders = arma_garch_orders_all,
                   d_vine_root = d_vine_root_all)

all_parameters <- list(params_p1, params_p2, params_p3, params_p4, params_all)
```

B.2 Manual fitting of models

The program used to fit the models, before pipeline functions were made to speed up the process

```
    # Load packages ----
source("functions.R")
source("parameters.R")

library(tseries)
```

```
library(FinTS)
library(e1071)
library(astsa)
library(fGarch)
library(sn)
library(quantmod)
library(parallel)
library(ecp)

#library(VC2copula)
library(combinat)
library(GJRM)

# Load data ----
load_data_folder <- "Data_1999-01_2019-16"
df_list <- load_files_from_folder(folder_path = load_data_folder)

titles <- names(df_list)

## Make ts ----
ts_list <- lapply(df_list, function(df) xts(x = df$Close, order.by = df$Date))

# Make lists of periods ----
series_p1 <- list(params = params_p1)
series_p2 <- list(params = params_p2)
series_p3 <- list(params = params_p3)
series_p4 <- list(params = params_p4)
series_all <- list(params = params_all)

series <- list(series_p1 = series_p1,
               series_p2 = series_p2,
               series_p3 = series_p3,
               series_p4 = series_p4,
               series_all = series_all)

series <- list(series_p1 = series_p1)

## Cut time series ----
for (i in seq_along(series)) {
  series[[i]]$ts_list <- cut_ts_list(
    ts_list = ts_list,
    sp = series[[i]]$params$start_point,
    ep = series[[i]]$params$end_point)
}


## Make log return ----
# Removes mean for inspection purposes.
for (i in seq_along(series)) {
```

```
  series[[i]]$log_returns <- make_log_returns_list(ts_list = series[[i]]$ts_list)
}


# Inspect data ----
print_stats(series[[1]]$log_returns$log_returns_list)
# mapply(function(df, title) {plot_transformed(df, title)}, series_all$log_returns$log
# invisible(mapply(function(df, title) {acf2(df, main = title, max.lag = 30)}, series


# Find optimal ARMA-GARCH order ----
# NB: TAKES A WHILE!
# arch_garch_tune_list(
#   ts_list = series[[1]]$log_returns$log_returns_list,
#   ar_max = 1,
#   ma_max = 1,
#   arch_max = 2,
#   garch_max = 2,
#   distribution_model = "sstd",
#   solver = "nlminb",
#   save_folder_path = "arma_garch_all_mean_FALSE_1725")



# Fit best ARMA GARCH orders ----
# Based on best BIC
for (i in seq_along(series)) {
  series[[i]]$uGARCHfit_list <- fit_all_arma_garch(
    data = series[[i]]$log_returns$log_returns_list,
    arma_orders = series[[i]]$params$arma_garch_orders$ARMA,
    garch_orders = series[[i]]$params$arma_garch_orders$GARCH,
    solver = "nlminb",
    dist_model = series[[i]]$params$arma_garch_orders$dist)
}


## Residuals - ARMA-GARCH ----
for (i in seq_along(series)){ # Residuals does not converge for p2 GE. But fit looks
  series[[i]]$res <- check_res(uGARCHfit_list = series[[i]]$uGARCHfit_list)
}

# t <- series[[2]]$res$residuals$GE
# sum(is.nan(t))

## Plots ----
plot_all_hist(df_list = series[[1]]$res$residuals, mfrow = c(2,4), breaks = 30, x_lim
# invisible(mapply(function(df, title) {acf2(df, main = title, max.lag = 30)}, res, t
```

```
# Transform residuals to uniform ----
for (i in seq_along(series)){
  series[[i]]$res$unif_res_list <- lapply(series[[i]]$uGARCHfit_list, make_unif)
}


plot_all_hist(series[[1]]$res$unif_res_list, mfrow = c(2,4))
# invisible(mapply(function(df, title) {acf2(df, main = title, max.lag = 60)}, unif_list, ti

# Vines ----
for (i in seq_along(series)){
  series[[i]]$res$unif_res_df <- data.frame(series[[i]]$res$unif_res_list)
}

## C-Vine fit ----
for (i in seq_along(series)){
  series[[i]]$c_vine <- RVineStructureSelect(
    data = series[[i]]$res$unif_res_df,
    familyset = NA,
    type = 1,
    se = TRUE)
}

# contour(c_vine)

## R-Vine fit ----
for (i in seq_along(series)){
  series[[i]]$r_vine <- RVineStructureSelect(
    data = series[[i]]$res$unif_res_df,
    familyset = NA,
    type = 0,
    se = TRUE)
}

# contour(r_vine)

## D-vine ----

### Find best root ----
for (i in seq_along(series)){
  series[[i]]$d_vine_specs$root <- d_vine_root(unif_res_df = series[[i]]$res$unif_res_df)
}

# Print best root
for (i in seq_along(series)){
  print(series[[i]]$d_vine_specs$root)
}
d_vine_root_p1 <- "5173426"
d_vine_root_p2 <- "5164237"
```

```
d_vine_root_p3 <- "1763245"
d_vine_root_p4 <- "5716324"
d_vine_root_all <- "5164237"


### Fit D vine fit ----
# Based on best d_vine_root
# Make R-vine matrix of D-vine
for (i in seq_along(series)){
  series[[i]]$d_vine_specs$r_vine_matrix <- make_r_vine_matrix(str_seq = series[[i]]$
}

# based on R-vine matrix
for (i in seq_along(series)){
  series[[i]]$d_vine <- RVineCopSelect(
    data = series[[i]]$res$unif_res_df,
    Matrix = series[[i]]$d_vine_specs$r_vine_matrix,
    se = TRUE)
}

# contour(d_vine)

# Check likelihood-ratio
# VuongClarke(obj1 = c_vine, obj2 = d_vine)

## Change points ----

# divs <- e.divisive(X = unif_df, R = 500)
# divs$estimates
# cp <- divs$estimates[2]


# Simulate ----
# Add number of observations to simulate
n_sims <- 10000
for (i in seq_along(series)){
  series[[i]]$params$n_sims <- n_sims
}

## Simulate from vine copula ----
# set.seed(123)
series[[1]]$c_vine_sims$raw_sim_matrix <- RVineSim(
  N = series[[1]]$params$n_sims,
  RVM = series[[1]]$c_vine)


## Transform to skewed t distribution ----
series[[1]]$c_vine_sims$skew_t_list <- skew_t_transform_list(
  fit_ugarch_list = series[[1]]$uGARCHfit_list,
```

```
  unif_matrix = series[[1]]$c_vine_sims$raw_sim_matrix)

# plot_all_hist(series[[1]]$c_vine_sims$skew_t_list, mfrow = c(2,4))

## Simulate ARMA-GARCH ----
# Gives Warnings, but looks good!
series[[1]]$c_uGARCHsim_list <- sim_ARMA_GARCH_list(
  uGARCHfit_list = series[[1]]$uGARCHfit_list,
  n_series = series[[1]]$params$n_sims,
  residuals_list = series[[1]]$c_vine_sims$skew_t_list)
```

B.3 Pipeline function

This function controlled the fitting of the ARMA-EGARCH models and the Vine models

```
# Load packages ----
source("functions.R")  # File with all functions
source("parameters.R") # File with all parameters

library(tseries)
library(FinTS)
library(e1071)
library(astsa)
library(fGarch)
library(sn)
library(quantmod)
library(parallel)
library(ecp)
# library(R.utils)
# #library(VC2copula)
library(combinat)
library(GJRM)

#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
# Load data ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
load_data_folder <- "Data_1999-01_2019-16"
df_list <- load_files_from_folder(folder_path = load_data_folder)


#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
## Pipeline functions ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

fit_garch_models <- function(df_list, params, save_obj_name = FALSE) {

  results <- list(params = params) # Results object
  titles <- names(df_list)
  ts_list <- lapply(df_list, # Make ts
                    function(df) xts(x = df$Close, order.by = df$Date))
```

```
  # Set time frame
  results$ts_list <- cut_ts_list(ts_list = ts_list,
                                 sp = params$start_point,
                                 ep = params$end_point)

  # log returns and mu values
  results$log_returns <- make_log_returns_list(ts_list = results$ts_list)

  # Fit optimal ARMA-GARCH models
  results$uGARCHfit_list <- fit_all_arma_garch(
    data = results$log_returns$log_returns_list,
    arma_orders = params$arma_garch_orders$ARMA,
    garch_orders = params$arma_garch_orders$GARCH,
    solver = "nlminb",
    dist_model = "sstd")

  # Standardized residuals and parameters fit on residuals to check for convergence
  results$res <- check_res(uGARCHfit_list = results$uGARCHfit_list)

  # Transform to uniform residuals and keep parameters

  unif_res_list <- lapply(results$uGARCHfit_list, make_unif)
  results$res$unif_res_df <- data.frame(unif_res_list)

  if (save_obj_name != FALSE) {
    class_name <- sprintf("fit_arma_garch_obj_%s_%s",
                          results$params$desc,
                          save_obj_name)
    class(results) <- class_name
    filepath <- paste0("R_obj/", class_name, ".rds")
    saveRDS(results, file = filepath)
    cat("Saved", filepath, "\n")
  } else {
    class(results) <- "fit_arma_garch"
  }
  return(results)
}


#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX


sim_returns <- function(obj, vine_type, save_obj_name = FALSE, n_sims = 10000, use_se
  # Takes fit_garch_models_obj and saves simulated object.
  #
  # Input:
  #   obj: fit_garch_models_obj
  #   vine_type: Character - "C", "D", "R"
```

```
#   save_obj_name: character - Name to add to file when saving
#   n_sims: Number of simulations
#   use_seed: What seed to use.
# Output: Object with all info

set.seed(use_seed)

# Results object
## Add parameters used for clarity later
results <- list(params = obj$params)
results$params$n_sims <- n_sims
results$params$simualted_using <- class(obj)

## Add data to results
results$vine_type <- vine_type
results$input_logreturns <- obj$log_returns
results$unif_res_df <- obj$res$unif_res_df


# Select and fit vine
if (vine_type == "D") {
  results$d_vine_specs$r_vine_matrix <- make_r_vine_matrix(str_seq = results$params$d_vine
  results$vine <- RVineCopSelect(
    data = results$unif_res_df,
    Matrix = results$d_vine_specs$r_vine_matrix,
    se = TRUE)
} else if (vine_type == "C") {
  results$vine <- RVineStructureSelect(
    data = results$unif_res_df,
    familyset = NA,
    type = 1,
    se = TRUE)
} else if (vine_type == "R") {
  results$vine <- RVineStructureSelect(
    data = results$unif_res_df,
    familyset = NA,
    type = 0,
    se = TRUE)
}


# Simulate from vine copula
results$vine_sims$raw_sim_matrix <- RVineSim(
  N = results$params$n_sims,
  RVM = results$vine)


# Transform simulations to skewed t distribution
results$vine_sims$skew_t_list <- skew_t_transform_list(
```

```
    fit_ugarch_list = obj$uGARCHfit_list,
    unif_matrix = results$vine_sims$raw_sim_matrix)


  # MAke log-returns matrix from uGARCHfit object with simulated residuals

  ## Make uGARCHfsim from uGARCHfit
  results$uGARCHsim_list <- sim_ARMA_GARCH_list(
    uGARCHfit_list = obj$uGARCHfit_list,
    n_series = results$params$n_sims,
    residuals_list = results$vine_sims$skew_t_list)

  ## Make log-returns matrix
  results$sim_logreturns_matrix <- sim_logreturns_matrix(
    uGARCHsim_list = results$uGARCHsim_list,
    series_mean_list = obj$log_returns$mean_log_return_list)

  if (save_obj_name != FALSE) {
    class_name <- sprintf("vine_sims_obj_%s_%s_%s_%s", results$params$desc, results$v
    class(results) <- class_name
    filepath <- paste0("R_obj/", class_name, ".rds")
    saveRDS(results, file = filepath)
    cat("Saved", filepath, "\n")
  } else {
    class(results) <- "vine_sims"
  }

  return(results)
}



#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX




sim_returns_mnormal <- function(obj, save_obj_name = FALSE, n_sims = 10000, use_seed =
  # Takes fit_garch_models_obj, then uses multinormal distribution to simulate more re
  #
  # Input:
  #   obj: fit_garch_models_obj
  #   save_obj_name: character - Name to add to file when saving
  #   n_sims: Number of simulations
  #   use_seed: What seed to use.
  # Output: Object with all info

  # Set initial values
  set.seed(use_seed)
```

```
  results <- list(params = obj$params) # Results object
  results$params$n_sims <- n_sims
  results$params$simualted_using <- class(obj)
  results$input_logreturns <- obj$log_returns


  # Simulate residuals from multinormal distribution
  results$sim_mn_matrix <- sim_multinormal_residuals_matrix(obj$uGARCHfit_list, n_series = n


  # Make uGARCHsim objects from uGARCHfit objects and simulated residuals
  sim_list <- vector(mode = "list", length = 7)
  names(sim_list) <- names(uGARCHfit_list)
  for (i in (1:7)) {
    sim_list[[i]] <- sim_ugarch(
      uGARCHfit_object = obj$uGARCHfit_list[[i]],
      n_series = n_sims,
      residuals = results$sim_mn_matrix[,i])
  }


  # Make log-returns matrix from uGARCHsim objects
  results$sim_logreturns_matrix <- sim_logreturns_matrix(
    uGARCHsim_list = sim_list,
    series_mean_list = obj$log_returns$mean_log_return_list)
  colnames(results$sim_logreturns_matrix) <- names(obj$uGARCHfit_list)

  if (save_obj_name != FALSE) {
    class_name <- sprintf("sim_returns_mnormal_%s_%s_%s", results$params$desc, n_sims, save_
    class(results) <- class_name
    filepath <- paste0("R_obj/", class_name, ".rds")
    saveRDS(results, file = filepath)
    cat("Saved", filepath, "\n")
  } else {
    class(results) <- "sim_returns_mnormal"
  }

  return(results)
}



#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
# Run of pipelines ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX


#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
# fit_garch_models ----
```

```
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
for (par in all_parameters) {
  temp <- fit_garch_models(df_list = df_list, params = par, save_obj_name = "04_12_18
}


#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
# sim_returns ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

# Read object
fit_garch_models_obj <- readRDS(file = "R_obj/fit_arma_garch_obj_all_04_12_1705.rds")
plot_all_hist(fit_garch_models_obj$log_returns$log_returns_list, mfrow = c(2,4))

vine_types <- list("D", "C", "R")
for (vine_type in vine_types) {
  sims <- sim_returns(obj = fit_garch_models_obj, vine_type = vine_type, save_obj_nam
}

plot_all_hist_matrix(sims$sim_returns_matrix, mfrow = c(2,4))
sims


#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
# sim_returns_mnormal ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

arma_garch_filenames <- list.files(path = "R_obj", pattern = "fit_arma_garch", full.na

for (arma_garch_file in arma_garch_filenames) {
  temp <- readRDS(file = arma_garch_file)
  temp_sim_returns_mnormal <- sim_returns_mnormal(obj = temp,
                                                  n_sims = 20000,
                                                  save_obj_name = "07_12_1540")
}
```

B.4 Efficiency frontier

The following code was used to find the efficiency frontier, and make the plots.

```
# library(VineCopula)
library(rugarch)
library(parallel)

#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
# Functions ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
## Optimisation algorithm ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

CVaR_func <- function(w, x, optimize = FALSE, pen_w = FALSE, pen_exp = FALSE, exp_ret = FALS
  # Calculates CVaR and other parameters. When optimization is give
  # Input:
  #   w: Vector - Weights
  #   x: Matrix - Log returns matrix
  #   optimize:
  #     "CVAR": Returns CVaR for optimization run
  #     "r2CVaR": Returns return-to-CVaR for optimization run
  #     FALSE: Runs calculation on given inputs
  #   pen_w: penalty factor for wheights deviating from summing to 1
  #   pen_exp: penalty factor for expected return deviating from "exp_ret"
  #   exp_ret: value for expected return when optimizing for a given return
  x <- x*100 # In percent
  penalty_w <- 0
  penalty_e <- 0
  w <- exp(w)
  if (pen_w) {
    penalty_w <- pen_w*(sum(w)-1)^2
  }
  w <- w/sum(w) # Make weights sum to 1
  y <- x %*% w  # Returns vector
  E_r <- mean(y)  # expected return
  if (exp_ret) {
    penalty_e <- pen_exp*(exp_ret-E_r)^2
  }
  VaR <- (-quantile(y, 0.01)[[1]]) # VaR is a positive value
  CVaR <- (-mean(y[y<(-VaR)]))  # CVaR is a positive value

  if (optimize == "CVaR") { # Only used when run by nlminb
    # print(penalty_e)
    return(CVaR + penalty_w + penalty_e)
  }

  r2CVaR <- E_r/CVaR      # Return to CVaR

  if (optimize == "r2CVaR") {# Only used when run by nlminb
    return(-r2CVaR + penalty_w)
  }

  # If no optimisation is used. Returns more parameters
  results <- list(w = w,
                  returns = y,
                  E_r = E_r,
                  VaR = VaR,
                  CVaR = CVaR,
```

```
                  r2CVaR = r2CVaR,
                  penalty_weigth = penalty_w,
                  penalty_exp_ret = penalty_e)
  class(results) <- "CVaR function"
  return(results)
}


#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
## nlminb ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

nlminb_func <- function(start_par,
                        returns_matrix,
                        optim_val = "CVaR",
                        pen_w = FALSE,
                        pen_exp = FALSE,
                        exp_ret = 0,
                        max_eval = 500,
                        max_iter = 500,
                        trace_n = 100) {
  # Optimises and returns nlminb object
  # Also returns a run with the parameters found during optimisation
  # returns_matrix <- returns_matrix*100 # In percent

  nlminb_run <- nlminb(start = start_par,
         objective = CVaR_func,
         x = returns_matrix,
         optimize = optim_val,
         pen_w = pen_w,
         pen_exp = pen_exp,
         exp_ret = exp_ret,
         control = list(eval.max = max_eval,
                        iter.max = max_iter,
                        trace = trace_n),
         lower = -Inf,
         upper = Inf)

  optimal_run <- CVaR_func(w = nlminb_run$par,
                       x = returns_matrix,
                       optimize = FALSE,
                       pen_w = pen_w,
                       pen_exp = pen_exp,
                       exp_ret = exp_ret)

  results <- list(nlminb_run = nlminb_run, optimal_run = optimal_run)
  class(results) <- "nlminb_func"
  return(results)
}
```

```
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
## Find efficiency frontier ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
# Find efficiency frontier for one series

efficiency_frontier <- function(exp_center = "USE_MIN_CVAR",
                                exp_lower,
                                exp_upper_extra = 0,
                                step_size,
                                returns_matrix,
                                pen_w = 1e3,
                                pen_exp = 1e7,
                                max_eval = 10000,
                                max_iter = 10000,
                                trace_n = 100,
                                scale_step = 1,
                                transf_mult = FALSE,
                                transf_add = FALSE,
                                transf_pow = FALSE) {
  #   If "FIND_BEST", first calculates expected value for minimum CVaR
  #   exp_lower: Window to check around expectation center

  # Calculate best CVaR values
  nlminb_run <- nlminb_func(start_par = log(rep(1/7,7))+rnorm(7),
                  returns_matrix = returns_matrix,
                  optim_val = "CVaR",
                  pen_w = pen_w,
                  pen_exp = FALSE,
                  exp_ret = FALSE,
                  max_eval = max_eval,
                  max_iter = max_iter,
                  trace_n = trace_n)
  CVaR_min <- list()
  CVaR_min$CVaR_val <- nlminb_run$optimal_run$CVaR
  CVaR_min$E_r <- nlminb_run$optimal_run$E_r
  CVaR_min$weights <- nlminb_run$optimal_run$w

  # Calculate best r2CVaR values
  nlminb_run <- nlminb_func(start_par = log(rep(1/7,7))+rnorm(7),
                            returns_matrix = returns_matrix,
                            optim_val = "r2CVaR",
                            pen_w = pen_w,
                            pen_exp = FALSE,
                            exp_ret = FALSE,
                            max_eval = max_eval,
                            max_iter = max_iter,
                            trace_n = trace_n)
```

```
r2CVaR_max <- list()
r2CVaR_max$CVaR_val <- nlminb_run$optimal_run$CVaR
r2CVaR_max$E_r <- nlminb_run$optimal_run$E_r
r2CVaR_max$weights <- nlminb_run$optimal_run$w


# Sets exp_center to the minimum for CVaR if selected
if (exp_center == "USE_MIN_CVAR") {
  exp_center <- CVaR_min$E_r
}

# Expectations to search over
exp_seq <- seq(from = exp_center-exp_lower,
               to = (r2CVaR_max$E_r + step_size*scale_step*2 + exp_upper_extra),
               by = step_size*scale_step)
# Transforms sequence. Fewer points in middle where flat. Has optimal value anyways
if (transf_mult) {
  exp_seq <- (exp_seq-exp_center)/(transf_add+transf_mult*abs(exp_seq - exp_center))
}
l <- length(exp_seq)
exp_ret_list <- rep(NA,l)
CVaR_list <- rep(NA,l)
tot_pen_list <- rep(NA,l)
weights_list <- vector("list", length = l)
for (i in (1:l)) {
  t <- nlminb_func(start_par = log(rep(1/7,7))+rnorm(7),
                   returns_matrix = returns_matrix,
                   optim_val = "CVaR",
                   pen_w = pen_w,
                   pen_exp = pen_exp,
                   exp_ret = exp_seq[i],
                   max_eval = max_eval,
                   max_iter = max_iter,
                   trace_n = trace_n)
  exp_ret_list[i] <- t$optimal_run$E_r
  CVaR_list[i] <- t$optimal_run$CVaR
  tot_pen_list[i] <- t$optimal_run$penalty_w + t$optimal_run$penalty_e
  weights_list[[i]] <- t$optimal_run$w
  if(tot_pen_list[i] > 1e-3) {
    cat("\nWARNING!!! Total penalty: ", tot_pen_list[i], ", for expected return: ",
  }
}
params_used <- list(exp_center = exp_center,
                    exp_lower = exp_lower,
                    step_size = step_size,
                    pen_w = pen_w,
                    pen_exp = pen_exp)
results <- list(params_used = params_used,
                exp_search_seq = exp_seq,
```

```
                  exp_ret_list = exp_ret_list,
                  CVaR_list = CVaR_list,
                  CVaR_min = CVaR_min,
                  r2CVaR_max = r2CVaR_max,
                  tot_pen_list = tot_pen_list,
                  weights_list = weights_list)
  return(results)
}

#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
## Find efficiency frontier from weights ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
# Find efficiency frontier for one series

efficiency_frontier_from_weights <- function(obj, returns_matrix){
  # Takes inn efficiency frontier fitted on mean-variance and returns matrix
  # Then caluclates everything on the matrix instead
  l <- length(obj$weights_list)
  # browser()

  # Calculate new lists
  obj$exp_ret_list <- rep(NA,l)
  obj$CVaR_list <- rep(NA,l)
  for (i in 1:l){
    optimal_run <- CVaR_func(w = log(obj$weights_list[[i]]),
                             x = returns_matrix,
                             optimize = FALSE,
                             pen_w = FALSE,
                             pen_exp = FALSE,
                             exp_ret = FALSE)
    obj$exp_ret_list[i] <- optimal_run$E_r
    obj$CVaR_list[i] <- optimal_run$CVaR
  }

  # Calculate Min CVaR
  optimal_run <- CVaR_func(w = log(obj$CVaR_min$weights),
                           x = returns_matrix,
                           optimize = FALSE,
                           pen_w = FALSE,
                           pen_exp = FALSE,
                           exp_ret = FALSE)
  obj$CVaR_min$E_r <- optimal_run$E_r
  obj$CVaR_min$CVaR_val <- optimal_run$CVaR


  # Calculate max r2CVaR
  optimal_run <- CVaR_func(w = log(obj$r2CVaR_max$weights),
                           x = returns_matrix,
                           optimize = FALSE,
```

```
                            pen_w = FALSE,
                            pen_exp = FALSE,
                            exp_ret = FALSE)
  obj$r2CVaR_max$E_r <- optimal_run$E_r
  obj$r2CVaR_max$CVaR_val <- optimal_run$CVaR

  # Equal weights
  optimal_run <- CVaR_func(w = rep(1/7,7),
                            x = returns_matrix,
                            optimize = FALSE,
                            pen_w = FALSE,
                            pen_exp = FALSE,
                            exp_ret = FALSE)
  obj$equal_weights <- list()
  obj$equal_weights$E_r <- optimal_run$E_r
  obj$equal_weights$CVaR_val <- optimal_run$CVaR

  return(obj)
}


#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
## Calculate frontiers ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
# Find efficiency frontier for alle series in one time frame

frontiers_for_time_frame <- function(
    time_obj,
    exp_center = "USE_MIN_CVAR",
    exp_lower = 0.3,
    step_size = 0.02,
    pen_w = 1e3,
    pen_exp = 1e7,
    max_eval = 10000,
    max_iter = 10000,
    trace_n = 200,
    scale_step = c(1,1,1,1),
    transf_add,
    transf_mult,
    transf_pow,
    scale_windom = c(1,1,1,1)
) {
  # Input
  # time_obj:
  # exp_center = "USE_MIN_CVAR",
  # exp_lower
  # step_size
  # pen_w
  # pen_exp
```

```
  # max_eval
  # max_iter
  # trace_n
  # scale_step = c(1,1,1,1),
  # transf_add,
  # transf_mult,
  # transf_pow
  # scale_windom: Vector - Scales the windows to search for different series
  # Takes object and calculates frontier
  frontiers <- list()
  # browser()
  for(i in 1:4) {
    series_type <- names(time_obj)[i]
    print(series_type)
    frontiers[[series_type]] <- efficiency_frontier(
      exp_center = exp_center,
      exp_lower = exp_lower * scale_windom[i],
      step_size = step_size * scale_windom[i],
      returns_matrix = time_obj[[series_type]]$sim_logreturns_matrix,
      scale_step = scale_step[i],
      pen_w = pen_w,
      pen_exp = pen_exp,
      max_eval = max_eval,
      max_iter = max_iter,
      trace_n = trace_n,
      transf_add = transf_add,
      transf_mult = transf_mult,
      transf_pow = transf_pow)
  }
  return(frontiers)
}


#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
## Plot efficiency frontier ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

plot_eff_front <- function(time_obj, not_last = 0) {
  par(mar = c(3.5, 3.5, 2.5, .5), mgp = c(2,0.5,0), oma = c(0,0,0,0))

  frontiers <- time_obj$frontiers[-length(time_obj$frontiers)-not_last]
  n_obj <- length(frontiers)
  CVaR_ew_list <- rep(NA, n_obj)
  E_r_ew_list <- rep(NA, n_obj)
  for (i in 1:n_obj) {
    CVaR_ew_list[i] <- time_obj[[i]]$equal_weights$CVaR
    E_r_ew_list[i] <- time_obj[[i]]$equal_weights$E_r
  }
  main <- paste("From:", time_obj[[1]]$params$start_point,
```

```
                "    To:", time_obj[[1]]$params$end_point)
  x_min <- (min(unlist(lapply(frontiers, function(x) x$CVaR_list)))-1)
  x_max <- max(unlist(lapply(frontiers, function(x) x$CVaR_list)), max(CVaR_ew_list))
  y_min <- min(range(lapply(frontiers, function(x) x$exp_ret_list)), min(E_r_ew_list))
  y_max <- max(range(lapply(frontiers, function(x) x$exp_ret_list)),
               range(lapply(frontiers, function(x) x$r2CVaR$E_r)),
               max(E_r_ew_list))
  # browser()
  plot(NULL,
       xlim = c(x_min - (1/4)*(x_max-x_min), x_max),
       ylim = c(y_min, y_max),
       main = main,
       xlab = "CVaR at 1% VAR (%)",
       ylab = "Expected log-return (%)",
       type = "n")
  axis(side = 1)
  axis(side = 2)
  # frontiers
  for (i in (1:n_obj)) {
    lines(frontiers[[i]]$CVaR_list, frontiers[[i]]$exp_ret_list, col = i, pch = 14+i,
  }

  # min CVaR and max r2CVaR
  for (i in (1:n_obj)) {
    points(frontiers[[i]]$CVaR_min$CVaR_val, frontiers[[i]]$CVaR_min$E_r, col = i, pch
    points(frontiers[[i]]$r2CVaR_max$CVaR_val, frontiers[[i]]$r2CVaR_max$E_r, col = i
    # points(time_obj[[i]]$equal_weights$CVaR, time_obj[[i]]$equal_weights$E_r, col =
  }

  # Equal weights
  points(frontiers$mnorm$equal_weights$CVaR, frontiers[[i]]$equal_weights$E_r, col = 5
  # browser()
  abline(h = 0, lty = 2)
  legend(x = x_min - (x_max-x_min)*0.295,
         y = y_max + (y_max-y_min)*0.035,
         legend = c(names(frontiers), "EW"),
         col = (1:(n_obj+1)),
         lwd = 3,
         lty = 0,
         pch = c(15:(14+n_obj), 10))
  a <- "temp"
}

#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
## Equal weigthts calculations ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

CVaR_E_R_equal_weights <- function(logreg_matrix) {
  # logreg_matrix
```

```
  logreg_matrix <- logreg_matrix * 100
  y <- apply(logreg_matrix, 1, sum)/7
  E_r <- mean(y)
  VaR <- (-quantile(y, 0.01)[[1]]) # VaR is a positive value
  CVaR <- (-mean(y[y<(-VaR)]))  # CVaR is a positive value
  results <- list(E_r = E_r, VaR = VaR, CVaR = CVaR)
  return(results)
}


#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
## Efficiency frontier object ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
# Generate efficiency frontier object to store all data in one place

# Inititate object to keep track
eff_front <- list()



# For vines
vine_sim_filenames <- list.files(path = "R_obj", pattern = "vine_sims", full.names = TRUE)
vine_sim_filenames <- vine_sim_filenames[grep("07_12_1145", vine_sim_filenames)]

for (file in vine_sim_filenames) {
  t <- readRDS(file = file)
  time_period <- paste(t$params$desc)
  type <- paste(t$vine_type)
  eff_front[[time_period]][[type]] <- list()
  eff_front[[time_period]][[type]]$params <- t$params
  eff_front[[time_period]][[type]]$sim_logreturns_matrix <- t$sim_logreturns_matrix
  eff_front[[time_period]][[type]]$equal_weights <- CVaR_E_R_equal_weights(
    logreg_matrix = t$sim_logreturns_matrix)
}


# For multinormal on ARMA GARCH models
sim_ret_mnorm_filenames <- list.files(path = "R_obj", pattern = "mnorm", full.names = TRUE)

for (file in sim_ret_mnorm_filenames) {
  t <- readRDS(file = file)
  time_period <- paste(t$params$desc)
  multinormal_name <- paste("mnorm")
  eff_front[[time_period]][[multinormal_name]] <- list()
  eff_front[[time_period]][[multinormal_name]]$params <- t$params
  eff_front[[time_period]][[multinormal_name]]$sim_logreturns_matrix <- t$sim_logreturns_mat
  eff_front[[time_period]][[multinormal_name]]$equal_weights <- CVaR_E_R_equal_weights(
    logreg_matrix = t$sim_logreturns_matrix)
}
```

```
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
# RUN ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
## Load filenames ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX


eff_front <- readRDS("R_obj/eff_frontier_13_12_1240_fixed_end_date.rds")
saveRDS(eff_front, file = "R_obj/eff_frontier_06_12_1040.rds")




#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
## Run frontier calculations ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX


#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
### For a whole time frame ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
# NB! Overwrites exsisting

# eff_front$p1$frontiers <- list()
eff_front$all$frontiers <- frontiers_for_time_frame(time_obj = eff_front$all,
                                                    exp_center = "USE_MIN_CVAR",
                                                    exp_lower = 0.07,
                                                    step_size = 0.003,
                                                   transf_add = .5,
                                                   transf_mult = 2,
                                                   transf_pow = .5,
                                                   scale_windom = c(1,1,1.1,1.5),
                                                   trace_n = 0,
                                                   scale_step = c(1,1,1,1.5))


#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
### For a singe series in a time series ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
# NB! Overwrites existing

eff_front$all$frontiers$mnorm_weights_list <- efficiency_frontier(exp_center = "USE_M
                         exp_lower = 0.03,
                         exp_upper_extra = 0.01,
```

```
                        step_size = .01,
                        returns_matrix = eff_front$all$mnorm$sim_logreturns_matrix,
                        pen_w = 1e3,
                        pen_exp = 1e7,
                        scale_step = 1,
                        max_eval = 10000,
                        max_iter = 10000,
                        trace_n = 0,
                        transf_add = .5,
                        transf_mult = 2,
                        transf_pow = .5)


#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
### For a singe series in a time frame ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
# Overwrite one bad simulation

t <- nlminb_func(start_par = log(rep(1/7,7))+rnorm(7),
            returns_matrix = eff_front$p1$C$sim_logreturns_matrix,
            optim_val = "r2CVaR",
            pen_w = 1e3,
            pen_exp = FALSE,
            exp_ret = FALSE,
            max_eval = 10000,
            max_iter = 10000,
            trace_n = 100)

eff_front$p1$frontiers$C$r2CVaR_max$CVaR_val <- t$optimal_run$CVaR
eff_front$p1$frontiers$C$r2CVaR_max$E_r <- t$optimal_run$E_r

#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
### Transform mean var weights to frontier ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

eff_front$all$frontiers$mnorm <- efficiency_frontier_from_weights(obj = eff_front$all$fronti

eff_front$all$frontiers$mnorm_weights_list <- eff_temp$all$frontiers$mnorm
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
### Plot frontier ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

plot_eff_front(time_obj = eff_front$all, not_last = 0)
eff_temp <- eff_front
eff_temp$all$frontiers$mnorm_weights_list <- NULL

#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
### Save frontier object ----
#XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
saveRDS(eff_front, file = "R_obj/eff_frontier_08_12_1700.rds")
```

Bibliography

(2023). URL: https://www.uio.no/studier/emner/matnat/math/STK4520/h10/undervisningsmateriale/copula.pdf (visited on 11/12/2023).

Aas, K. et al. (2009). 'Pair-copula constructions of multiple dependence'. In: *Insurance: Mathematics and Economics* 44, pp. 182–198.

Bedford, T. and R. M. Cooke (2001). 'Probability density decomposition for conditionally dependent random variables modeled by vines'. In: *Annals of Mathematics and Artificial Intelligence* 32, pp. 245–268.

— (2002). 'Vines–a new graphical model for dependent random variables'. In: *The Annals of Statistics* 30, pp. 1031–1068. DOI: 10.1214/aos/1031689016.

Brenchmann, E. and C. Czado (2015). 'Copar – multivariate time series modeling using the copula autoregressive model.' In: *Applioed Stochastial Models In Business and Industry*, pp. 495–514.

Brockwell, Peter J. and Richard A. Davis (2016). *Introduction to Time Series and Forecasting (3rd ed.)* Springer. ISBN: 9783319298528.

Dissmann, J. et al. (2013). 'Selecting and estimating regular vine copulae and application to financial returns'. In: *Computational Statistics & Data Analysis*.

Embrechts, Paul, Filip Lindskog and Alexander McNeil (2001). 'Modelling Dependence with Copulas and Applications to Risk Management'. In: -.

Fernandez, C. and M.F. Steel (1998). 'On bayesian modeling of fat tails and skewness'. In: *Journal of the American Statistical Association* 93, pp. 359–371.

Fréchet, M (1957). 'Les tableaux de corrélation dont les marges et des bornes sont données,' in: *Annales de l'Université de Lyon, Sciences Mathématiques et Astronomie, 20*.

Ghalanos, Alexios (19th Sept. 2023). *Introduction to the rugarch package*. Version 1.4-3.

Joe, H. (1997). *Multivariate Models and Dependence Concepts*. Chapman & Hall.

Krouthén, Johannes (2015). 'Extreme joint dependencies with copulas A new approach for the structure of C-Vines'. Uppsala University.

Kurowicka, D. and R.M. Cooke (2005). 'Distribution — free continuous bayesian belief nets'. In: *Series on Quality, Reliability and Engineering StatisticsModern Statistical and Mathematical Methods in Reliability*, pp. 309–322. DOI: 10.1142/9789812703378_0022.

Markowitz, Harry (1952). 'Portfolio Selection'. In: *Journal of Finance* 7, pp. 77–91.

Nagler, T., D. Krüger and A. Min (2022). 'Stationary vine copula models for multivariate time series'. In: *Journal of Econometrics*, pp. 305–324.

Nelson, Roger B. (2006). *An introduction to copulas (2nd. ed.)* Springer.

Nguyen, P. M. and W. Liu (2023). 'Portfolio management using time-varying vine copula: an application on the G7 equity market indices'. In: *The European Journal of Finance* 29.11. DOI: 10.1080/1351847X.2022.2124119.

Shumway, Robert H. and David S. Stoffer (2017). *Time Series Analysis and Its Applications (4th ed.)* Springer. ISBN: 9783319524511.

Sklar, A (1959). 'Fonctions de répartition à n dimensions e leurs marges'. In: *Publications de l'Institut de Statistique de l'Univiversité de Paris 8*.

Webby, Roger Brian (2009). 'APPLICATIONS OF CONDITIONAL VALUE-AT-RISK TO WATER RESOURCES MANAGEMENT'. The Universtiy of Adelaide.

heading=bibintoc