Master Thesis

Report

Yurong Li

May 28, 2017

yurong@student.chalmers.se

Contents

Abstract

1 Introduction

In the stock market, the portfolio management is one of the most heated and significant problem. It is not difficult to image that the model selection will play a decisive role in the portfolio and the corresponding risk, and the risk of the portfolio is strongly associated to the dependence of the stocks.

It is well-known that the correlation coeffcient $\rho = \frac{cov(X,Y)}{\sqrt{varX} \cdot \sqrt{varY}}$ is an effective approch to describe the corelation between random variable X and Y. However, the correlation coefficient has several drawbacks, to begin with, it can only measure the linear dependence. In addition, the correlation coefficient can only depict the dependence of random variables X and Y if their joint distribution is elliptical. If we are not sure the denpendence structure of X and Y or they are not satisfied the condition above, we should choose another way to describe the dependence.

In the thesis, we will introduce the definitions and theories of copula, which is a multivariate probability distribution with the uniformly distributed margins. And we will establish the preliminary copula model of four stocks in Swedish market.

2 Theory

In statistics, copula is a multvariate probability distribution with the uniformly distributed margins, it is usually used to describe the dependence between random variables. In this section, we will introduce the basic definitions and theories of copula. We should begin from the two dimensional case.

2.1 Bivariate Copula

To begin with, we need to know the definitions of groundness and 2-incresing.

Definition 2.1.1 Let us consider two non-empty subsets A_1 and A_2 of R and a function G: $A_1 \times A_2 \to R$. Denote with a_i the least element of A_i, $i = 1, 2$. The function G is **grounded** if for every (u, v) of $A_1 \times A_2$,

$$G(a_1, v) = G(u, a_2) = 0$$

Definition 2.1.2 G: $A_1 \times A_2 \to R$ is called **2-incresing** if for every retangle $[u_1, u_2] \times [v_1, v_2]$ whose vertices lie in $A_1 \times A_2$, such that $u_1 \leq u_2$, $v_1 \leq v_2$,

$$G(u_2, v_2) - G(u_2, v_1) - G(u_1, v_2) + G(u_1, v_1) \geq 0$$

The above notions allow us to define copula in two dimensions.

Definition 2.1.3 A two-dimensional copula C is a real function of defined on $A \times B$, where A and B are non-empty subsets of $I = [0, 1]$,

$$C : A \times B \to R$$

(i) grounded. i.e. $C(u, 0) = C(0, v) = 0$.

(ii) 2-incresing.

(iii) such that

$$C(u, 1) = u, \quad C(1, v) = v$$

for every (u, v) of $A \times B$.

2.2 Copula Bounds

From the definition of copula, we can observe that the copula function lies in the unit cube $\mathbf{I}^3$. The Frechet-Hoeffding Theorem states the upper and lower bounds of the copula.

Theoren 2.2.1 (Frechet-Hoeffding Theorem) For $\forall(u, v) \in I^2$, the copula function C satisfied,

$$W(u, v) = \max\{u + v - 1, 0\} \leq C(u, v) \leq \min\{u, v\} = M(u, v)$$

This inequality is know as the *Frechet-Hoeffding bounds inequality*, and the function W and M are known as the Frechet-Hoeffding lower and upper bounds respectively.

By using mathematical software, we can graphically observe the two dimensional Frechet-Hoeffding bounds:

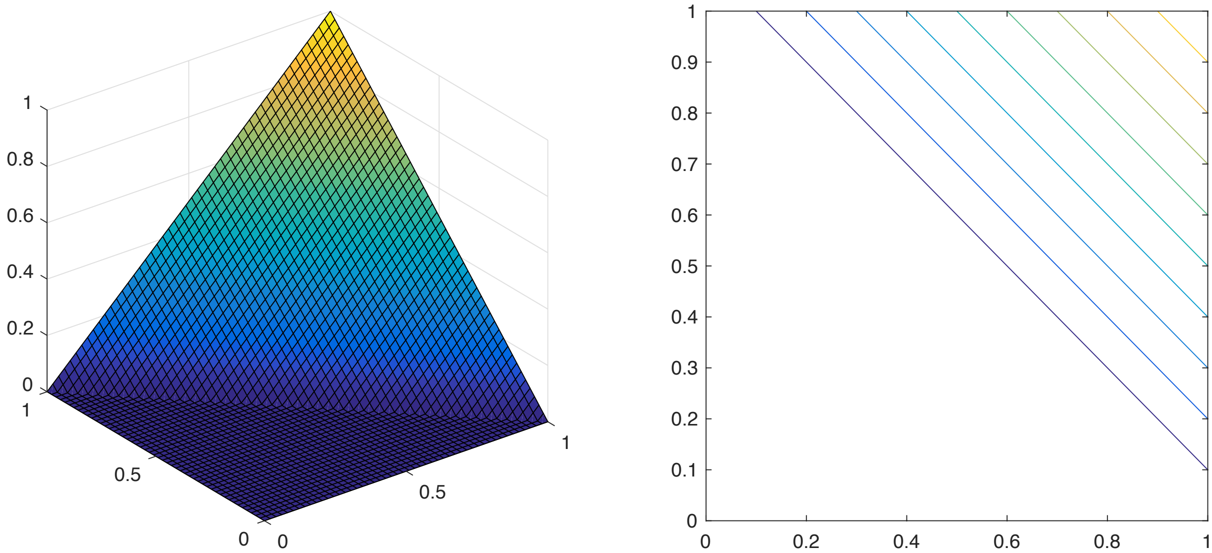


Figure 1: Lower Frechet-Hoeffding bound $W(u_1, u_2)$

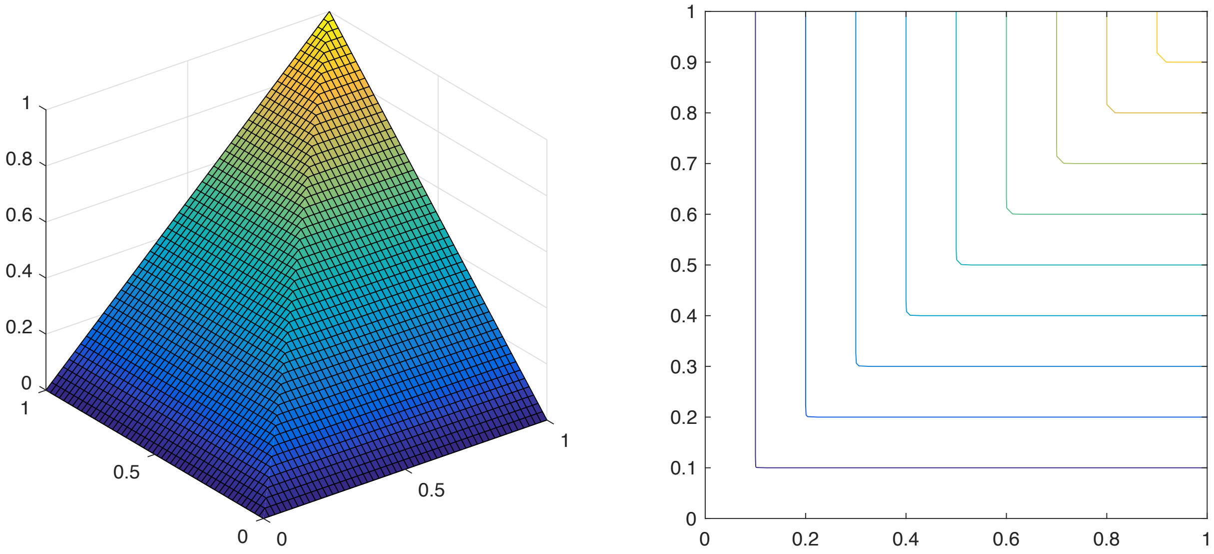


Figure 2: Upper Frechet-Hoeffding bound $M(u_1, u_2)$

2.3 Product Copula

The other important conception in copula is the product copula, which can be defined as,

$$\prod(u, v) = uv, \qquad \forall (u, v) \in I$$

The two dimensional product copula can be displayed as following:

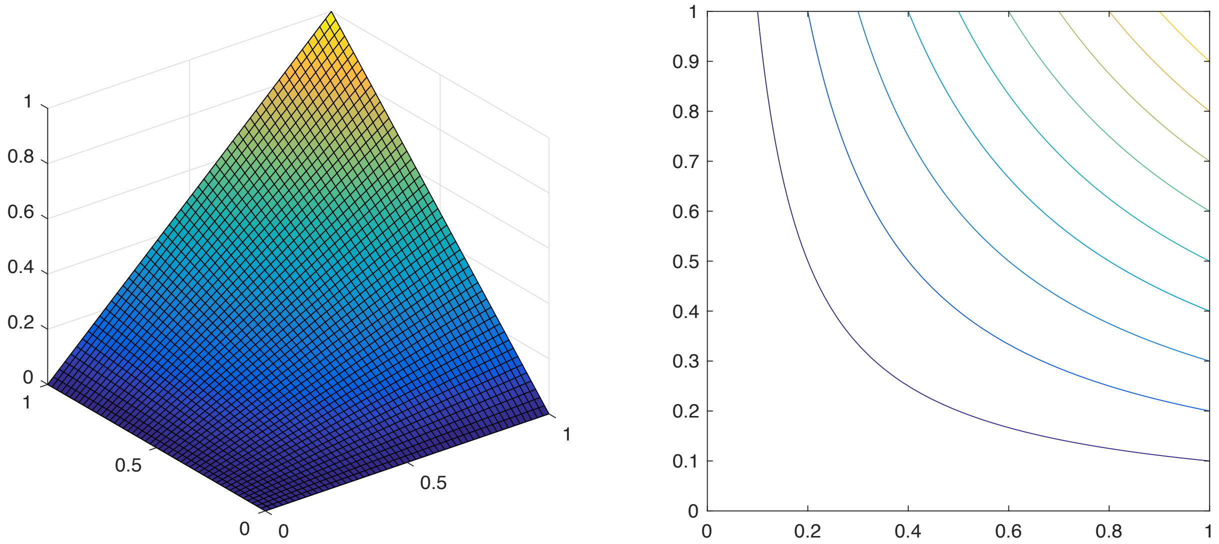


Figure 3: Product Copula $\prod(u_1, u_2)$

2.4 Two dimensional Gaussian Copula

For two dimensional Gaussian copula, it can be defined as following:

$$C^{Ga}(u,v) = \Phi_\rho(\Phi^{-1}(u), \Phi^{-1}(v))$$

where Φ_ρ is the joint distribution function of a two-dimensional standard normal variable, with linear correlation coefficient ρ. Therefore, the culmulative distribution function of two dimensional Gaussian copula is[1]

$$C^{Ga}(u,v) = \int_{-\infty}^{\Phi^{-1}(u)} \int_{-\infty}^{\Phi^{-1}(v)} \frac{1}{2\pi\sqrt{1-\rho^2}} exp\Big(\frac{-(r^2+s^2-2\rho rs)}{2(1-\rho^2)}\Big) drds$$

and the density can be and its corresponding contour can be displayed in Figure 4

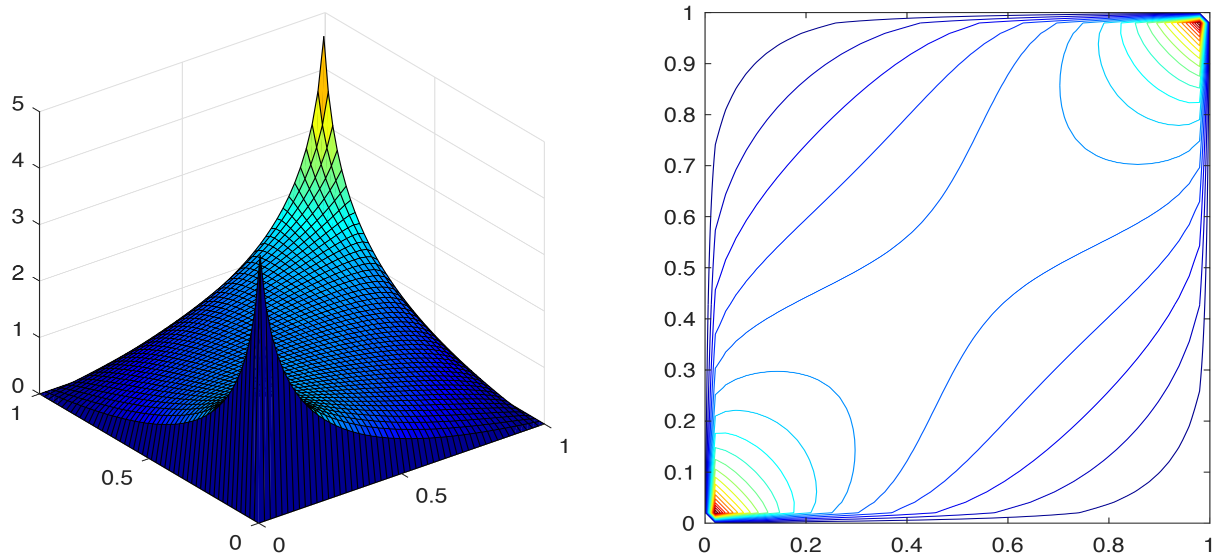


Figure 4: Two dimensional Gaussian copula

2.5 Multivariate Copula

In this part, we will give the definitions of the multivariate copula. For the n-dimensional copula, $n > 2$, the notions of groundedness and n-increasing are the extensions of the 2-dimensional case.

Definition 2.5.1 Let the function $G : R^n \to R$ has a domain $Dom\ G = A_1 \times A_2 \times \cdots \times A_n$, where the non-empty sets A_i have a least element a_i. The function G is called **grounded** if and only if for every $\mathbf{v} \in DomG$, with at least on index k such that $v_k = a_k$:

$$G(\mathbf{v}) = G(v_1, v_2, ..., v_{k-1}, a_k, v_{k+1}, ..., v_n) = 0$$

As the two-dimensional case, we need also to know the definition of n-increasing. Before to define that, it is essential to know the following notions:

Definition 2.5.2 The **n-box A** is defined as:

$$A = [u_{11}, u_{12}] \times [u_{21}, u_{22}] \times \cdots \times [u_{n1}, u_{n2}]$$

with $u_{i1} \leq u_{i2}$, $i = 1, 2, ..., n$. We can observe that an n-box is the Carestian product of n closed intervals

Let us denote with $\mathbf{w}$ any vertex of A and with $ver(A)$ the set of all vertices of A. $\mathbf{w} \in ver(A)$ if and only if its i-th component w_i, $i = 1, 2, ..., n$, is either equal to u_{i1} or to u_{i2}. Consider the product

$$\prod_{i=1}^{n} sgn(2w_i - u_{i1} - u_{i2})$$

Since each factor in the product is -1 if $w_i = u_{i1} < u_{i2}$, is equal to zero if $w_i = u_{i1} = u_{i2}$, and is 1 if $w_i = u_{i2} > u_{i1}$,

$$\prod_{i=1}^{n} sgn(2w_i - u_{i1} - u_{i2}) = \begin{cases} -1 & \text{if } u_{i1} \neq u_{i2},\ \forall i,\ \#\{i : w_i = u_{i2}\} = 2m+1 \\ 0 & \text{if } \exists i : u_{i1} = u_{i2} \\ 1 & \text{if } u_{i1} \neq u_{i2},\ \forall i,\ \#\{i : w_i = u_{i1}\} = 2m \end{cases} \qquad m \in N$$

If $ver(A) \subset DomG$, define the G-volume of A as the sum

$$\sum_{w \in ver(A)} G(\mathbf{w}) \prod_{i=1}^{n} sgn(2w_i - u_{i1} - u_{i2})$$

Now we can define the n-increasing function:
Definition 2.5.3 The function $G : A_1 \times A_1 \times \cdots \times A_n \rightarrow R$ is said to be n**-increasing** if the G-volume of A is non-negative for every n-box A for which $ver(A) \subset DomG$:

$$\sum_{w \in ver(A)} G(\mathbf{w}) \prod_{i=1}^{n} sgn(2w_i - u_{i1} - u_{i2}) \geq 0$$

Now we are ready for defining the copula in n dimensional space.
Definition 2.5.4 An n-dimensional copula is a function $C : A_1 \times A_2 \times \cdots \times A_n \rightarrow R$, where, for each i, $A_i \subset I$ and contains at least 0 and 1, such that

(i) C is grounded

(ii) C is n-increasing

(iii) its one-dimensional margins are the identity function on I: $C_i(u) = u$, $i = 1, 2, ..., n$.

Now, we will introduce the Sklar's theorem, which is significant in the copula application. Sklar's theorem provides the theoretical foundation when we use copula, it states the role that copula play in the relation between multivariate distribution function and their univariate margins.

Theorem 2.5.5 Let $F_1(x_1)$, $F_2(x_2)$,..., $F_n(x_n)$ be (given) marginal distribution functions. Then, for every $\mathbf{x} = (x_1, x_2, ..., x_n) \in R^n$:

(i) If C is any copula whose domain contains $RanF_1 \times RanF_2 \times \cdots \times RanF_n$,

$$C(F_1(x_1), F_2(x_2), ..., F_n(x_n))$$

is a joint distribution function with margins $F_1(x_1), F_2(x_2), ..., F_n(x_n)$.

(ii) Conversely, if F is a joint distribution function with margins

$$F_1(x_1), F_2(x_2), ..., F_n(x_n)$$

there exists a unique copula C, with domain $RanF_1 \times RanF_2 \times \cdots \times RanF_n$, such that

$$F(\mathbf{x}) = C(F_1(x_1), F_2(x_2), ..., F_n(x_n))$$

As it is in two dimensional copula case, the Multivariate copula also has its Frechet-Hoeffding upper and lower bounds: for any copula $C : [0,1]^d \to [0,1]$ and any $(u_1, ..., u_d) \in [0,1]^d$ the following bounds hold:

$$W(u_1, ..., u_d) \leq C(u_1, ..., u_d) \leq M(u_1, ..., u_d)$$

The function **W** is called lower Frechet-Hoeffding bound and it is defined as:

$$W(u_1, ..., u_d) = max\Big\{1 - d + \sum_{i=1}^{d} u_i, 0\Big\}$$

The function **M** is called upper Frechet-Hoeffding bound and it is defined as:

$$M(u_1, ..., u_d) = min\Big\{u_1, ..., u_d\Big\}$$

The multivariate product copula can be defined as $\prod(u_1, ..., u_d) = \prod_{i=1}^{d} u_i$ if and only if u_i are independent.

2.6 Multivariate Gaussian copula

Definition 2.6.1 Let R be a symmetric, positive definite matrix with diagonal elements one and the Φ_R the standized multivariate normal distribution with correlation matrix R. The **multivariate Gaussian copula** is defined as follows:

$$C_R^{Ga}(\mathbf{u}) = \Phi_R(\Phi^{-1}(u_1), \Phi^{-1}(u_2), ..., \Phi^{-1}(u_n))$$

where Φ^{-1}, as usual, is the inverse of the standard univariate normal distribution function Φ and Φ_R is the joint cumulative distribuiton function of a multivariate normal distribution with mean vector zero and the covariance matrix equal to the correlation matrix R. The density of the multivariate Gaussian copula can be approxiamted by the numerical integration as following:[3]

$$c_R^{Ga}(u) = \frac{1}{\sqrt{detR}} exp\left(-\frac{1}{2}\begin{pmatrix}\Phi^{-1}(u_1)\\ \vdots \\ \Phi^{-1}(u_d)\end{pmatrix}^T \cdot \left(R^{-1} - I\right) \cdot \begin{pmatrix}\Phi^{-1}(u_1)\\ \vdots \\ \Phi^{-1}(u_d)\end{pmatrix}\right)$$

where I is the indentity matrix.

And it is vital to know the following proposition about the Gaussian copula.

Proposition 2.6.2 The Gaussian copula generates the standard joint normal distribution function-via Sklar's Theorem-if and only if the margins are standard normal.[2]

3 Modelling

In this part, we will introduce how to establish copula model of four stocks. We will give the mathematical notions first and then build the model.

Step 1 Find the Gaussian Copula

Let us assume that there are d stocks data $S_i(t)$, $i = 1, 2, ..., d$, and $t = 1, 2, ..., n$. And X_i, $i = 1, 2, ..., d$ be the log increments (i.e. $X_i(t) = log(S_i(t)) - log(S_i(t-1))$) of the d stocks, and they are d continuous random variables. Each of the X_i has its empirical cumulative distribution function, which can be denoted as $F_i(X_i)$, $i = 1, ..., d$, i.e.

$$F_i(X_i) = \frac{1}{n}\sum_{j=1}^{n} \mathbf{1}(X_j \leq X_i)$$

To simplify, we use U_i to be the empirical cumulative distribution function of X_i, i.e. $U_i(t) = F_i(X_i(t))$, $i = 1, 2, ..., d$. Since U_i are the empirical CDF of four log returns and in generally we assume X_is are independently identically distributed, so the U_is are uniformly $[0, 1]$ distributed, and we will prove it graphically in the next section. In the next step, what we need to do is to find the standard normal inverse of U_i, which we denote as Y_i, $i = 1, 2, ..., d$, and

$$Y_i(t) = \Phi^{-1}(U_i(t))$$

We can prove that Y_i are standard normally distributed, since $U \sim Uniform[0, 1]$, so we have

$$P(U \leq u) = u$$

Futher, we can get,

$$\begin{aligned} P(Y \leq y) &= P(\Phi_i^{-1}(U) \leq y) \\ &= P(U \leq \Phi(y)) = \Phi(y) \end{aligned}$$

According to the **Proposition 2.6.2**, there exists the Gaussian copula C_R^{Ga} that connect the Y_i, since the margin are standard normlly distributed. We assume that C_R^{Ga} is the multivariate normally distributed with mean $\mathbf{0}$ and variance $\mathbf{R}$. i.e. $C_R^{Ga} \sim \mathbf{N}(\mathbf{0}, \mathbf{R})$, where the mean have d dimenstions and $\mathbf{R}$ is the covariance matrix of Y_i. i.e.

$$\mathbf{R} = \begin{pmatrix} Var(Y_1) & Cov(Y_1, Y_2) & \dots & Cov(Y_1, Y_d) \\ Cov(Y_2, Y_1) & Var(Y_2) & \dots & Cov(Y_2, Y_d) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(Y_d, Y_1) & Cov(Y_d, Y_2) & \dots & Var(Y_d) \end{pmatrix}$$

Step 2 Simulation and Comparison

From Step 1, we will get the Gaussian copula $C_R^{Ga} \sim \mathbf{N}(\mathbf{0}, \mathbf{R})$, so by applying mathematical software, we can randomly choose $\{y(t)\}_{t=1}^{n}$, so $y(t)$ will be d dimensional vector. And we can denote the $i - th$ column in $y(t)$ as $y_i(t)$, $i = 1, 2, ..., d$, so what we need to do is to return them to the original values as following:

$$\begin{aligned} u(t) &= (\Phi(y_1(t)), \Phi(y_2(t)), ..., \Phi(y_d(t))) \\ x(t) &= (F_1^{-1}(\Phi(y_1(t))), F_2^{-1}(\Phi(y_2(t))), ..., F_d^{-1}(\Phi(y_d(t)))) \\ &= (F_1^{-1}(u_1(t)), F_2^{-1}(u_2(t))), ..., F_d^{-1}(u_d(t)))) \end{aligned}$$

From the content above, we know that it is impossible to calculate the inverse by the regular method since the funciton is piecewise constant, so it is not invertible. Under this circumstance, let us introduce the definition of the **right inverse**:

Definition 3.2.1 Given a function $F : R \to [0, 1]$ we write

$$F^{-1}(u(t)) = inf\{x : F(x) \geq u(t)\} \qquad for \quad u(t) \in (0, 1)$$

by using this method to get the inverse $F^{-1}(u(t))$ and since,

$$x_i(t) = log(s_i(t)) - log(s_i(t-1))$$

so,

$$\sum_{j=1}^{t} x_i(j) = log(s_i(t)) - log(s_i(1))$$

$$\frac{s_i(t)}{s_i(1)} = exp(\sum_{j=1}^{t} x_i(j))$$

and finally we will get,

$$s_i(t) = s_i(1) \cdot exp(\sum_{j=1}^{t} x_i(j))$$

At the end of this step, we can compare the raw data and the simulated data by observing the contour plot of the empirical copula density respectively. We know that there is no specific formula of the empirical copula density, so we can utilize some methods to estimate. Here I choose to use the Gaussian Kernel density, so we need to introduce some relative definitions below.

Definition 3.2.2 Let $\mathbf{x}_1$, $\mathbf{x}_2$,..., $\mathbf{x}_n$ be a sample of d-variate random vectors drawn from a common distribution described by the density function f. The kernel density is defined to be[5]

$$\hat{f}_{\mathbf{H}}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i)$$

where,

(i) $\mathbf{x} = (x_1, x_2, ..., x_d)^T$, $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{id})^T$, $i = 1, 2, ..., n$ are d-vectors.

(ii) $\mathbf{H}$ is the bandwidth $d \times d$ matrix which is symmetric and positive definite.

(iii) K is the kernel function which is a symetric multivariate density.

Since we choose to utilize the Gaussian Kernel, so

$$\mathbf{K}_{\mathbf{H}}(\mathbf{x}) = (2\pi)^{-\frac{d}{2}}|\mathbf{H}|^{-\frac{1}{2}}e^{-\frac{1}{2}\mathbf{x}^T\mathbf{H}^{-1}\mathbf{x}}$$

For two dimensional case, for $\forall(u, v) \in [0, 1]^2$

$$\hat{f}(\mathbf{x}, \mathbf{H}) = \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i)$$

where, $\mathbf{x} = (x_1, x_2)^T$ and $\mathbf{x}_i = (x_{i1}, x_{i2})^T$, $i = 1, 2, ..., n$, $K_{\mathbf{H}}(\mathbf{x}) = |\mathbf{H}|^{-\frac{1}{2}}K(\mathbf{H}^{-\frac{1}{2}}\mathbf{x})$ and the choice of $K(\mathbf{x})$ here is the Gaussian, i.e.

$$K(\mathbf{x}) = \frac{1}{2\pi}exp(-\frac{1}{2}\mathbf{x}^T\mathbf{x}) \qquad \forall \mathbf{x} \in [0, 1]^2$$

$\mathbf{H}$ is the bandwidth matrix with $\mathbf{H} = \begin{pmatrix} h_1^2 & 0 \\ 0 & h_2^2 \end{pmatrix}$ and (h_1, h_2) is the bandwidth which minimize

$$argmin_{(h_1,h_2)}\mathbb{E}\Big[\int_0^1\int_0^1[\hat{f}(u, v) - f(u, v)]dudv\Big]$$

After the calculation above, finally we can get the two dimensional Guassian Kernel density can be estimated as

$$\hat{f}(\mathbf{x}, \mathbf{H}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h_1 \cdot h_2}K(\frac{x_1 - x_{i1}}{h_1}, \frac{x_2 - x_{i2}}{h_2})$$
$$K(\mathbf{x}) = \frac{1}{2\pi}exp(-\frac{1}{2}(x_1^2 + x_2^2))$$

After these steps, we will get the two dimensional contour plot and three dimensional surface between the two-stock combinations. In order to observe the accuracy of the simulation result by using copula model, we can simulate the data by some tranditional models and compare the result with the copula model. Here we choose to use the Gaussian model to simulate the data again and then compare the result with the copula model.

Definition 3.2.3 The multivariate Gaussian model of a d-dimensional random vector $\mathbf{X} = [X_1, X_2, ..., X_d]$ can be written in the following notation:[4]

$$\mathbf{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

where $\boldsymbol{\mu}$ is the mean vector,

$$\boldsymbol{\mu} = \begin{pmatrix} mean(X_1) \\ mean(X_2) \\ \vdots \\ mean(X_d) \end{pmatrix}^T$$

and $\boldsymbol{\Sigma}$ is the covariance matrix of $\mathbf{X}$,

$$\boldsymbol{\Sigma} = \begin{pmatrix} Var(X_1) & Cov(X_1, X_2) & \dots & Cov(X_1, X_d) \\ Cov(X_2, X_1) & Var(X_2) & \dots & Cov(X_2, X_d) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_d, X_1) & Cov(X_d, X_2) & \dots & Var(X_d) \end{pmatrix}$$

After these steps, we can find the distributions of those X_is which simulated by the Gaussian model, and then compare the kernel density between the two-stocks combinations by showing the three dimensional surf and two dimensional contour plot.

Step 3 Portfolio

In this section, we will establish a portfolio of the d stocks by using coupula model.

1. Establish a portfolio of the d stocks: $P = \sum_{i=1}^{d} S_i(t)$.

2. In Step 2, we got $s_i(t) = s_i(1) \cdot exp(\sum_{j=1}^{t} x_i(j))$. Now we set $s_i(1) = M$, $i = 1, 2, ..., d$, and then we do the simulation T times, after this process we will got $s_i(t)$.

3. Calculate the result $S = \sum_{i=1}^{d} s_i(t)$ and observe the trends of the portfolio.

4 Data Analysis and Results

In this section, we will study four stocks in the Swedish market, they are very famours Swedish companies: AstraZeneca, Ericsson, Volvo and SEB. By using the mathematical software, we will finish the data fetching, processing and simulating by different models, and then do the portfolio for these four stocks.

4.1 Data Processing

To begin with, we should download the raw data. In this report, we firstly fetch the close price from 1/1/2011 to 1/1/2017 of these four stocks, 1559 in total, from *yahoo finance*, so we can easliy plot the close price curve of them.

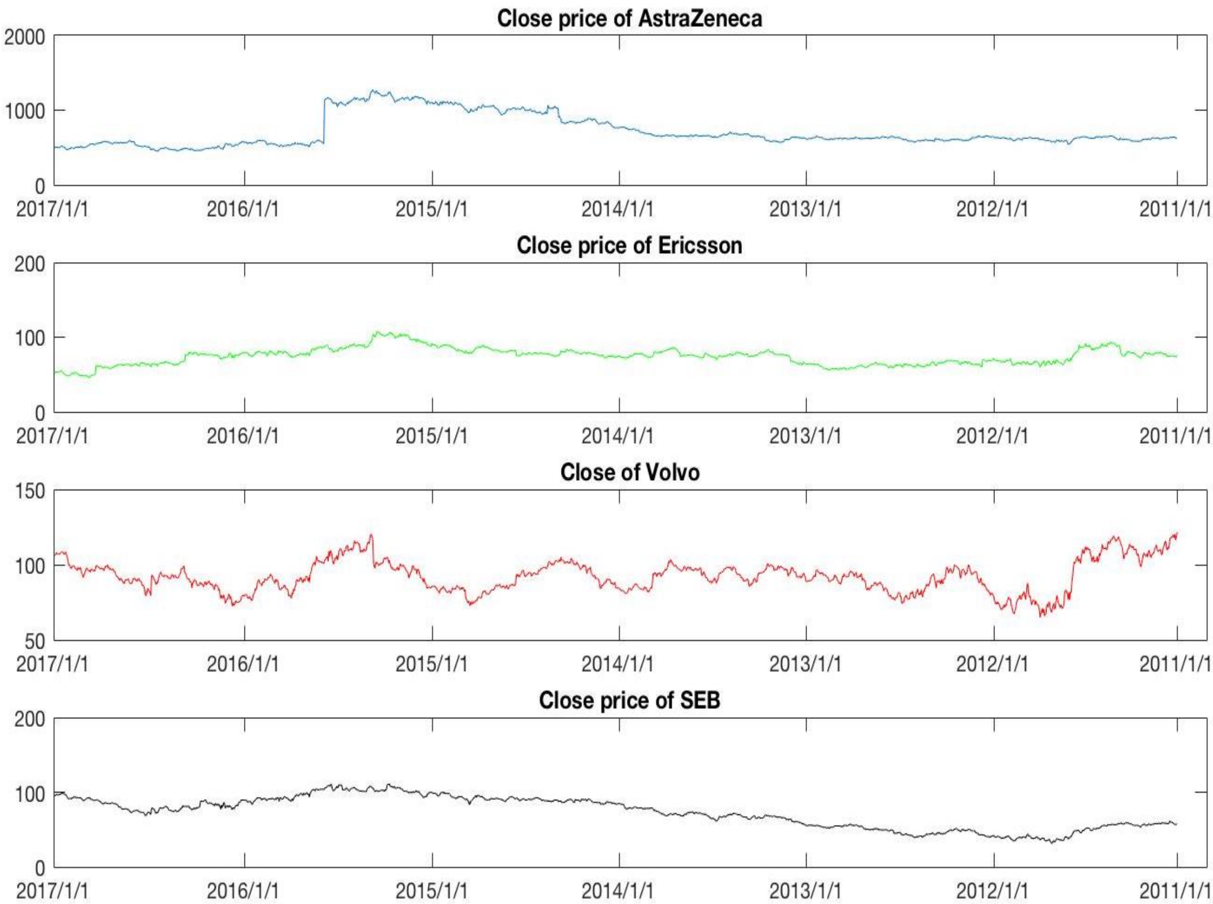


Figure 5: Stock curve of close price

From the **Figure 5**, we can observe that there is a obvious jump of the stock's price in AstraZeneca between the trading year 2015 to 2016, we call this phenomenon as **Stock Split**:

Definition 4.1.1 A **Stock Split** is a decision by the company's board of directors to increase the number of shares that are outstanding by issuing more shares to current shareholders. For example, in a 2-for-1 stock split, every shareholder with one stock is given an additional share. So if a company had 10 million shares outstanding before the split, it will have 20 million shares outstanding after a 2-for-1 split.

The price of the stock will be affected by the stock split. After the split, the stock price will reduce since the number of shares has increased, so the close cannot reflect the stocks' price accurately. To avoid the the split, we should choose to use the **Adjusted Close Price** as our raw data.

Definition 4.1.2 An **Adjusted Close Price** is a stock's closing price on any given day of trading that has been amended to include any distributions and corporate actions that occurred at any time prior to the next day's open. The adjusted closing price uses the closing price as a starting point, but it takes into account factors such as dividends, stock splits and new stock offerings. The adjusted closing price represents a more accurate reflection of a stock's value, since distributions and new offerings can alter the closing price.

As what we have discussed above, we can fetch the adjusted close price from the *finance*, and then we have the price curve as following:

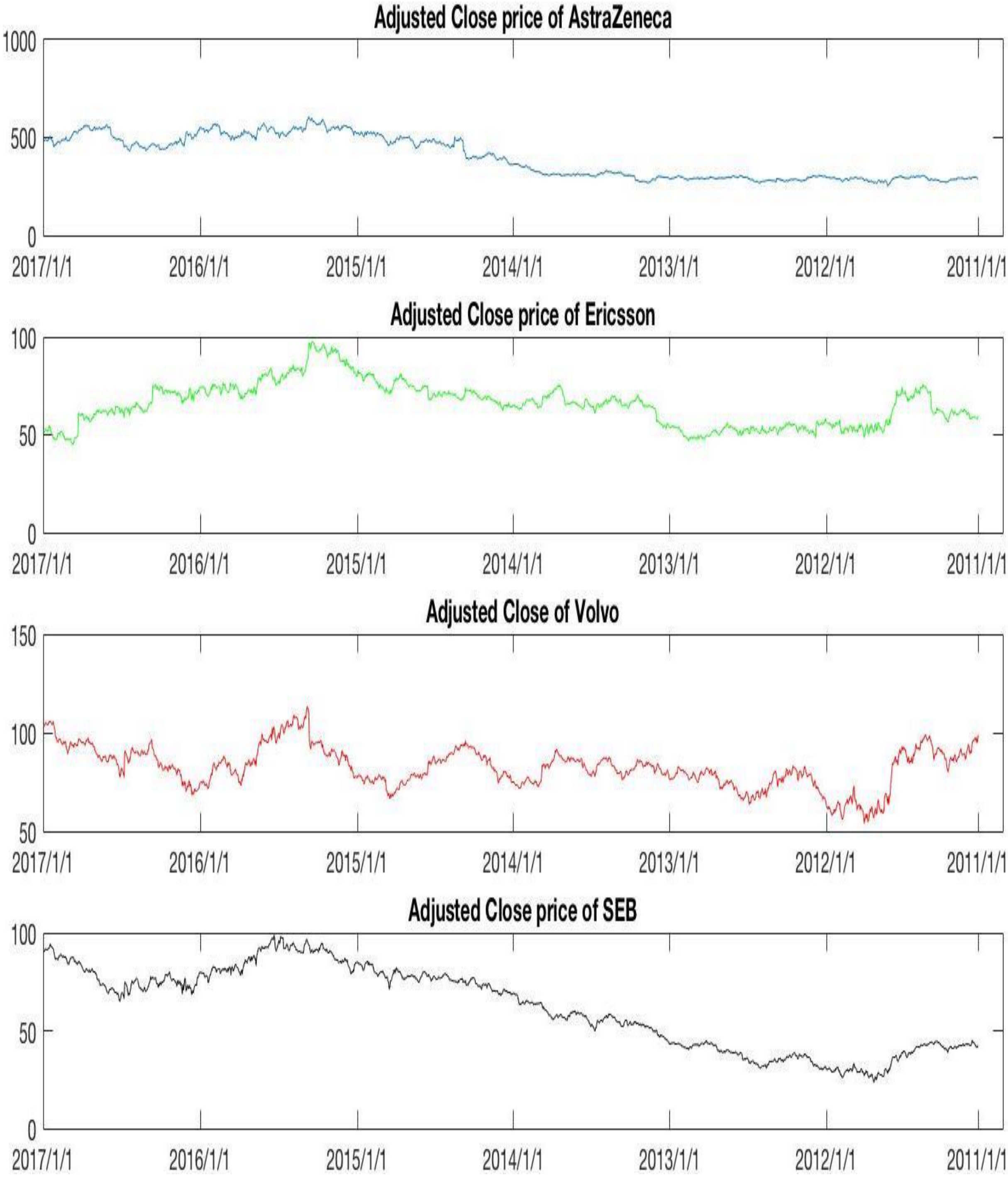


Figure 6: Stock curve of adjusted close price

In the stocks market, we often assume that the stocks are some normally distributed noise, but in the most of the practical cases, they are not normal distribution, we can check our data by the graphical method and measure the skewness and kurtosis, i.e. the third and the fourth moment of the data, which can reflect their distributions.

We can firstly plot the distribution of S_i as following:

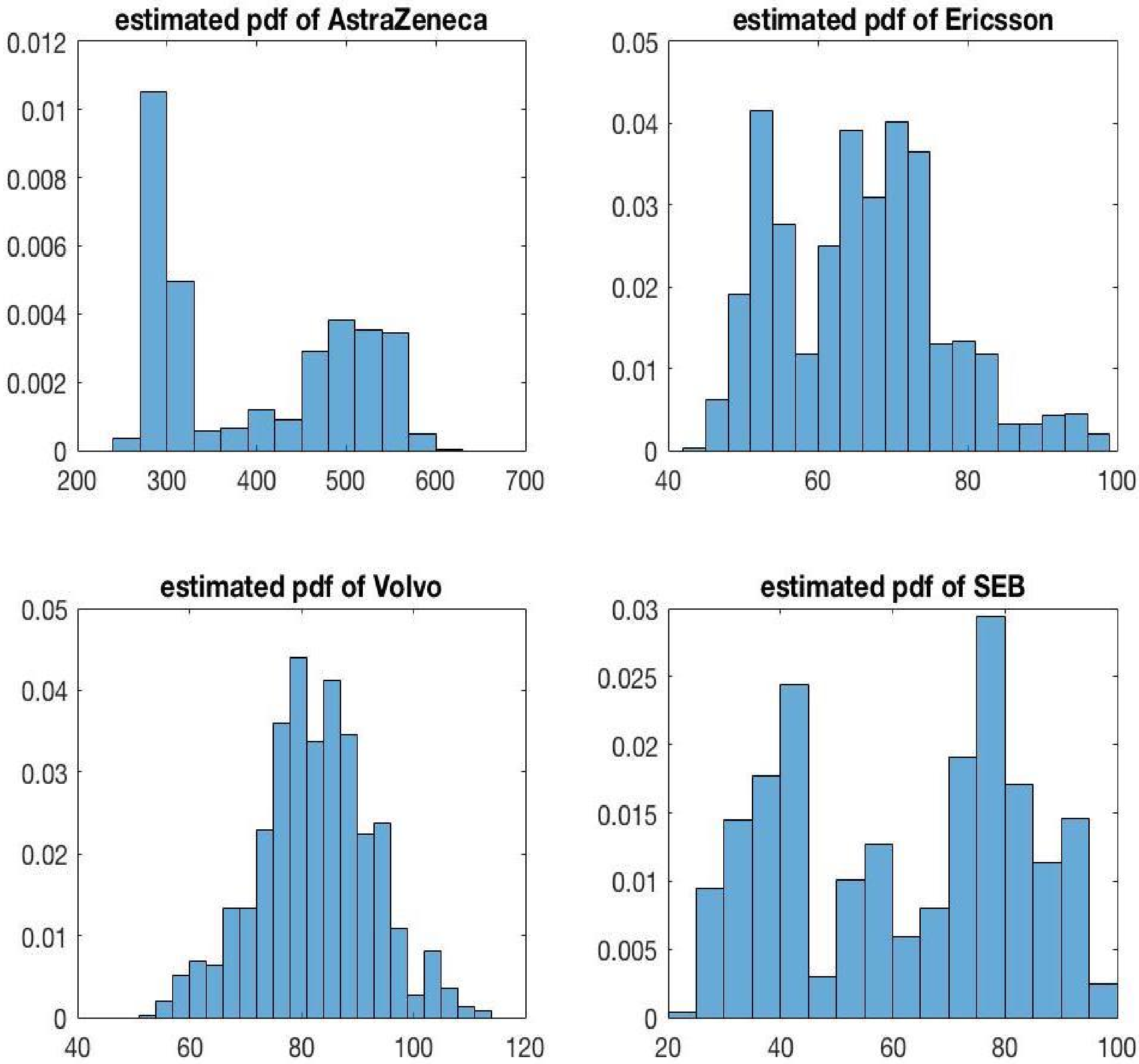


Figure 7: the distribution of S_i

The skewness and kurtosis of the four stocks are:

	S_1	S_2	S_3	S_4
skewness	0.2323	0.3974	-0.0449	-0.1553
kurtosis	1.3662	2.7765	3.0417	1.6344

From the Figure7 and the table above, we can observe that all of the stocks skews on a certain extent, especically for S_1 and S_2, which means they are not symetrically distributed. And the kurtosis of them reflect that S_1 and S_4 have the light tails. On conclusion, the four stocks are not strictly normal.

And next we can calculate and plot the log increments $X_i(t)$ of all stocks,

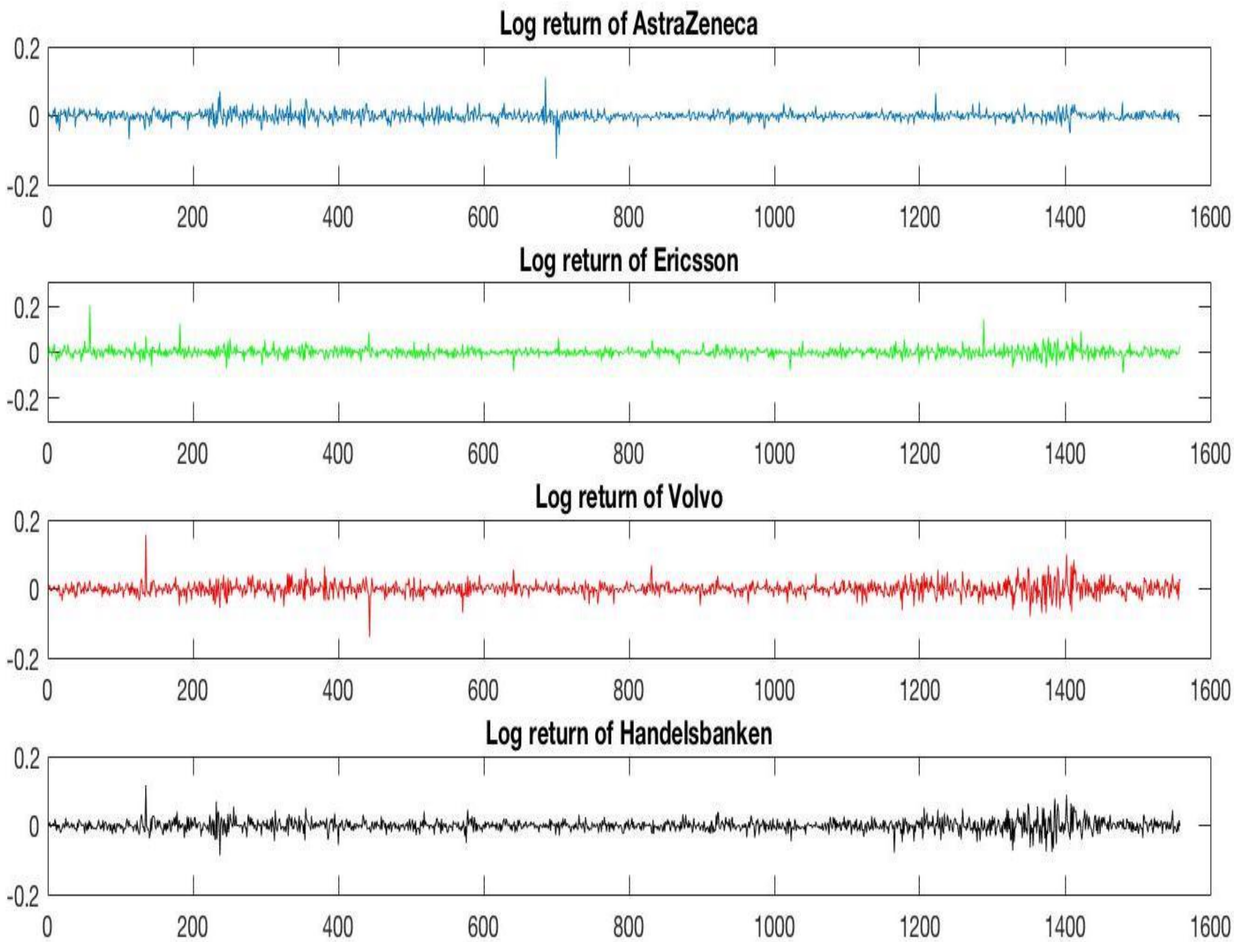


Figure 8: Stock log increment

Generally, we assume that the log increments, i.e. $X_i(t)$, $i = 1, 2, 3, 4$, are independently identically distributed, so the distributions of the log returns should be uniformly $[0, 1]$ distributed. By using of math software, we can find the distributions $U_i(t) = F_i(X_i(t))$, we can check if U_i are uniformly $[0, 1]$ distributed by observing the following plot,

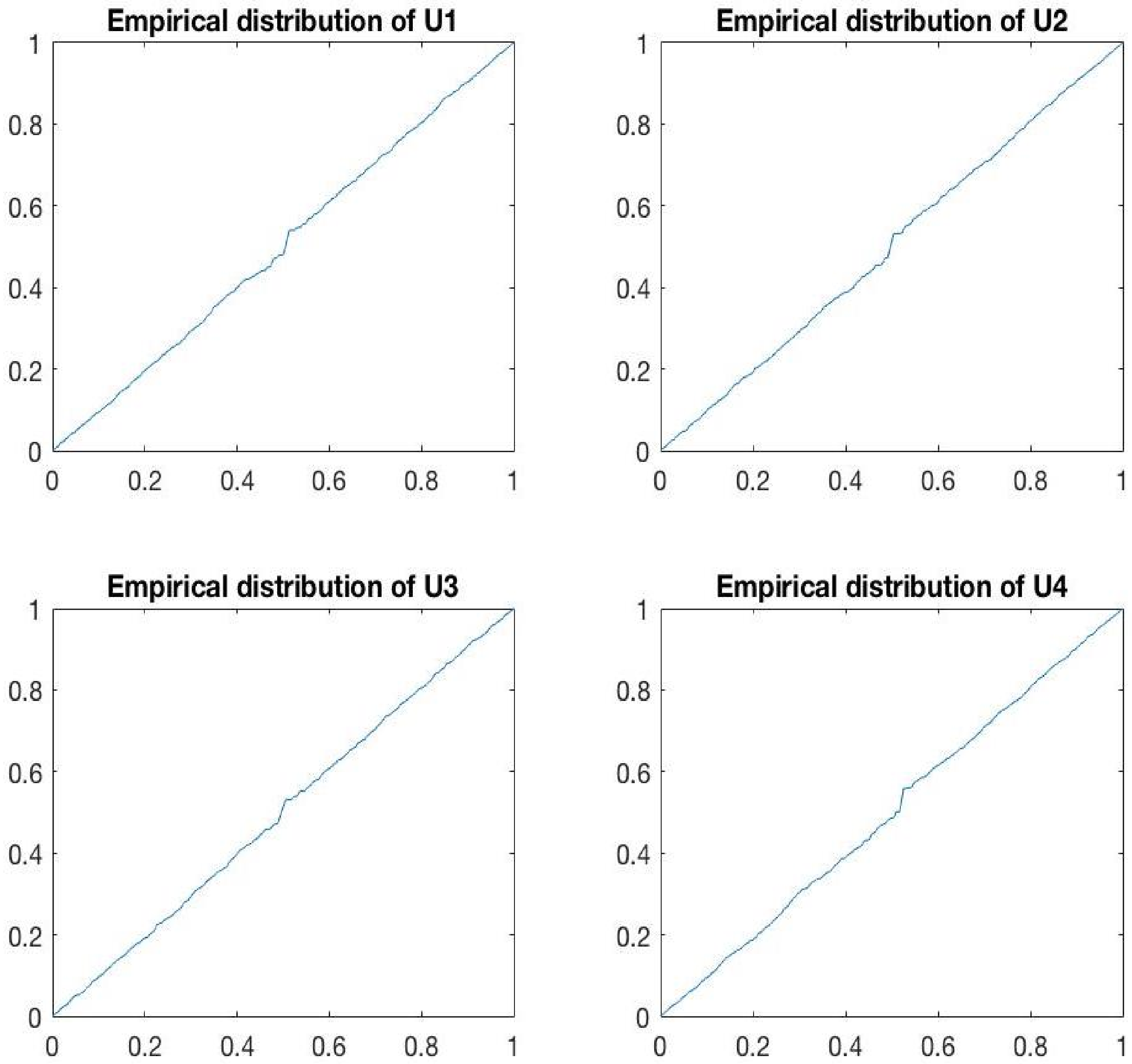


Figure 9: Empirical CDF of U_i

From Figure 9, it is obvious that U_is are approximately distributed in $Uniform[0,1]$, and next we could calculate the values of the inverse of standard normal distribution of $U_i(t)$, we denote this value as $Y_i(t)$, that is $Y_i(t) = \Phi^{-1}(U_i(t))$, $i = 1,2,3,4$. From the Step 1 of last section, we know that $Y_i(t)$ will be (approximately) normal distribution, which is proved on the **Section 3.2**. After this step, we should recall the **Definition 2.6.1**, and then we follow the Step 1 of last section, by applying mathematical software, we can finally establish the Multivariate Gaussian Copula Model C_R^{Ga}:

$$C_R^{Ga} \sim \boldsymbol{N}\left([0,0,0,0], \begin{pmatrix} 0.9655 & 0.3328 & 0.3363 & 0.3715 \\ 0.3328 & 0.9656 & 0.4776 & 0.4204 \\ 0.3363 & 0.4776 & 0.9662 & 0.6222 \\ 0.3715 & 0.4204 & 0.6222 & 0.9659 \end{pmatrix}\right)$$

4.2 Simulation

Next we will use the results above to simulate by using Multivarite Copula Model and Gaussian Model respectively. Let us begin from the copula model. To start with, we randomly choose 5000 variables from C_R^{Ga}, we denote them as $y(t)$, and $y(t)$ has four dimensions, so it is 5000×4 matrix. Then we follow the Step 2 in the last section, we will return $y(t)$ to $u_i(t)$. And then we can calculate the kernel density by using Matlab.

As for the Gaussian model, first we can also simulate 5000 data by the Gaussian Model,

$$\boldsymbol{N}\left(\begin{pmatrix} mean(X_1) \\ mean(X_2) \\ \vdots \\ mean(X_d) \end{pmatrix}^T, \begin{pmatrix} Var(X_1) & Cov(X_1,X_2) & \dots & Cov(X_1,X_d) \\ Cov(X_2,X_1) & Var(X_2) & \dots & Cov(X_2,X_d) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_d,X_1) & Cov(X_d,X_2) & \dots & Var(X_d) \end{pmatrix}\right)$$

we will also get the simulated data, which can be denoted as $x_{i_}$G and use the method as the Step 1 of last section to get the distribution of $x_{i_}$G, which can be denoted as $u_{i_}$G, and finally we can make a comparison of the raw data and simulated data. In this thesis, I choose to utilize the three dimensional surf and two dimensional contour plot to compare the kernel density of the raw data and simulated data. Since we have four stocks, so we have $C_4^2 = 6$ combinations. We will observe and analysis all of them in the following.

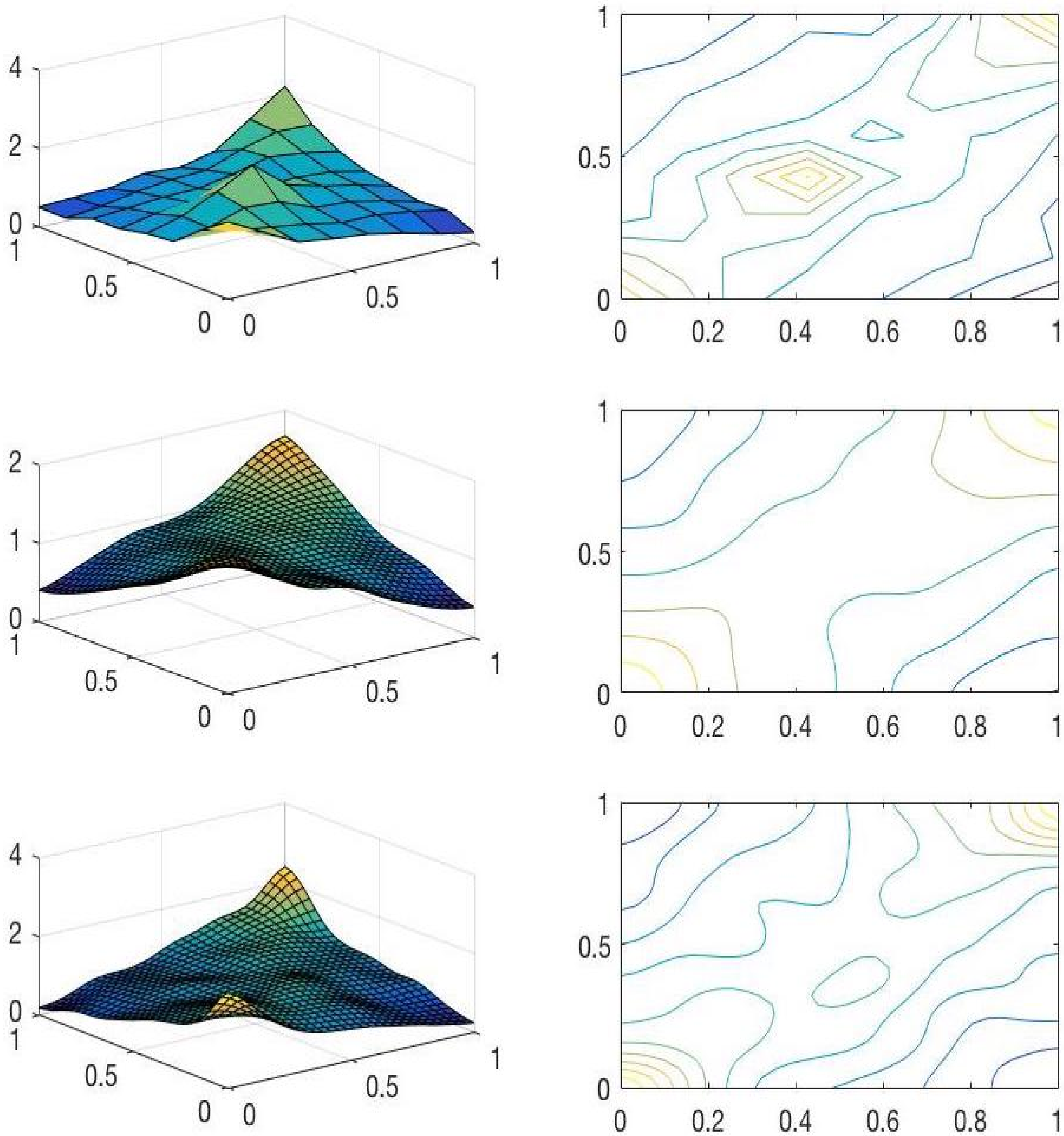


Figure 10: Empirical copula density between AstraZeneca and Ericsson

Figure 10 shows the comparison between AstraZeneca and Ericsson, the first row are the kernel density surf and contour plot of raw data, the second raw are the kernel density surf and contour plot of data simulated by Gaussian Model, and the third raw are the kernel density surf and contour plot of data simulated by Copula Model. We can observe that the Copula Model is better than the Gaussian, since the former one is not only shows the positive relation of two stocks but shows the peak around $[0.5, 0.5]$.

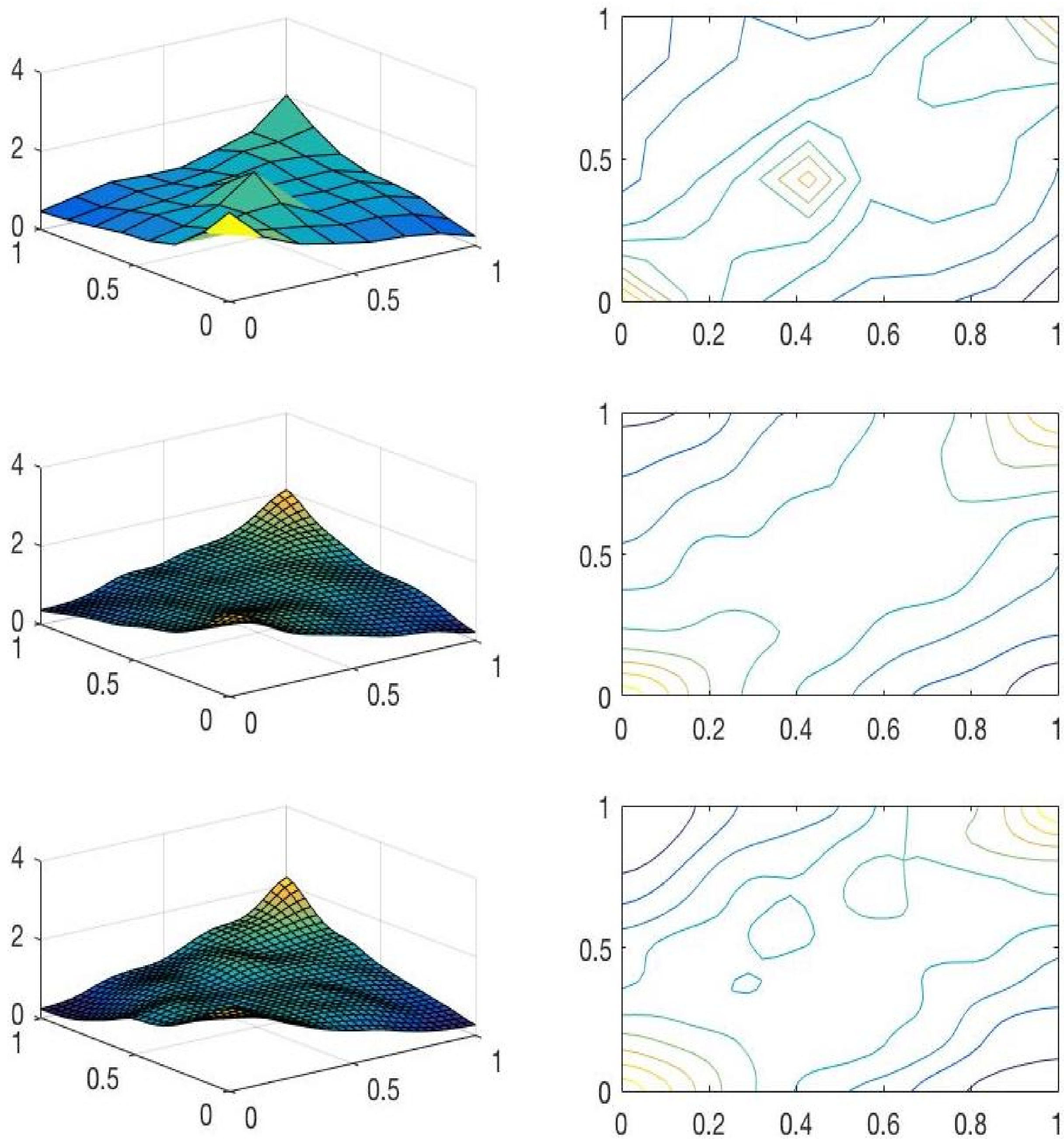


Figure 11: Empirical copula density between AstraZeneca and Volvo

Figure 11 shows the comparison between AstraZeneca and Volvo, the first row are the kernel density surf and contour plot of raw data, the second raw are the kernel density surf and contour plot of data simulated by Gaussian Model, and the third raw are the kernel density surf and contour plot of data simulated by Copula Model, and the plots depict that the Copula Model is better than the Gaussian, since the obvious positive relation and peak are in the contour plots.

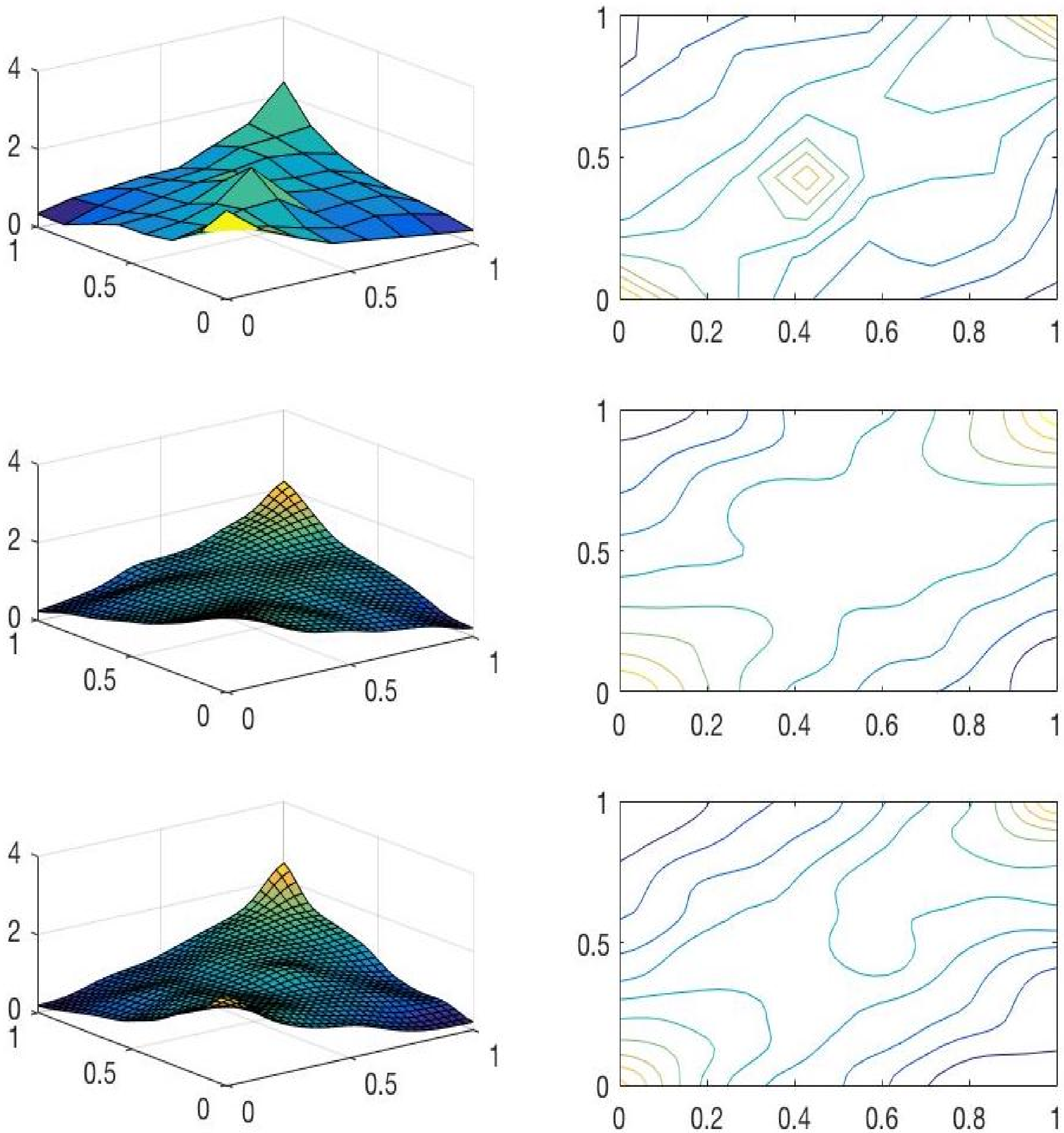


Figure 12: Empirical copula density between AstraZeneca and SEB

Figure 12 shows the comparison between AstraZeneca and SEB, the first row are the kernel density surf and contour plot of raw data, the second raw are the kernel density surf and contour plot of data simulated by Gaussian Model, and the third raw are the kernel density surf and contour plot of data simulated by Copula Model, and it can be observed that the Copula Model is still a little better than the Gaussian, since from the contour plots, we can see the positive relation and peak in the approximate similar place.

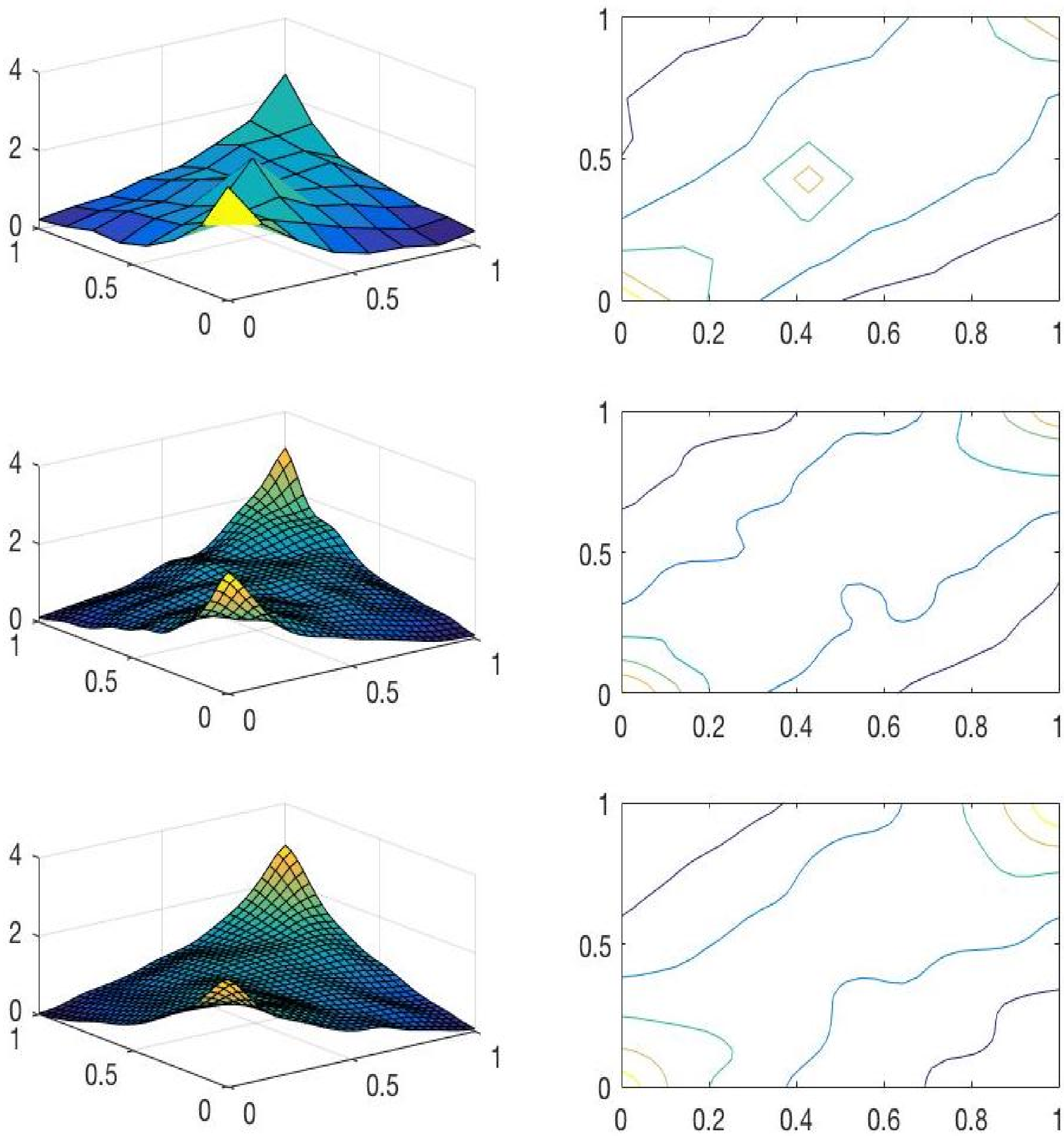


Figure 13: Empirical copula density between Ericsson and Volvo

Figure 13 shows the comparison between Ericsson and Volvo, the first row are the kernel density surf and contour plot of raw data, the second raw are the kernel density surf and contour plot of data simulated by Gaussian Model, and the third raw are the kernel density surf and contour plot of data simulated by Copula Model. From this group, the simulations of Gaussian and Copula are simular, so it is not very obvious that which one is better.

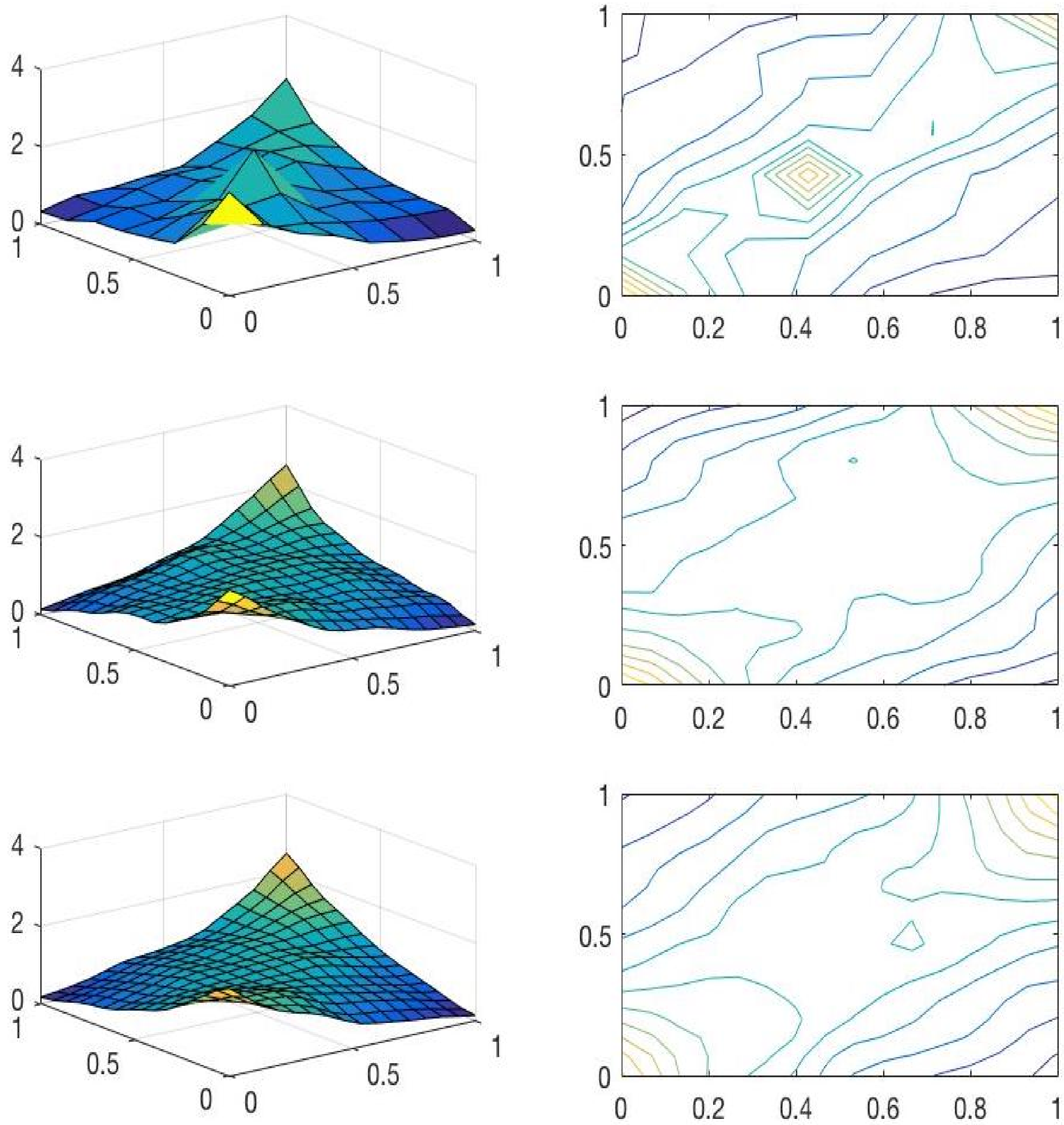


Figure 14: Empirical copula density between Ericsson and SEB

Figure 14 shows the comparison between Ericsson and SEB, the first row are the kernel density surf and contour plot of raw data, the second raw are the kernel density surf and contour plot of data simulated by Gaussian Model, and the third raw are the kernel density surf and contour plot of data simulated by Copula Model, and it can be observed that the Copula Model is still a little better than the Gaussian, since from the contour plots, we can also see the positive relation and peak in the approximate similar place.

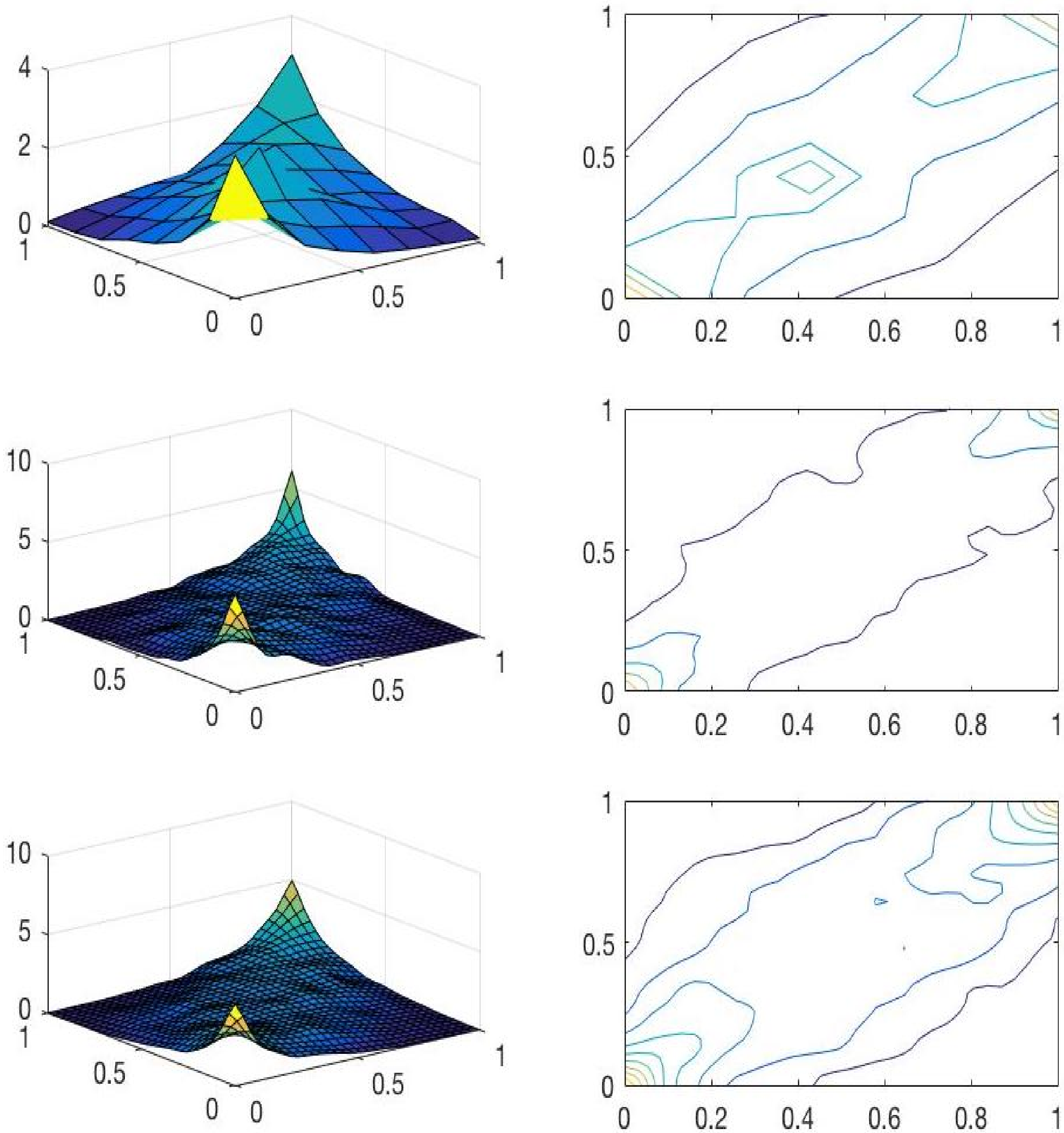


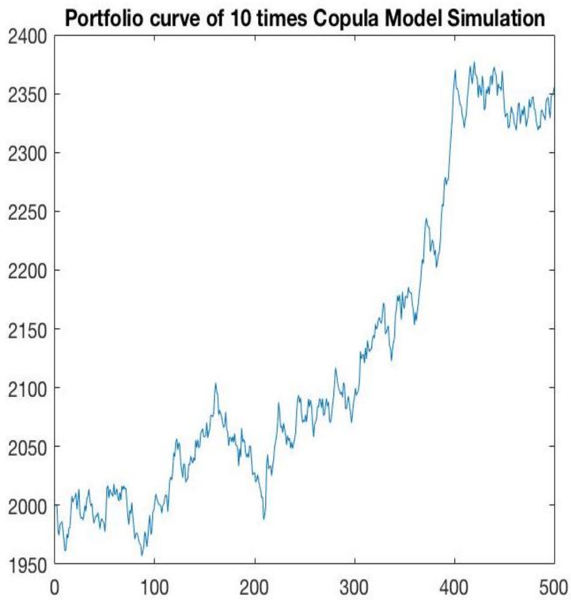
Figure 15: Empirical copula density between Volvo and SEB

Figure 15 shows the comparison between Volvo and SEB, the first row are the kernel density surf and contour plot of raw data, the second raw are the kernel density surf and contour plot of data simulated by Gaussian Model, and the third raw are the kernel density surf and contour plot of data simulated by Copula Model, and it is obvious that the Copula Model is better than the Gaussian, since from the contour plots, we can observe that the copula simulation is more similar to the raw data.

4.3 Portfolio and Risk Analysing

In the last part, we will establish the portfolio of the four stocks by using the Gaussian Model and Copula Model respectively. To begin with, Let us set $M = s_i(1) = 1000$, $i = 1, 2, 3, 4$. And then, we can follow the steps of last section, we can simulate approximate two years data (500) of two models several times (in this thesis I will simulate each model 10 times and 1000 times) and calculate the sample mean of the results. In the end, we can calculate the portfolio value $S = \sum_{i=1}^{4} s_i(t)$ and plot the results by using Matlab.

As for Copula Model, following the steps above, we will get,



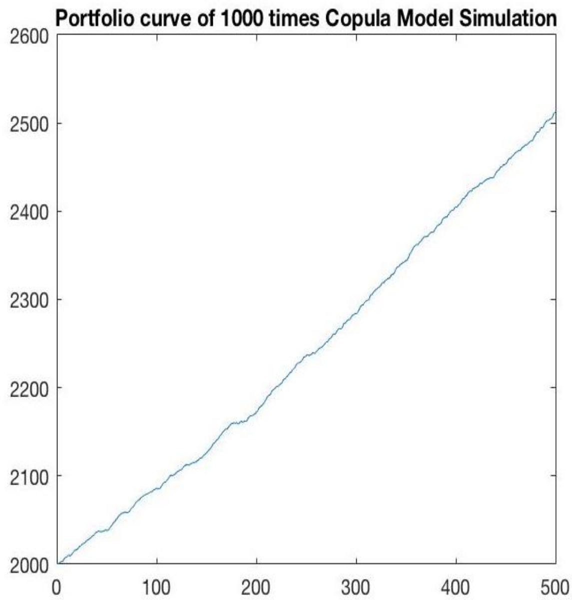


Figure 16: Portfolio curve of Copula Model Simulaiton

From the left plot in Figure 16, we can observe that the portfolio curve is fluctuant and it has a increasing trend, at the point $t = 410$, the portfolio value will attach to the maximum, it is about 2380, while the portfolio value will attach to the minimum value at the point $t = 90$. In the curve, there is an obvious decrease, i.e. from $t = 160$ to $t = 205$, and we can also observe that the curve ascent dramatically from $t = 370$ to $t = 400$. However, if we simulate the model 1000 times, from the right plot in Figure 16, we can observe that the portfolio curve will approximate to a line, and it will attach to the maximum value at the end point $t = 500$.

As for Gaussian Model, following the steps above and doing the same times simulations as the Copula Model, we will get,

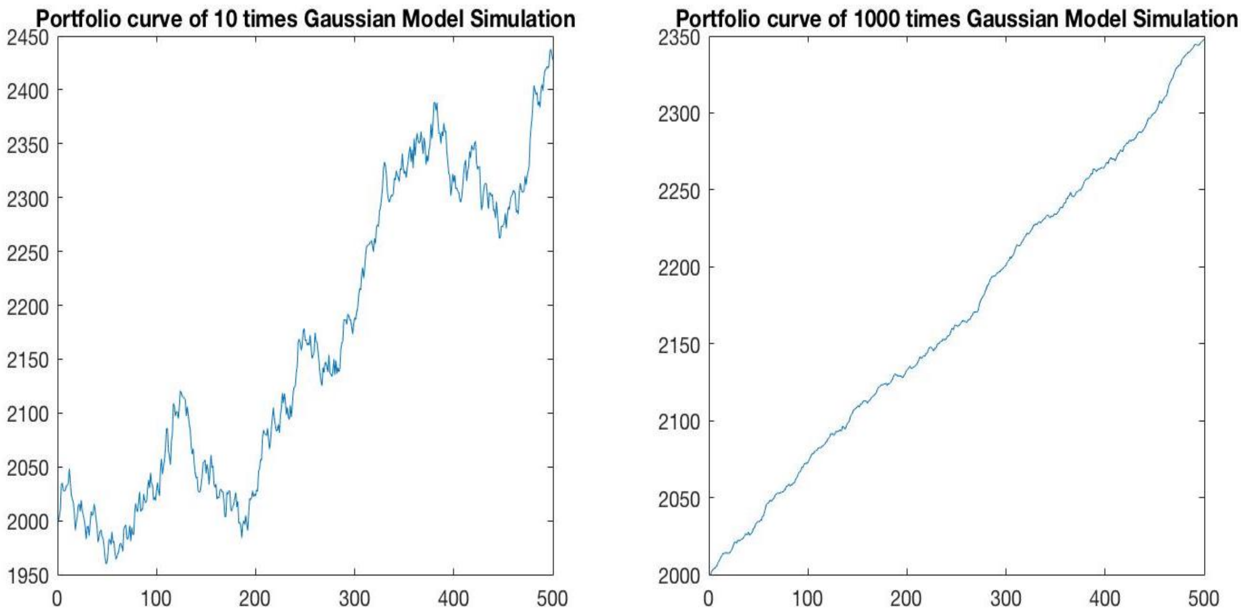


Figure 17: Portfolio curve of Copula Model Simulaiton

From the left plot in Figure 17, we can observe that the portfolio curve is similar to the Copula Model, it is also fluctuant and it has a increasing trend, at the end point, the portfolio value will attach to the maximum, while the portfolio value will attach to the minimum value at the point $t = 50$. In the curve, there are two obvious decreases, i.e. from $t = 120$ to $t = 200$ and from $t = 390$ to $t = 450$, and we can also observe that the curve is increasing from $t = 190$ to $t = 400$. But if we simulate the model 1000 times, from the right plot in Figure 17, we can observe that the portfolio curve will approximate to a line, and it will attach to the maximum value at the end point $t = 500$.

From the Figure 16 and Figure 17, it seems not obvious that Copula Model is better than Gaussian Model, so we will make the comparison more clear in the next step. In order to more accurate, let us simulate both model 10000 times and calculate the portfolio. Then we save the 10000 end point values of each model, that is we have 10000 values $t = 500$ of each model. We can estimate the probability density of the portfolio values at the end point, and we will get the curves in the following:

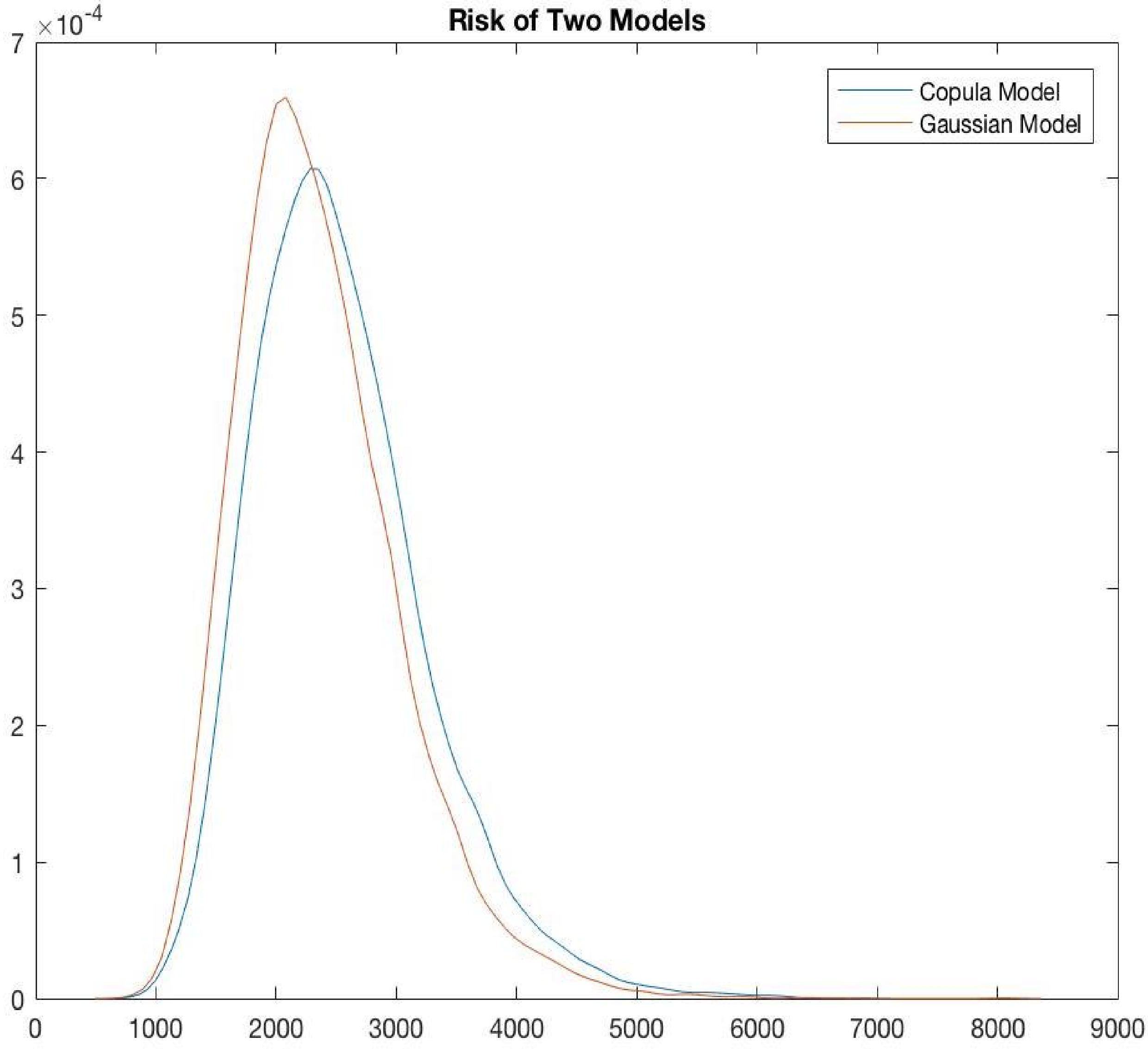


Figure 18: Risk of Two Models

From the Figure 18, we can observe that Copula Model has the smaller risk.

5 Conclusion

From the figures in the last section, we can observe that the most of raw data and the simulated data by copula model are distributed similarly. What we can also see from the figures is the price of AstraZeneca seems has strong positive dependence with Ericsson and Volvo, and the price of SEB seems have strong dependence with Ericsson and Volvo. If the data has strong dependence, it will be a good news to execute the portfolio next.

As the figures of the portfolio, we can observe that the portfolio increase both of the two models, and from the plots of 1000 times simulation, we can observe that the copula model simulation increase a little more than the Gaussian model. In the meantime, we can also observe that the copula model has the smaller risk than Gaussian model.

References

[1] Michel Goossens, Elisa Luciano, and Walter Vecchiato. *Copula Methods in Finance*. John Wiley and Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2003

[2] Piotr Jaworski, Fabrizio Durante, Wolfgang Härdle, Tomasz Rychlik. *Copula Theory and Its Applications*. Proceedings of the Workshop Held in Warsaw, 25-26 September 2009

[3] WIKIPEDIA, Copula (probability theory)
`https://en.wikipedia.org/wiki/Copula_(probability_theory)`

[4] WIKIPEDIA, Multivariate normal distribution
`https://en.wikipedia.org/wiki/Multivariate_normal_distribution`

[5] Z. I. BOTEV1, J. F. GROTOWSKI and D. P. KROESE *KERNEL DENSITY ESTIMATION VIA DIFFUSION*. University of Queensland, The Annals of Statistics, 2010

Appendix

Matlab Function Part:

```
%% collect the data from Yahoo Finance
function data = get_yahoo_data(tick_name, freq, fromdate, todate)
start_date = strsplit(fromdate, '-');
end_date = strsplit(todate, '-');
try
url = strcat('https://ichart.finance.yahoo.com/table.csv?s=',...
    tick_name,'&a=',start_date(2),'&b=',start_date(3),'&c=',start_date(1),
    ...'&d=', end_date(2),'&e=',end_date(3),'&f=',end_date(1),'&g=',freq,
    ...'&ignore=.csv');
catch ERROR
    disp(ERROR)
    return
end
[temp, status] = urlread(cell2mat(url));
data = containers.Map
temp = strsplit(temp, {'\n',','});
for i = 1:1:7
    key = temp{i};
    data(key) = temp(i+7:7:end);
end
end

%2D Kernel Density Estimation
function [bandwidth,density,X,Y]=kde2d(data,n,MIN_XY,MAX_XY)
global N A2 I
```

```
if nargin<2
    n=2^8;
end
n=2^ceil(log2(n)); % round up n to the next power of 2;
N=size(data,1);
if nargin<3
    MAX=max(data,[],1); MIN=min(data,[],1); Range=MAX-MIN;
    MAX_XY=MAX+Range/4; MIN_XY=MIN-Range/4;
end
scaling=MAX_XY-MIN_XY;
if N<=size(data,2)
    error('data has to be an N by 2 array where each row represents a two ...
    dimensional observation')
end
transformed_data=(data-repmat(MIN_XY,N,1))./repmat(scaling,N,1);
%bin the data uniformly using regular grid;
initial_data=ndhist(transformed_data,n);
% discrete cosine transform of initial data
a= dct2d(initial_data);
% now compute the optimal bandwidth^2
  I=(0:n-1).^2; A2=a.^2;

t_star=fzero( @(t)(t-evolve(t)),[0,0.1]);
% options = optimset('FunValCheck','off');
%    try
%         t_star=fzero( @(t)(t-evolve(t)),[0,0.1],options);
%    catch
%         defaultError('override error, be cautious!');
```

```
%         t_star=0;
%     end

p_02=func([0,2],t_star);p_20=func([2,0],t_star); p_11=func([1,1],t_star);
t_y=(p_02^(3/4)/(4*pi*N*p_20^(3/4)*(p_11+sqrt(p_20*p_02))))^(1/3);
t_x=(p_20^(3/4)/(4*pi*N*p_02^(3/4)*(p_11+sqrt(p_20*p_02))))^(1/3);
% smooth the discrete cosine transform of initial data using t_star
a_t=exp(-(0:n-1)'.^2*pi^2*t_x/2)*exp(-(0:n-1).^2*pi^2*t_y/2).*a;
% now apply the inverse discrete cosine transform
if nargout>1
    density=idct2d(a_t)*(numel(a_t)/prod(scaling));
    [X,Y]=meshgrid(MIN_XY(1):scaling(1)/(n-1):MAX_XY(1),MIN_XY(2):...
    scaling(2)/(n-1):MAX_XY(2));
end
bandwidth=sqrt([t_x,t_y]).*scaling;
end
%#######################################
function  [out,time]=evolve(t)
global N
Sum_func = func([0,2],t) + func([2,0],t) + 2*func([1,1],t);
time=(2*pi*N*Sum_func)^(-1/3);
out=(t-time)/time;
end
%#######################################
function out=func(s,t)
global N
if sum(s)<=4
    Sum_func=func([s(1)+1,s(2)],t)+func([s(1),s(2)+1],t); const=...
```

```
    (1+1/2^(sum(s)+1))/3;
    time=(-2*const*K(s(1))*K(s(2))/N/Sum_func)^(1/(2+sum(s)));
    out=psi(s,time);
else
    out=psi(s,t);
end


end
%#######################################
function out=psi(s,Time)
global I A2
% s is a vector
w=exp(-I*pi^2*Time).*[1,.5*ones(1,length(I)-1)];
wx=w.*(I.^s(1));
wy=w.*(I.^s(2));
out=(-1)^sum(s)*(wy*A2*wx')*pi^(2*sum(s));
end
%#######################################
function out=K(s)
out=(-1)^s*prod((1:2:2*s-1))/sqrt(2*pi);
end
%#######################################
function data=dct2d(data)
% computes the 2 dimensional discrete cosine transform of data
% data is an nd cube
[nrows,ncols]= size(data);
if nrows~=ncols
    error('data is not a square array!')
```

```
end
% Compute weights to multiply DFT coefficients
w = [1;2*(exp(-i*(1:nrows-1)*pi/(2*nrows))).'];
weight=w(:,ones(1,ncols));
data=dct1d(dct1d(data)')';
    function transform1d=dct1d(x)

        % Re-order the elements of the columns of x
        x = [ x(1:2:end,:); x(end:-2:2,:) ];

        % Multiply FFT by weights:
        transform1d = real(weight.* fft(x));
    end
end
%########################################
function data = idct2d(data)
% computes the 2 dimensional inverse discrete cosine transform
[nrows,ncols]=size(data);
% Compute wieghts
w = exp(i*(0:nrows-1)*pi/(2*nrows)).';
weights=w(:,ones(1,ncols));
data=idct1d(idct1d(data)');
    function out=idct1d(x)
        y = real(ifft(weights.*x));
        out = zeros(nrows,ncols);
        out(1:2:nrows,:) = y(1:nrows/2,:);
        out(2:2:nrows,:) = y(nrows:-1:nrows/2+1,:);
    end
```

```
end
%#######################################
function binned_data=ndhist(data,M)
[nrows,ncols]=size(data);
bins=zeros(nrows,ncols);
for i=1:ncols
    [dum,bins(:,i)] = histc(data(:,i),[0:1/M:1],1);
    bins(:,i) = min(bins(:,i),M);
end
% Combine the  vectors of 1D bin counts into a grid of nD bin
% counts.
binned_data = accumarray(bins(all(bins>0,2),:),1/nrows,M(ones(1,ncols)));
end
```

Main Code:

```
%% collect the data from Yahoo Finance

clear all

S1=get_yahoo_data('AZN.ST','d','2011-00-01','2017-00-01');

S1=S1('Adj Close')';

S1=str2num(char(S1));

S11=flipud(S1);


S2=get_yahoo_data('ERIC-A.ST','d','2011-00-01','2017-00-01');

S2=S2('Adj Close')';

S2=str2num(char(S2));

S22=flipud(S2);


S3=get_yahoo_data('VOLV-B.ST','d','2011-00-01','2017-00-01');

S3=S3('Adj Close')';

S3=str2num(char(S3));

S33=flipud(S3);


S4=get_yahoo_data('SEB-A.ST','d','2011-00-01','2017-00-01');

S4=S4('Adj Close')';

S4=str2num(char(S4));

S44=flipud(S4);


%% skewness and kertosis

sk1=skewness(S1);

sk2=skewness(S2);

sk3=skewness(S3);

sk4=skewness(S4);
```

```
ku1=kurtosis(S1);
ku2=kurtosis(S2);
ku3=kurtosis(S3);
ku4=kurtosis(S4);

%% log return calculation
X1=diff(log(S11));
X2=diff(log(S22));
X3=diff(log(S33));
X4=diff(log(S44));
n=length(X1);

%% cdf
%Find the covariance matrix
U1=ksdensity(X1,X1,'function','cdf');
U2=ksdensity(X2,X2,'function','cdf');
U3=ksdensity(X3,X3,'function','cdf');
U4=ksdensity(X4,X4,'function','cdf');

Y1=norminv(U1);
Y2=norminv(U2);
Y3=norminv(U3);
Y4=norminv(U4);
Y=[Y1,Y2,Y3,Y4];
mu=zeros(1,4);
sigma=cov(Y);
```

```
%% copula model simulation
m=5000;
y=mvnrnd(mu,sigma,m);
u1=min(max(U1),normcdf(y(:,1)));
u2=min(max(U2),normcdf(y(:,2)));
u3=min(max(U3),normcdf(y(:,3)));
u4=min(max(U4),normcdf(y(:,4)));

%% Gaussian model simulation
mu_x=[mean(X1),mean(X2),mean(X3),mean(X4)];
X=[X1,X2,X3,X4];
sigma_x=cov(X);
l=5000;
Gaussian_x=mvnrnd(mu_x,sigma_x,l);
x1_G=Gaussian_x(:,1);
x2_G=Gaussian_x(:,2);
x3_G=Gaussian_x(:,3);
x4_G=Gaussian_x(:,4);
u1_G=ksdensity(x1_G,x1_G,'function','cdf');
u2_G=ksdensity(x2_G,x2_G,'function','cdf');
u3_G=ksdensity(x3_G,x3_G,'function','cdf');
u4_G=ksdensity(x4_G,x4_G,'function','cdf');

%% emprical copula pdf estimate
[b,f12,cx,cy]=kde2d([U1,U2],8,[0,0],[1,1]);
subplot(3,2,1)
surf(cx,cy,f12)
subplot(3,2,2)
```

```
contour(cx,cy,f12)
[b,f12g,cx,cy]=kde2d([u1_G,u2_G],20,[0,0],[1,1]);
subplot(3,2,3)
surf(cx,cy,f12g)
subplot(3,2,4)
contour(cx,cy,f12g)
[b,f12c,cx,cy]=kde2d([u1,u2],20,[0,0],[1,1]);
subplot(3,2,5)
surf(cx,cy,f12c)
subplot(3,2,6)
contour(cx,cy,f12c)

figure
[b,f13,cx,cy]=kde2d([U1,U3],8,[0,0],[1,1]);
subplot(3,2,1)
surf(cx,cy,f13)
subplot(3,2,2)
contour(cx,cy,f13)
[b,f13g,cx,cy]=kde2d([u1_G,u3_G],20,[0,0],[1,1]);
subplot(3,2,3)
surf(cx,cy,f13g)
subplot(3,2,4)
contour(cx,cy,f13g)
[b,f13c,cx,cy]=kde2d([u1,u3],20,[0,0],[1,1]);
subplot(3,2,5)
surf(cx,cy,f13c)
subplot(3,2,6)
contour(cx,cy,f13c)
```

```
figure
[b,f14,cx,cy]=kde2d([U1,U4],8,[0,0],[1,1]);
subplot(3,2,1)
surf(cx,cy,f14)
subplot(3,2,2)
contour(cx,cy,f14)
[b,f14g,cx,cy]=kde2d([u1_G,u4_G],20,[0,0],[1,1]);
subplot(3,2,3)
surf(cx,cy,f14g)
subplot(3,2,4)
contour(cx,cy,f14g)
[b,f14c,cx,cy]=kde2d([u1,u4],20,[0,0],[1,1]);
subplot(3,2,5)
surf(cx,cy,f14c)
subplot(3,2,6)
contour(cx,cy,f14c)

figure
[b,f23,cx,cy]=kde2d([U2,U3],8,[0,0],[1,1]);
subplot(3,2,1)
surf(cx,cy,f23)
subplot(3,2,2)
contour(cx,cy,f23)
[b,f23g,cx,cy]=kde2d([u2_G,u3_G],20,[0,0],[1,1]);
subplot(3,2,3)
surf(cx,cy,f23g)
subplot(3,2,4)
```

```
contour(cx,cy,f23g)
[b,f23c,cx,cy]=kde2d([u2,u3],20,[0,0],[1,1]);
subplot(3,2,5)
surf(cx,cy,f23c)
subplot(3,2,6)
contour(cx,cy,f23c)

figure
[b,f24,cx,cy]=kde2d([U2,U4],8,[0,0],[1,1]);
subplot(3,2,1)
surf(cx,cy,f24)
subplot(3,2,2)
contour(cx,cy,f24)
[b,f24g,cx,cy]=kde2d([u2_G,u4_G],12,[0,0],[1,1]);
subplot(3,2,3)
surf(cx,cy,f24g)
subplot(3,2,4)
contour(cx,cy,f24g)
[b,f24c,cx,cy]=kde2d([u2,u4],12,[0,0],[1,1]);
subplot(3,2,5)
surf(cx,cy,f24c)
subplot(3,2,6)
contour(cx,cy,f24c)

figure
[b,f34,cx,cy]=kde2d([U3,U4],8,[0,0],[1,1]);
subplot(3,2,1)
surf(cx,cy,f34)
```

```
subplot(3,2,2)
contour(cx,cy,f34)
[b,f34g,cx,cy]=kde2d([u3_G,u4_G],20,[0,0],[1,1]);
subplot(3,2,3)
surf(cx,cy,f34g)
subplot(3,2,4)
contour(cx,cy,f34g)
[b,f34c,cx,cy]=kde2d([u3,u4],20,[0,0],[1,1]);
subplot(3,2,5)
surf(cx,cy,f34c)
subplot(3,2,6)
contour(cx,cy,f34c)

%% portfolio Copula
% x1=ksdensity(X1,u1,'function','icdf');
% x2=ksdensity(X2,u2,'function','icdf');
% x3=ksdensity(X3,u3,'function','icdf');
% x4=ksdensity(X4,u4,'function','icdf');

l=499;  %sample size
h=1000; %simulation times
s0=500;  %initial value

s1=zeros((l+1),1);
s2=zeros((l+1),1);
s3=zeros((l+1),1);
s4=zeros((l+1),1);
s1_c=zeros((l+1),h);
```

```
s2_c=zeros((l+1),h);
s3_c=zeros((l+1),h);
s4_c=zeros((l+1),h);
s1(1)=s0;
s2(1)=s0;
s3(1)=s0;
s4(1)=s0;
% s1(1)=S11(1);
% s2(1)=S22(1);
% s3(1)=S33(1);
% s4(1)=S44(1);

for j=1:h
   y=mvnrnd(mu,sigma,l);
   u1=min(max(U1),normcdf(y(:,1)));
   u2=min(max(U2),normcdf(y(:,2)));
   u3=min(max(U3),normcdf(y(:,3)));
   u4=min(max(U4),normcdf(y(:,4)));
   x1=zeros(l,1);
   x2=zeros(l,1);
   x3=zeros(l,1);
   x4=zeros(l,1);
   for i=1:l
     ind1=find(U1>=u1(i));
     x1(i)=min(X1(ind1));
     ind2=find(U2>=u2(i));
     x2(i)=min(X2(ind2));
     ind3=find(U3>=u3(i));
```

```
    x3(i)=min(X3(ind3));
    ind4=find(U4>=u4(i));
    x4(i)=min(X4(ind4));
  end


  for i=2:(l+1)
      s1(i)=s1(1)*exp(sum(x1(1:(i-1))));
      s2(i)=s2(1)*exp(sum(x2(1:(i-1))));
      s3(i)=s3(1)*exp(sum(x3(1:(i-1))));
      s4(i)=s4(1)*exp(sum(x4(1:(i-1))));
  end
  s1_c(:,j)=s1;
  s2_c(:,j)=s2;
  s3_c(:,j)=s3;
  s4_c(:,j)=s4;
end
m1_c=mean(s1_c,2);
m2_c=mean(s2_c,2);
m3_c=mean(s3_c,2);
m4_c=mean(s4_c,2);
p_c=m1_c+m2_c+m3_c+m4_c;
plot(p_c)
s1_cl=s1_c((l+1),:)';
s2_cl=s2_c((l+1),:)';
s3_cl=s3_c((l+1),:)';
s4_cl=s4_c((l+1),:)';
p_cl=s1_cl+s2_cl+s3_cl+s4_cl;
```

```
%% portfolio Gaussian
l=499;  %sample size
h=1000; %simulation times
s0=500;  %initial value
mu_x=[mean(X1),mean(X2),mean(X3),mean(X4)];
X=[X1,X2,X3,X4];
sigma_x=cov(X);
s11=zeros((l+1),1);
s22=zeros((l+1),1);
s33=zeros((l+1),1);
s44=zeros((l+1),1);
s1_g=zeros((l+1),h);
s2_g=zeros((l+1),h);
s3_g=zeros((l+1),h);
s4_g=zeros((l+1),h);
s11(1)=s0;
s22(1)=s0;
s33(1)=s0;
s44(1)=s0;
for j=1:h
   Gaussian_x=mvnrnd(mu_x,sigma_x,l);
   x1_G=Gaussian_x(:,1);
   x2_G=Gaussian_x(:,2);
   x3_G=Gaussian_x(:,3);
   x4_G=Gaussian_x(:,4);
   for i=2:(l+1)
       s11(i)=s11(1)*exp(sum(x1_G(1:(i-1))));
```

```
        s22(i)=s22(1)*exp(sum(x2_G(1:(i-1))));
        s33(i)=s33(1)*exp(sum(x3_G(1:(i-1))));
        s44(i)=s44(1)*exp(sum(x4_G(1:(i-1))));
    end
    s1_g(:,j)=s11;
    s2_g(:,j)=s22;
    s3_g(:,j)=s33;
    s4_g(:,j)=s44;
end
m1_g=mean(s1_g,2);
m2_g=mean(s2_g,2);
m3_g=mean(s3_g,2);
m4_g=mean(s4_g,2);
p_g=m1_g+m2_g+m3_g+m4_g;
plot(p_g)
s1_gl=s1_g((l+1),:)';
s2_gl=s2_g((l+1),:)';
s3_gl=s3_g((l+1),:)';
s4_gl=s4_g((l+1),:)';
p_gl=s1_gl+s2_gl+s3_gl+s4_gl;

%% risk
ksdensity(p_cl)
hold on
ksdensity(p_gl)
legend('Copula Model','Gaussian Model')
title('Risk of Two Models')
```

```
%% 3 copulas

%Frecht-Hoeffing upper bound copula M(u,v)

s=50;

x=linspace(0,1,s);

[X1,X2] = meshgrid(x,x);

F=copulacdf('Gaussian',[X1(:) X2(:)],1-exp(-10));

subplot(1,2,1)

surf(X1,X2,reshape(F,s,s));

subplot(1,2,2)

contour(X1,X2,reshape(F,s,s));

figure


%Frecht-Hoeffing lower bound copula W(u,v)

[X1,X2] = meshgrid(x,x);

F=copulacdf('Gaussian',[X1(:) X2(:)],-(1-exp(-10)));

subplot(1,2,1)

surf(X1,X2,reshape(F,s,s));

subplot(1,2,2)

contour(X1,X2,reshape(F,s,s));

figure


%The product copula

[X1,X2] = meshgrid(x,x);

F=copulacdf('Gaussian',[X1(:) X2(:)],0);

subplot(1,2,1)

surf(X1,X2,reshape(F,s,s));

subplot(1,2,2)

contour(X1,X2,reshape(F,s,s));
```

```
%% 2-dimensional Gaussian copula
clear all
[x,y]=meshgrid(linspace(0,1,50));
P=copulapdf('Gaussian',[x(:),y(:)],0.5);
subplot(1,2,1)
surf(x,y,reshape(P,[50,50]));
colormap jet;
subplot(1,2,2)
contour(x,y,reshape(P,[50,50]),20);

%% curve
subplot(4,1,1)
plot(S1)
title('Adjusted Close price of AstraZeneca')
subplot(4,1,2)
plot(S2,'color','g')
title('Adjusted Close price of Ericsson')
subplot(4,1,3)
plot(S3,'color','r')
title('Adjusted Close of Volvo')
subplot(4,1,4)
plot(S4,'color','k')
title('Adjusted Close price of SEB')

%% S PDF
subplot(2,2,1)
histogram(S1,'Normalization','pdf')
```

```
title('estimated pdf of AstraZeneca')
subplot(2,2,2)
histogram(S2,'Normalization','pdf')
title('estimated pdf of Ericsson')
subplot(2,2,3)
histogram(S3,'Normalization','pdf')
title('estimated pdf of Volvo')
subplot(2,2,4)
histogram(S4,'Normalization','pdf')
title('estimated pdf of SEB')

%% LOG curve
subplot(4,1,1)
plot(X1)
title('Log return of AstraZeneca')
subplot(4,1,2)
plot(X2,'color','g')
title('Log return of Ericsson')
subplot(4,1,3)
plot(X3,'color','r')
title('Log return of Volvo')
subplot(4,1,4)
plot(X4,'color','k')
title('Log return of Handelsbanken')

%% empirical distribution of U
[f1,ux1]=ecdf(U1);
[f2,ux2]=ecdf(U2);
```

```
[f3,ux3]=ecdf(U3);
[f4,ux4]=ecdf(U4);

subplot(2,2,1)
plot(ux1,f1)
title('Empirical distribution of U1')
subplot(2,2,2)
plot(ux2,f2)
title('Empirical distribution of U2')
subplot(2,2,3)
plot(ux3,f3)
title('Empirical distribution of U3')
subplot(2,2,4)
plot(ux4,f4)
title('Empirical distribution of U4')
```