

R routines for performing estimation and statistical process control under copula-based time series models

Takeshi Emura, Ting-Hsuan Long & Li-Hsien Sun

To cite this article: Takeshi Emura, Ting-Hsuan Long & Li-Hsien Sun (2017) R routines for performing estimation and statistical process control under copula-based time series models, Communications in Statistics - Simulation and Computation, 46:4, 3067-3087, DOI: 10.1080/03610918.2015.1073303

To link to this article: <http://dx.doi.org/10.1080/03610918.2015.1073303>



Accepted author version posted online: 28 Sep 2015.
Published online: 28 Sep 2015.



Submit your article to this journal [↗](#)



Article views: 58



View related articles [↗](#)



View Crossmark data [↗](#)

R routines for performing estimation and statistical process control under copula-based time series models

Takeshi Emura, Ting-Hsuan Long, and Li-Hsien Sun

Graduate Institute of Statistics, National Central University, Zhongli, Taoyuan, Taiwan

ABSTRACT

Modeling serial dependence in time series is an important step in statistical process control. We provide a set of automatic routines useful for simulating and analyzing time series under a copula-based serial dependence. First, we introduce routines that generate time series data under a given copula. Second, we provide fully automated routines for obtaining maximum likelihood estimates for given time series data and then drawing a Shewhart-type control chart. Finally, real data are analyzed for illustration. We make the routines available as “Copula.Markov” package in R.

ARTICLE HISTORY

Received 21 October 2014
Accepted 13 July 2015

KEYWORDS

Clayton copula; Joe copula;
Newton–Raphson algorithm;
Shewhart control chart

MATHEMATICS SUBJECT CLASSIFICATION

62M10; 62H12; 62H20; 60J20

1. Introduction

Modeling serial correlations in time series data plays a fundamental role in statistical process control (SPC) (Montgomery, 2009). Many observed data from daily manufacturing process are not independent since the present operating conditions depend on the past. For instance, a high-observation in chemical concentration may be followed by a high-observation in the next sequence of products (Bisgaard and Kulahci, 2007). Hence, appropriate models for serial dependence are necessary, which have been extensively studied in the literature. A comprehensive overview of the serial correlation models in SPC is found in Wieringa (1999) and Knoth and Schmid (2004), while a more concise review is available in Box and Narasimhan (2010). The literature mostly focuses on the first-order (Markov) models, including a first-order autoregressive AR(1), a first-order moving average MA(1), and a first-order integrated moving average IMA(1) or IMA(1,1). Recently, Long and Emura (2014) consider a copula-based Markov chain model to perform SPC for serially correlated data.

Any bivariate distribution function $H(y_1, y_2)$ with the marginal distributions $G_1(y_1) = H(y_1, \infty)$ and $G_2(y_2) = H(\infty, y_2)$ can be written as

$$H(y_1, y_2) = C(G_1(y_1), G_2(y_2)), \quad (1)$$

where $C : [0, 1]^2 \rightarrow [0, 1]$ is called copula (Nelsen, 2006). Conversely, if we give any copula C , and marginal distribution functions G_1 and G_2 , Eq. (1) yields a bivariate distribution function. This implies that a copula characterizes the correlation structure separately from the marginal distributions.

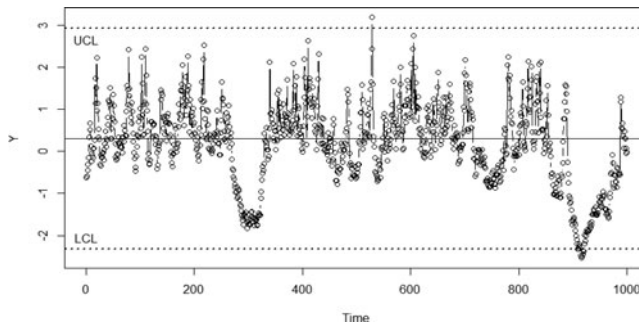


Figure 1. The plot of $\{Y_t; t = 1, \dots, 1,000\}$ under the Clayton copula with $\alpha = 8$ and the marginal (stationary) distributions $G_1 = G_2 = N(0, 1)$.

Darsow et al. (1992) first introduced copula-based Markov chain models for serially correlated observations $\{Y_t : t = 1, \dots, n\}$, where Eq. (1) defines the bivariate distribution function between Y_{t-1} and Y_t . The resultant series become a stationary process with a stationary distribution $G_1 = G_2$ (Chen and Fan, 2006; Joe, 1997). The copula-based time series models have been extensively used for financial time series models to capture temporal dependence (see Domma et al., 2009, and references therein). Such models have potential applications in many different fields; see Abegaz and Naik-Nimbalkar (2008) for some interesting application to coffee beans price.

The simplest example is the one-parameter Clayton copula defined as

$$C(u_1, u_2; \alpha) = (u_1^{-\alpha} + u_2^{-\alpha} - 1)^{-1/\alpha} \mathbf{I}(u_1^{-\alpha} + u_2^{-\alpha} - 1 > 0),$$

where $\alpha \in (-1, \infty) \setminus \{0\}$ describes the correlation between Y_{t-1} and Y_t , and $\mathbf{I}(\cdot)$ is the indicator function. It is known that the correlation between Y_{t-1} and Y_t is represented by Kendall's tau $\tau = \alpha / (\alpha + 2)$. Figure 1 shows the plot of the first-order Markov series $\{Y_t : t = 1, \dots, n\}$ under the Clayton copula with $\alpha = 8$, which corresponds to strong serial correlation ($\tau = 0.8$). Here, the stationary distribution is $G_1 = G_2 = N(0, 1)$.

The plot in Fig. 1 is drawn using R `Clayton.Markov.DATA` and `Clayton.Markov.MLE` functions that will be proposed in this article (see Appendices A and B for the definitions). After loading these R routines, one can easily obtain the plot by typing:

```
set.seed(1)
Y = Clayton.Markov.DATA(n = 1000, mu = 0, sigma = 1, alpha = 8)
Clayton.Markov.MLE(Y, plot = TRUE)
```

Similar routines (`Joe.Markov.DATA` and `Joe.Markov.MLE`) are developed under the Joe copula.

Once the time series data are given, the next step is to perform estimation. One of the most popular approaches is the parametric maximum likelihood estimator (Joe, 1997), which is shown to be particularly suitable for SPC (Long and Emura, 2014). In the literature, there seem no fully automated routines to calculate the maximum likelihood estimator (MLE) under the copula-based Markov chain models. Accordingly, Long and Emura (2014) developed the exact expressions of the Newton–Raphson algorithm for calculating the MLE under the Clayton copula. However, the implementation requires some programming effort of manually inputting the complicated expressions of the score vector and observed Fisher

information matrix. In addition, the choice of starting value and its appropriate randomization is required so that the algorithm can converge robustly. This article provides a set of R routines for automatically implementing the Newton–Raphson algorithms as well as other relevant analyses under both the Clayton and Joe copulas. Our routines are particularly useful for SPC work in which visualization of the time series and the estimation results are essential. We made our routines available as R `Copula.Markov` package (Emura and Long, 2015) in CRAN (The Comprehensive R Archive Network).

2. Parameter estimation and control chart

The transition probability density (conditional density) of Y_t given Y_{t-1} is

$$g(y_t)c(G(y_{t-1}), G(y_t); \alpha),$$

where $c(\cdot, \cdot; \alpha)$ is the copula density of $C(\cdot, \cdot; \alpha)$, and $g(\cdot)$ is the density of the true marginal (stationary) distribution $G(\cdot)$ (Chen and Fan, 2006; Joe, 1997). The parameter α describes the correlation between Y_{t-1} and Y_t . Under the first-order (Markov) model, the product of the transition densities for $t = 1, \dots, n$ yields the likelihood.

We demonstrate how the maximum likelihood estimator (MLE) of Joe (Joe, 1997) is computed. We parameterize G in terms of (μ, σ) by setting $G(y) = \Phi\{(y - \mu)/\sigma\}$, where $\mu = E(Y_t)$, $\sigma = \sqrt{\text{var}(Y_t)}$, and Φ is the distribution function of $N(0, 1)$. The log-likelihood function for data $\{Y_t : t = 1, \dots, n\}$ is given by

$$\begin{aligned} L(\mu, \sigma, \alpha) = & \frac{1}{n} \sum_{t=1}^n \log \left\{ \frac{1}{\sigma} \varphi \left(\frac{Y_t - \mu}{\sigma} \right) \right\} \\ & + \frac{1}{n} \sum_{t=2}^n \log c \left\{ \Phi \left(\frac{Y_{t-1} - \mu}{\sigma} \right), \Phi \left(\frac{Y_t - \mu}{\sigma} \right); \alpha \right\}. \end{aligned}$$

For any chosen copula, the MLE that maximizes the preceding formula is denoted by $(\hat{\mu}, \hat{\sigma}, \hat{\alpha})$. This article focuses on the Clayton copula and Joe copula for a dependence model, but the MLE can work for other well-known copulas, including Gaussian, Frank, and Gumbel copulas.

Under the Clayton copula, the log-copula density is

$$\begin{aligned} & \log c(u_1, u_2; \alpha) \\ & = \log(1 + \alpha) - (1 + \alpha) \log u_1 - (1 + \alpha) \log u_2 - \left(\frac{1}{\alpha} + 2 \right) \log(u_1^{-\alpha} + u_2^{-\alpha} - 1). \end{aligned}$$

The Clayton copula is well-known for the mathematical simplicity and remarkable popularity in applications. Some recent applications of the Clayton copula are referred to Chen and Fan (2006) for financial econometrics, Sari et al. (2009) for reliability analysis, de Uña-Álvarez and Veraverbeke (2013) for clinical medicine, and Emura and Chen (2014) and Emura et al. (2015) for bioinformatics.

The Joe copula is one of the most popular Archimedean copulas, which is originally proposed by Joe (1993). It has the log-copula density

$$\begin{aligned} l(u_1, u_2, \alpha) = & \log\{\alpha - 1 + A(u_1, u_2, \alpha)\} + (\alpha - 1) \log(1 - u_1) + (\alpha - 1) \log(1 - u_2) \\ & + \left(\frac{1}{\alpha} - 2 \right) \log A(u_1, u_2, \alpha), \quad \alpha \geq 1, \end{aligned}$$

where $A(u_1, u_2, \alpha) = (1 - u_1)^\alpha + (1 - u_2)^\alpha - (1 - u_1)^\alpha(1 - u_2)^\alpha$. The Clayton copula has the lower tail dependence while the Joe copula has the upper tail dependence. Hence, these two copulas capture quite different dependence structures and supplement each other in statistical modeling.

Numerical maximization of the log-likelihood is performed by using the Newton–Raphson algorithm that utilizes the first and second derivatives of the log-likelihood. However, even in the case of the Clayton copula, the derivative expressions of the log-likelihood can be very complicated; see the exact expressions in Long and Emura (2014), and the R coding in Appendix B. In the case of the Joe copula, the derivative expressions are formidably complicated (see Appendix C). Such expressions are not simply deduced from the derivative expressions for copulas obtained by Schepsmeier and Stöber (2014). Therefore, the development of the automatic Newton–Raphson algorithm will facilitate the calculation of the MLE.

In Appendix B, we provide R `Clayton.Markov.MLE` function to find the estimator $(\hat{\mu}, \hat{\sigma}, \hat{\alpha})$ by the so-called *randomized* Newton–Raphson algorithm (RNR) as previously proposed by Long and Emura (2014); see also Hu and Emura (2015) for the application of the RNR to other statistical models. Note that our codes in the algorithm do not use any existing optimization routine such as R `nlm`. They process the calculations through the exact derivative expressions of the log-likelihood. Hence, the codes can be easily translated into any primitive language such as C. Our routine automatically chooses the data-driven initial values to start the Newton–Raphson iterations. If the algorithm diverges, then it restarts after adequate randomization in the initial values. They are completely automated and independent algorithms that can work under any computational environment.

The following example calculates the MLE:

```
set.seed(1)
Y = Clayton.Markov.DATA(n = 1000,mu = 0,sigma = 1,alpha = 8)
Clayton.Markov.MLE(Y,plot = TRUE)
$estimates
mu sigma alpha UCL LCL
0.3052139 0.8740975 5.1890571 2.9275065 -2.3170787
$out_of_control
[1] 529 909 910 914 915 916 917 918 919 920
$Gradient
[1] 2.016333e-11 -4.821663e-11 7.171406e-12
$Hessian
[1,] [2,] [3,]
[1,] -0.7938720 0.8135396 -0.14625514
[2,] 0.8135396 -2.9587007 0.26846836
[3,] -0.1462551 0.2684684 -0.03458822
$Mineigenvalue_Hessian
[1] -3.258449
```

Here, we use the same data $\{Y_i:t = 1, \dots, 1,000\}$ as appeared in Fig. 1. In the output, “\$estimates” gives the MLE $(\hat{\mu}, \hat{\sigma}, \hat{\alpha})$, the lower control limit ($LCL = \hat{\mu} - 3\hat{\sigma}$) and the upper control limit ($UCL = \hat{\mu} + 3\hat{\sigma}$). Whether the MLE attains the maximum of the likelihood function or not can be confirmed by checking “\$Gradient” and “\$Mineigenvalue_Hessian.” In the example, the gradients are nearly equal to zero, which means that the likelihood function gives a proper solution. In addition, the output shows that the Hessian is negative definite

since the minimum eigenvalue of the Hessian matrix is negative. This guarantees that the MLE attains the local maximum of the log-likelihood (see p. 284, Theorem 7.7.1 of Khuri, 2003).

Even though $n = 1,000$ is quite large, the MLE $\hat{\mu} = 0.3052139$ and $\hat{\sigma} = 0.8740975$ are somewhat biased for the true value $\mu = 0$ and $\sigma = 1$. This bias is due to the large sampling variation caused by the strong serial correlation ($\tau = 0.8$). This phenomenon is expected from the highly unstable time series plot in Fig. 1. However, in the long run, the bias vanishes.

Note that the MLE $(\hat{\mu}, \hat{\sigma})$ are different from the sample mean (\bar{Y}) and sample standard deviation (S_Y) of $\{Y_t: t = 1, \dots, 1,000\}$. While \bar{Y} and S_Y are consistent estimators of the true (μ, σ) , they have larger mean square error than the MLE (Long and Emura, 2014). Hence, \bar{Y} and S_Y are not recommended under serially correlated observation.

In addition to the MLE, the `Clayton.Markov.MLE` also draws a Shewhart-type control chart, including $UCL = \hat{\mu} + k\hat{\sigma}$, $LCL = \hat{\mu} - k\hat{\sigma}$, and the center line $\hat{\mu}$ (Fig. 1). The default is $k = 3$ (3-sigma control limit), but the user can specify any value $k > 0$. In the output, 10 observations fall outside the interval $[LCL, UCL]$, which are called out-of-control signals. These 10 out-of-control signals are indeed identified from Fig. 1. In long-run, the rate of out-of-control signals is specified at 0.27% as in the usual 3-sigma control limits.

The implementation under the Joe copula is possible by the `Joe.Markov.MLE`, which is available in R `Copula.Markov` package. Appendix C gives the exact expressions of the first and second derivatives of the log-likelihood to perform the Newton–Raphson iterations. The expressions are fairly complicated but still work properly for purpose of maximizing the log-likelihood.

3. Data analysis

3.1. Chemical process data

We analyze the chemical process data (Box and Jenkins, 1976) for illustration. We use the same data that are conveniently prepared in the appendix of Bisgaard and Kulahci (2007). The data consist of a series of chemical concentrations $\{Y_t : t = 1, \dots, 197\}$ measured every 2 hours. Chemical engineers may need an SPC procedure to judge if the concentration level is kept within a reasonable range. The inputs and outputs of applying the `Clayton.Markov.MLE` function are given below:

```
Y = c(17.0, 16.6, 16.3, 16.1, 17.1, 16.9, 16.8, 17.4, 17.1, 17.0, 16.7,
17.4, 17.2, 17.4, 17.4, 17.0, 17.3, 17.2, 17.4, 16.8, 17.1, 17.4,
17.4, 17.5, 17.4, 17.6, 17.4, 17.3, 17.0, 17.8, 17.5, 18.1, 17.5,
17.4, 17.4, 17.1, 17.6, 17.7, 17.4, 17.8, 17.6, 17.5, 16.5, 17.8,
17.3, 17.3, 17.1, 17.4, 16.9, 17.3, 17.6, 16.9, 16.7, 16.8, 16.8,
17.2, 16.8, 17.6, 17.2, 16.6, 17.1, 16.9, 16.6, 18.0, 17.2, 17.3,
17.0, 16.9, 17.3, 16.8, 17.3, 17.4, 17.7, 16.8, 16.9, 17.0, 16.9,
17.0, 16.6, 16.7, 16.8, 16.7, 16.4, 16.5, 16.4, 16.6, 16.5, 16.7,
16.4, 16.4, 16.2, 16.4, 16.3, 16.4, 17.0, 16.9, 17.1, 17.1, 16.7,
16.9, 16.5, 17.2, 16.4, 17.0, 17.0, 16.7, 16.2, 16.6, 16.9, 16.5,
16.6, 16.6, 17.0, 17.1, 17.1, 16.7, 16.8, 16.3, 16.6, 16.8, 16.9,
17.1, 16.8, 17.0, 17.2, 17.3, 17.2, 17.3, 17.2, 17.2, 17.5, 16.9,
16.9, 16.9, 17.0, 16.5, 16.7, 16.8, 16.7, 16.7, 16.6, 16.5, 17.0,
16.7, 16.7, 16.9, 17.4, 17.1, 17.0, 16.8, 17.2, 17.2, 17.4, 17.2,
16.9, 16.8, 17.0, 17.4, 17.2, 17.2, 17.1, 17.1, 17.1, 17.4, 17.2,
16.9, 16.9, 17.0, 16.7, 16.9, 17.3, 17.8, 17.8, 17.6, 17.5, 17.0,
```

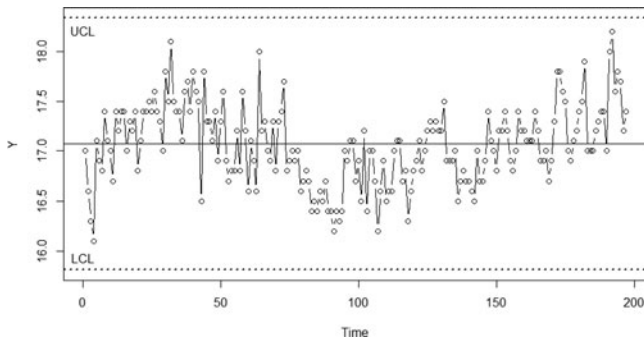


Figure 2. A plot of chemical concentrations $\{Y_t : t = 1, \dots, 197\}$ measured every 2 hours (Bisgaard and Kulahci, 2007; Box and Jenkins, 1976).

```
16.9, 17.1, 17.2, 17.4, 17.5, 17.9, 17.0, 17.0, 17.0, 17.2, 17.3,
17.4, 17.4, 17.0, 18.0, 18.2, 17.6, 17.8, 17.7, 17.2, 17.4)
```

```
>
```

```
Clayton.Markov.MLE(Y)
```

```
$estimates
```

```
mu sigma alpha UCL LCL
```

```
17.0732223 0.4213754 1.1777489 18.3373486 15.8090961
```

```
$out_of_control
```

```
[1] "NONE"
```

```
$Gradient
```

```
[1] -1.718471e-10 -5.056611e-10 3.708906e-10
```

```
$Hessian
```

```
[,1] [,2] [,3]
```

```
[1,] -2.5717301 0.5930541 -0.3865827
```

```
[2,] 0.5930541 -12.7133719 1.2185907
```

```
[3,] -0.3865827 1.2185907 -0.2155532
```

```
$Mineigenvalue_Hessian
```

```
[1] -12.86935
```

The outputs show $\hat{\mu} = 17.0732223$ and $\hat{\sigma} = 0.4213754$. The value $\hat{\mu} = 17.0732223$ is indeed the center of the time series plot (Fig. 2). The outputs also include $LCL = \hat{\mu} - 3\hat{\sigma} = 15.8090961$ and $UCL = \hat{\mu} + 3\hat{\sigma} = 18.3373486$. Figure 2 shows that all the points are between the LCL and UCL, which implies that the process is in-control. This is confirmed by "NONE" in \$out_of_control. The data exhibit positive serial correlation $\hat{\alpha} = 1.1777489$ with the corresponding Kendall's tau being $\hat{\tau} = \hat{\alpha}/(\hat{\alpha} + 2) = 0.37$. Such a serial correlation may be due to the short intervals (2 hours) between consecutive observations.

3.2. Financial data

We analyze the weekly returns of S&P 500 index consisting of 500 leading companies in leading industries of the U.S. economy. Data are loaded from FRED (Federal Reserve Economic Data) <https://research.stlouisfed.org/fred2/series/SP500/downloaddata> that is a data bank for economic time series. We extract weekly data from January 1, 2010 to January 3, 2014 (weekly, ending Friday) and write them as $\{Y_t : t = 1, \dots, 210\}$. The goal is to show that the weekly

returns stay within a reasonable range in order to verify some financial assumptions. We apply Clayton . Markov . MLE function and obtain the inputs and outputs below:

```

Y = c(-11.38, 29.88, -8.95, -44.27, -17.89, -7.68, 9.32, 33.66, -4.68, 34.21, 11.29, 9.91,
      6.69, 11.51,
      6.27, -2.24, 25.15, -30.59, -75.81, 24.80, -47.99, 1.72, -24.53, 26.72, 25.91, -40.75, -54.18,
      55.38,
      -13.08, 37.78, -1.06, 20.04, -42.39, -7.56, -7.10, 39.92, 5.04, 16.04, 23.08, -2.43, 18.91,
      11.04,
      6.89, 0.18, 42.59, -26.64, 0.52, -10.33, 35.31, 15.69, 3.51, 12.86, 0.87, 13.86, 21.74, -9.89,
      -7.01, 34.53, 18.28, 13.86, -23.13, 1.27, -16.87, -25.08, 34.60, 18.61, -4.24, -8.49, 17.70,
      26.23,
      -23.41, -2.43, -4.50, -2.17, -30.94, -29.18, 0.52, -3.05, 71.22, 4.13, -27.66, 28.88,
      -52.74, -92.09,
      -20.57, -55.28, 53.25, -2.83, -19.74, 61.78, -79.58, -5.01, 24.04, 69.12, 13.67, 46.84,
      -31.86, 10.62,
      -48.20, -56.98, 85.61, 10.91, -35.53, 45.67, -7.73, 20.21, 11.28, 26.29, 0.95, 28.57, -2.26,
      18.59,
      4.51, 3.89, 1.24, 33.30, -7.06, 11.36, -10.39, -27.82, 8.27, 24.83, -34.26, -15.71, -58.17,
      22.60,
      -39.78, 47.62, 17.18, -7.82, 27.14, -7.48, 2.10, 5.88, 23.31, 5.02, 14.88, 12.29, -7.03, -4.55,
      31.34, 27.85, -5.62, -19.48, 20.26, -32.34, 4.60, -21.25, 2.26, -34.35, -19.97, 49.27, 7.03,
      1.89,
      -4.49, 16.57, -27.72, 64.04, 5.58, 13.93, 16.98, 10.21, 4.76, 1.86, -4.19, 2.60, 32.98, 9.52,
      -3.81, 12.30, -15.91, 35.57, -33.60, 26.99, 32.18, 19.28, 33.77, -17.87, -18.86, 12.64,
      -16.65, -34.30,
      13.85, 25.61, 48.30, 11.90, -0.44, 18.02, -18.2, -35.59, 7.67, -30.53, 22.20, 32.82, 21.92,
      -18.16,
      -1.25, 12.70, 41.30, 15.27, 1.87, 8.97, 27.57, 6.58, 1.05, -0.72, -29.77, 43.00, 23.08, -10.03)
Clayton.Markov.MLE(Y)
$estimates
mu sigma alpha UCL LCL
3.32651318 27.47944488 0.04474872 85.76484782 -79.11182147
$out_of_control
[1] 84 91
$Gradient
[1] -6.484562e-18 -2.973812e-19 -9.537344e-15
$Hessian
[,1] [,2] [,3]
[1,] -1.244355e-03 -5.584538e-05 -0.003685264
[2,] -5.584538e-05 -2.599757e-03 0.001911515
[3,] -3.685264e-03 1.911515e-03 -1.764584035
$Mineigenvalue_Hessian
[1] -1.764594

```

We obtain the estimates $\hat{\mu} = 3.32651318$ and $\hat{\sigma} = 27.47944488$. Indeed, the three horizontal lines in the time series plot (Fig. 3) are the center $\hat{\mu} = 3.32651318$, LCL = $\hat{\mu} - 3\hat{\sigma} = -79.11182147$ and UCL = $\hat{\mu} + 3\hat{\sigma} = 85.76484782$. We observe that there are only two out-of-control points (84 and 91) as shown in \$out_of_control. In addition, the dependence

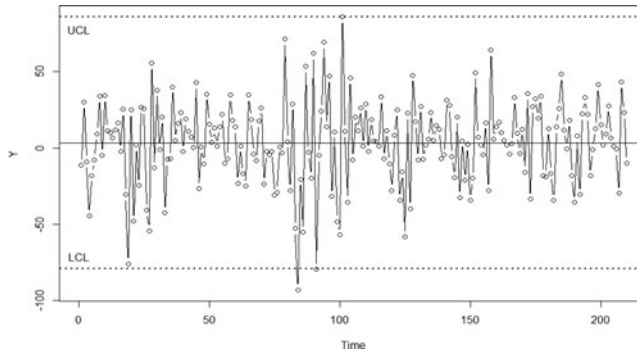


Figure 3. A plot of the weekly returns from January 1, 2010 to January 3, 2014 $\{Y_t : t = 1, \dots, 210\}$.

parameter is $\hat{\alpha} = 0.04474872$ with the corresponding Kendall's tau being $\hat{\tau} = \hat{\alpha}/(\hat{\alpha} + 2) = 0.021$. These two values are both very close zero, which implies that the returns are nearly independent between the successive weeks. Hence, the observed out-of-control signal would not affect the next week's return of S&P 500. We can conclude that the weekly returns for S&P500 stay nearly in-control, which will be useful information for risk aversion investors.

4. Conclusion and future work

In this article, we introduce a set of R routines that automatically generate time series data, perform maximum likelihood estimation, and draw the control chart under copula-based Markov chain models. As demonstrated through the data analysis, they can be performed by simply inputting only a few command lines in R. Our coding is completely original and primitive, which mean that we do not use any existing sub-routine. Hence, the present codes written in R language can be easily translated or extended into any primitive language. In addition, the outputs of our routines are well-annotated that allow researchers to check the convergence of the algorithms. We believe that our routines can greatly facilitate statistical inference and statistical process control under copula-based Markov models.

While the present `Clayton.Markov.DATA` or `Joe.Markov.DATA` can only generate data from the normal stationary distribution, it can be easily generalized to generate from other distributions by transformation. For instance, the original series $\{Y_t : t = 1, \dots, n\}$ can yield heavy-tailed data $\{Y_t^* : t = 1, \dots, n\}$ using the t -transformation

$$Y_t^* = \mu + \sigma \sqrt{\frac{\nu - 2}{\nu}} \Psi^{-1}[\Phi\{(Y_t - \mu)/\sigma\}; \nu],$$

where $\Psi^{-1}[\cdot; \nu]$ is the quantile function of the t -distribution with degree of freedom ν .

Although we develop a set of R functions applied a single copula-generated time series data, many manufacturing process may encounter multiple series of correlated data. For instance, engineers often wish to monitor two quality characteristics $\{X_t : t = 1, \dots, n\}$ and $\{Y_t : t = 1, \dots, n\}$ simultaneously. The most popular way to simultaneously monitor the two series is a control ellipse or Hotelling T^2 -chart (chap. 11 of Montgomery, 2009). These approaches must be modified to take into account serial correlation in the two series. For instance, the input temperature X_t is clearly related to the output thickness Y_t (Tseng et al., 2007). Here, simultaneous control for the two variables is essential. Clearly copulas are not only useful for serial correlation, but also for the correlation between series; see recent articles Brechmann and Cazo (2013) and Huh (2014). In the presence of two series, one is not only interested in

monitoring the process mean, but also the causal relationship between the two series. In terms of Granger causality, a way to extract the causal information from two time series is studied by Brechmann and Cazo (2013) and Hung and Tseng (2013).

Author contribution

- Emura T: Designed and supervised the study. Wrote and revised the draft. Derived mathematical expressions. Made the R package.
- Long TH: Made part of R routines. Wrote part of the draft.
- Sun LH: Collected and analyzed the financial data (Section 3.2).

Appendix A: R functions Clayton.Markov.DATA and Joe.Markov.DATA

• Description

The R function Clayton.Markov.DATA (or Joe.Markov.DATA) generates the datasets under a copula-based Markov chain model. The serial dependence follows the Clayton copula (or the Joe copula) and the marginal (stationary) distribution follows the normal distribution.

• Usage

Clayton.Markov.DATA(n,mu,sigma,alpha)

Joe.Markov.DATA(n,mu,sigma,alpha)

• Arguments

n: sample size

mu: mean

sigma: standard deviation ($\sigma > 0$)

alpha: association parameter

(Clayton: $-1 < \alpha < 0$ for negative association; $\alpha > 0$ for positive association)

(Joe: $\alpha > = 1$ for positive association)

• Definition

```
#####
Clayton.Markov.DATA = function(n,mu,sigma,alpha){
  Y = numeric(n)
  Y[1] = rnorm(1,mu,sigma)
  for(i in 2:n){
    U1 = pnorm(Y[i-1],mu,sigma)
    Y[i] = qnorm((1+(runif(1)^(-alpha/(alpha+1))-1)*(U1 ^(-alpha)))^(-1/alpha),mu,sigma)
  }
  return(Y)
}
#####
Joe.Markov.DATA = function(n,mu,sigma,alpha){
  Y = numeric(n)
  Y[1] = rnorm(1,mu,sigma)
  for(i in 2:n){
    U1 = pnorm(Y[i-1],mu,sigma)
    V = runif(1)
    Joe_func = function(u2){
      A = (1-U1)^alpha+(1-u2)^alpha-((1-U1)^alpha)*((1-u2)^alpha)
```

```

V-A^(1/alpha-1)*(1-(1-u2)^alpha)*(1-U1)^(alpha-1)
}
U2 = uniroot(Joe_func,interval = c(0.0000001,0.99999))$root
Y[i] = qnorm(U2,mu,sigma)
}
return(Y)
}

```

Appendix B: R function `Copula.Markov.MLE`

• Description

The R function `Clayton.Markov.MLE` produces the maximum likelihood estimates and draws the Shewhart chart with k -sigma control limits (e.g., 3-sigma). The dependence model follows the Clayton copula and the marginal (stationary) distribution follows the normal distribution

• Usage

```
Clayton.Markov.MLE(Y,k = 3,plot = TRUE)
```

• Arguments

Y: vector of datasets

k: constant determining the length between LCL and UCL ($k = 3$ corresponds to 3-sigma limit)

D: diameter for $U(-D, D)$ used in randomized Newton–Raphson

• Values

\$estimates: Estimates of $(\hat{\mu}, \hat{\sigma}, \hat{\alpha})$, UCL and LCL.

\$out_of_control: Indices for out-of-control signals.

\$Gradient: The gradient of the log-likelihood at the solution. They should be close to zero.

\$Hessian: The Hessian matrix of the log-likelihood at the solution.

\$Mineigenvalue_Hessian: The minimum eigenvalue of the Hessian matrix.

• Definition

```

#####
Clayton.Markov.MLE = function(Y,k = 3,D = 1,plot = TRUE){
n = length(Y) ##sample size##
G = function(y,mu,sigma){pnorm((y-mu)/sigma)} #G function
g = function(y,mu,sigma){dnorm((y-mu)/sigma)} #g function
#####Log-likelihood function #####
L_function = function(mu,sigma,alpha){
U_t_1 = G(Y[1:n-1],mu,sigma);U_t = G(Y[2:n],mu,sigma)
u_t_1 = g(Y[1:n-1],mu,sigma);u_t = g(Y[2:n],mu,sigma)
A = U_t_1^(-alpha)+U_t^(-alpha)-1
Z = log(1+alpha)-(1+alpha)*log(U_t_1)-(1+alpha)*log(U_t)-(1/alpha+2)*log(A)
ZZ = log(g(Y[1:n],mu,sigma)/sigma)
return((sum(Z)+sum(ZZ))/n)
}
#####dL/dmu#####
dL_dmu = function(mu,sigma,alpha){
U_t_1 = G(Y[1:n-1],mu,sigma);U_t = G(Y[2:n],mu,sigma)
u_t_1 = g(Y[1:n-1],mu,sigma);u_t = g(Y[2:n],mu,sigma)
A1 = u_t_1/U_t_1+u_t/U_t

```

```

A2 = (u_t_1*U_t_1^(-(alpha+1))+u_t*U_t^(-(alpha+1)))/(U_t_1^(-alpha)
      +U_t^(-alpha)-1)
A = (alpha+1)/sigma*A1-(2*alpha+1)/sigma*A2
A3 = (Y[1:n]-mu)/sigma^2
return((sum(A)+sum(A3))/n)
}
#####dL/dsigma#####
dL_dsigma = function(mu,sigma,alpha){
  U_t_1 = G(Y[1:n-1],mu,sigma);U_t = G(Y[2:n],mu,sigma)
  u_t_1 = g(Y[1:n-1],mu,sigma);u_t = g(Y[2:n],mu,sigma)
  B1 = (Y[1:n-1]-mu)/sigma^2*u_t_1/U_t_1
  B2 = (Y[2:n]-mu)/sigma^2*u_t/U_t
  B3 = (Y[1:n-1]-mu)/sigma^2*U_t_1^(-(alpha+1))*u_t_1
  B4 = (Y[2:n]-mu)/sigma^2*U_t^(-(alpha+1))*u_t
  B5 = (U_t_1^(-alpha)+U_t^(-alpha)-1)
  B = (alpha+1)*B1+(alpha+1)*B2-(2*alpha+1)*(B3+B4/B5
  B6 = (Y[1:n]-mu)^2/sigma^3-1/sigma
  return((sum(B)+sum(B6))/n)
}
#####dL/dalpha#####
dL_dalpha = function(mu,sigma,alpha){
  U_t_1 = G(Y[1:n-1],mu,sigma);U_t = G(Y[2:n],mu,sigma)
  u_t_1 = g(Y[1:n-1],mu,sigma);u_t = g(Y[2:n],mu,sigma)
  E1 = log(U_t_1*U_t)
  E2 = log(U_t_1^(-alpha)+U_t^(-alpha)-1)/alpha^2
  E3 = U_t_1^(-alpha)*log(U_t_1)+U_t^(-alpha)*log(U_t)
  E5 = (U_t_1^(-alpha)+U_t^(-alpha)-1)
  E = 1/(1+alpha)-E1+E2+(2+1/alpha)*E3/E5
  return(sum(E)/n)
}
#####F function#####
F = function(mu,sigma,alpha){
  c(dL_dmu(mu,sigma,alpha),dL_dsigma(mu,sigma,alpha),dL_dalpha(mu,sigma,alpha))
}
#####d^2L/dmu^2#####
d2L_dmu2 = function(mu,sigma,alpha){
  U_t_1 = G(Y[1:n-1],mu,sigma);U_t = G(Y[2:n],mu,sigma)
  u_t_1 = g(Y[1:n-1],mu,sigma);u_t = g(Y[2:n],mu,sigma)
  H1 = ((Y[1:n-1]-mu)/sigma^2*u_t_1*U_t_1+u_t_1^2/sigma)/U_t_1^2
  H2 = ((Y[2:n]-mu)/sigma^2*u_t*U_t+u_t^2/sigma)/U_t^2
  H3 = U_t_1^(-(2+alpha))*u_t_1^2
  H4 = (Y[1:n-1]-mu)/sigma^2*u_t_1*U_t_1^(-(1+alpha))
  H5 = U_t^(-(2+alpha))*u_t^2
  H6 = (Y[2:n]-mu)/sigma^2*u_t*U_t^(-(1+alpha))
  H7 = (U_t_1^(-alpha)+U_t^(-alpha)-1)
  H8 = U_t_1^(-(1+alpha))*u_t_1+U_t^(-(1+alpha))*u_t
  H = (alpha+1)/sigma*(H1+H2)-(2*alpha+1)/sigma*(((alpha+1)/sigma*H3+H4
      +(alpha+1)/sigma*H5+H6)*H7-alpha/sigma*H8^2)/H7^2
}

```

```

return((sum(H)+(-n/sigma^2))/n)
}
#####d^2L/dsigma^2#####
J = J1 = J2 = J3 = J4 = J5 = J6 = J7 = J8 = J9 = J10 = J11 = J12 = J13 = J14 = J15
    = J16 = c()
d2L_dsigma2 = function(mu,sigma,alpha){
U_t_1 = G(Y[1:n-1],mu,sigma);U_t = G(Y[2:n],mu,sigma)
u_t_1 = g(Y[1:n-1],mu,sigma);u_t = g(Y[2:n],mu,sigma)
J1 = (Y[1:n-1]-mu)/sigma^3*u_t_1/U_t_1
J2 = (Y[1:n-1]-mu)^2/sigma^2+(Y[1:n-1]-mu)/sigma*u_t_1/U_t_1
J4 = (Y[2:n]-mu)/sigma^3*u_t/U_t
J5 = (Y[2:n]-mu)^2/sigma^2+(Y[2:n]-mu)/sigma*u_t/U_t
J7 = (Y[1:n-1]-mu)/sigma^3*U_t_1^(-(alpha+1))*u_t_1
J8 = (Y[1:n-1]-mu)/sigma*u_t_1/U_t_1
J9 = (Y[1:n-1]-mu)^2/sigma^2
J10 = (Y[2:n]-mu)/sigma^3*U_t^(-(alpha+1))*u_t
J11 = (Y[2:n]-mu)/sigma*u_t/U_t
J12 = (Y[2:n]-mu)^2/sigma^2
J13 = (U_t_1^(-alpha)+U_t^(-alpha)-1)
J15 = (Y[1:n-1]-mu)/sigma^2*U_t_1^(-(alpha+1))*u_t_1
J16 = (Y[2:n]-mu)/sigma^2*U_t^(-(alpha+1))*u_t
J = (alpha+1)*(J1*(-2+J2)+J4*(-2+J5))-(2*alpha+1)*((J7*(-2+(alpha+1)*J8+J9)+J10
    *(-2+(alpha+1)*J11+J12))*J13-alpha*(J15+J16)^2)/J13^2
J14 = -3*(Y[1:n]-mu)^2/sigma^4+1/sigma^2
return((sum(J)+sum(J14))/n)
}
#####d^2L/dalpha^2#####
d2L_dalpha2 = function(mu,sigma,alpha){
U_t_1 = G(Y[1:n-1],mu,sigma);U_t = G(Y[2:n],mu,sigma)
u_t_1 = g(Y[1:n-1],mu,sigma);u_t = g(Y[2:n],mu,sigma)
K1 = (U_t_1^(-alpha)+U_t^(-alpha)-1)
K2 = U_t_1^(-alpha)*log(U_t_1)+U_t^(-alpha)*log(U_t)
K4 = U_t_1^(-alpha)*log(U_t_1)^2+U_t^(-alpha)*log(U_t)^2
K = -1/(1+alpha)^2-2/alpha^3*log(K1)-2/alpha^2*K2/K1+(1/alpha+2)
    *(K2^2/K1^2-K4/K1)
return(sum(K)/n)
}
#####d^2L/dmudsigma#####
d2L_dmudsigma = function(mu,sigma,alpha){
U_t_1 = G(Y[1:n-1],mu,sigma);U_t = G(Y[2:n],mu,sigma)
u_t_1 = g(Y[1:n-1],mu,sigma);u_t = g(Y[2:n],mu,sigma)
L1 = -(1+alpha)/sigma^2*(u_t_1/U_t_1+u_t/U_t)
L2 = ((Y[1:n-1]-mu)^2/sigma^3*u_t_1*U_t_1+(Y[1:n-1]-mu)/sigma^2*
    (u_t_1)^2)/U_t_1^2
L3 = ((Y[2:n]-mu)^2/sigma^3*u_t*U_t+(Y[2:n]-mu)/sigma^2*(u_t)^2)/U_t^2
L4 = (U_t_1^(-1-alpha)*u_t_1+U_t^(-1-alpha)*u_t)/(U_t_1^(-alpha)+U_t^(-alpha)-1)
L5 = 1/(U_t_1^(-alpha)+U_t^(-alpha)-1)^2
L6 = (alpha+1)*U_t_1^(-alpha-2)*(Y[1:n-1]-mu)/sigma^2*u_t_1^2

```

```

L7 = U_t_1^(-alpha-1)*u_t_1*(Y[1:n-1]-mu)^2/sigma^3
L8 = (alpha+1)*U_t^(-alpha-2)*(Y[2:n]-mu)/sigma^2*u_t^2
L9 = U_t^(-alpha-1)*u_t*(Y[2:n]-mu)^2/sigma^3
L10 = (U_t_1^(-alpha)+U_t^(-alpha)-1)
L11 = U_t_1^(-1-alpha)*u_t_1+U_t^(-1-alpha)*u_t
L12 = (Y[1:n-1]-mu)/sigma^2*U_t_1^(-1-alpha)*u_t_1+(Y[2:n]-mu)/sigma^2*
      U_t^(-1-alpha)*u_t
L = L1+(alpha+1)/sigma*(L2+L3)+(2*alpha+1)/sigma^2*L4-(2*alpha+1)/sigma*L5
      *((L6+L7+L8+L9)*L10-alpha*L11*L12)
LL = -2*(Y[1:n]-mu)/sigma^3
return((sum(L)+sum(LL))/n)
}

#####d^2L/dmudalpha#####
d2L_dmudalpha = function(mu,sigma,alpha){
  U_t_1 = G(Y[1:n-1],mu,sigma);U_t = G(Y[2:n],mu,sigma)
  u_t_1 = g(Y[1:n-1],mu,sigma);u_t = g(Y[2:n],mu,sigma)
  M1 = (u_t_1/U_t_1+u_t/U_t)/sigma
  M2 = (U_t_1^(-(alpha+1))*u_t_1+U_t^(-(alpha+1))*u_t)/(U_t_1^(-alpha)
        +U_t^(-alpha)-1)^2/sigma
  M3 = -U_t_1^(-(1+alpha))*log(U_t_1)*u_t_1-U_t^(-(1+alpha))*log(U_t)*u_t
  M4 = (U_t_1^(-alpha)+U_t^(-alpha)-1)
  M5 = U_t_1^(-(alpha+1))*u_t_1+U_t^(-(alpha+1))*u_t
  M6 = -U_t_1^(-alpha)*log(U_t_1)-U_t^(-alpha)*log(U_t)
  M = M1-M2-(2*alpha+1)/sigma*(M3*M4-M5*M6)/M4^2
  return(sum(M)/n)
}

#####d^2L/dsigmadalpha#####
d2L_dsigmadalpha = function(mu,sigma,alpha){
  U_t_1 = G(Y[1:n-1],mu,sigma);U_t = G(Y[2:n],mu,sigma)
  u_t_1 = g(Y[1:n-1],mu,sigma);u_t = g(Y[2:n],mu,sigma)
  O1 = (Y[1:n-1]-mu)/sigma^2*u_t_1/U_t_1+(Y[2:n]-mu)//sigma^2*u_t/U_t
  O3 = (Y[1:n-1]-mu)/sigma^2*U_t_1^(-(1+alpha))*u_t_1
  O4 = (Y[2:n]-mu)/sigma^2*U_t^(-(1+alpha))*u_t
  O5 = (U_t_1^(-alpha)+U_t^(-alpha)-1)
  O6 = U_t_1^(-alpha)*log(U_t_1)+U_t^(-alpha)*log(U_t)
  O = O1-2*(O3+O4)/O5-(2*alpha+1)*((-log(U_t_1)*O3-log(U_t)*O4)*O5
      +(O3+O4)*O6)/O5^2
  return(sum(O)/n)
}

#####Jacobian function#####
Ja = function(mu,sigma,alpha){
  AA = matrix(c(d2L_dmu2(mu,sigma,alpha),d2L_dmudsigma(mu,sigma,alpha),
               d2L_dmudalpha(mu,sigma,alpha),d2L_dmudsigma(mu,sigma,alpha),
               d2L_dsigma2(mu,sigma,alpha),d2L_dsigmadalpha(mu,sigma,alpha),
               d2L_dmudalpha(mu,sigma,alpha),d2L_dsigmadalpha(mu,sigma,alpha),
               d2L_dalpha2(mu,sigma,alpha)),3,3)
  return(AA)
}

```

```
#####Multivariate Newton Raphson#####
X = matrix(1,3)
tau = cor(Y[1:n-1],Y[2:n],method = "kendall")
alpha_est = -2*tau/(tau-1)
X[1,] = c(mean(Y),sd(Y),alpha_est) #initial value
i = 2
Ran.num = 1
repeat{
Z = X
X = matrix(i,3)
X[1:i-1,] = Z[1:i-1,]
##
Aa = Ja(X[i-1,1],X[i-1,2],X[i-1,3])
Ainv11 = Aa[2,2]*Aa[3,3]-Aa[3,2]*Aa[2,3]
Ainv12 = Aa[1,2]*Aa[3,3]-Aa[3,2]*Aa[1,3]
Ainv13 = Aa[1,2]*Aa[2,3]-Aa[2,2]*Aa[1,3]
Ainv21 = Aa[2,1]*Aa[3,3]-Aa[3,1]*Aa[2,3]
Ainv22 = Aa[1,1]*Aa[3,3]-Aa[3,1]*Aa[1,3]
Ainv23 = Aa[1,1]*Aa[2,3]-Aa[1,3]*Aa[2,1]
Ainv31 = Aa[2,1]*Aa[3,2]-Aa[3,1]*Aa[2,2]
Ainv32 = Aa[1,1]*Aa[3,2]-Aa[1,2]*Aa[3,1]
Ainv33 = Aa[1,1]*Aa[2,2]-Aa[1,2]*Aa[2,1]
Ainv = matrix(c(Ainv11,-Ainv21,Ainv31,-Ainv12,Ainv22,-Ainv32,Ainv13,-Ainv23,
Ainv33),3,3)/det(Aa)
##
X[i,] = X[i-1,]-Ainv%*%F(X[i-1,1],X[i-1,2],X[i-1,3])
if(1*is.nan(X)[i,1] == 1){
X = matrix(2,3)
X[1,] = c(mean(Y),sd(Y),alpha_est+runif(1,-D,D)) #initial value
Ran.num = Ran.num+1
i = 1
}else if(abs(X[i,1]-X[i-1,1])<0.0001&abs(X[i,2]-X[i-1,2])<0.0001&abs(X[i,3]-X[i-1,3])<0.0001&abs(X[i,3]-alpha_est)>5){
X = matrix(2,3)
X[1,] = c(mean(Y),sd(Y),alpha_est+runif(1,-D,D)) #initial value
Ran.num = Ran.num+1
i = 1
}else if(abs(X[i,1]-X[i-1,1])<0.0001&abs(X[i,2]-X[i-1,2])<0.0001&abs(X[i,3]-X[i-1,3])<0.0001&X[i,2]>0&abs(X[i,3]-alpha_est)<5){break
}else if(abs(X[i,1]-X[i-1,1])<0.0001&abs(X[i,2]-X[i-1,2])<0.0001&abs(X[i,3]-X[i-1,3])<0.0001&X[i,2]<0){
X = matrix(2,3)
X[1,] = c(mean(Y),sd(Y),alpha_est+runif(1,-D,D)) #initial value
Ran.num = Ran.num+1
i = 1
}else if(abs(X[i,1]-X[i-1,1])>10^10&abs(X[i,2]-X[i-1,2])>10^10&abs(X[i,3]-X[i-1,3])>10^10){
X = matrix(2,3)
```

```

X[1,] = c(mean(Y),sd(Y),alpha_est+runif(1,-D,D)) #initial value
Ran.num = Ran.num+1
i = 1
}
if(Ran.num>= 100){break}
i = i+1
}
mle.res = X[length(X[,1]),]
if(Ran.num>= 10){mle.res = c(mean(Y),sd(Y),alpha_est)}
UCL = mle.res[1]+k*mle.res[2]
LCL = mle.res[1]-k*mle.res[2]
result = c(mu = mle.res[1],sigma = mle.res[2],alpha = mle.res[3],UCL = UCL,LCL =
LCL)
##### Plot Control Chart #####
if(plot == TRUE){
Min = min(min(Y),LCL)
Max = max(max(Y),UCL)
ts.plot(Y,type = "b",ylab = "Y",ylim = c(Min,Max))
abline(h = result[1])
abline(h = UCL,lty = "dotted",lwd = 2)
abline(h = LCL,lty = "dotted",lwd = 2)
text(0,LCL+(result[1]-LCL)*0.1,"LCL")
text(0,UCL-(UCL-result[1])*0.1,"UCL")
}
out_control = which((Y<LCL)|(UCL<Y))
if(length(out_control) == 0){out_control = "NONE"}
Gradient = F(mle.res[1],mle.res[2],mle.res[3])
Hessian = Ja(mle.res[1],mle.res[2],mle.res[3])
return(
list(estimates = result,out_of_control = out_control,Gradient = Gradient,Hessian =
Hessian,Mineigenvalue_Hessian = min(eigen(Hessian)$value))
)
}

```

Appendix C: Details on the Joe copula

The Joe copula (Joe, 1993) is defined as

$$C_{\alpha}(u_1, u_2) = 1 - A(u_1, u_2, \alpha)^{\frac{1}{\alpha}}, \quad \alpha \geq 1,$$

where $A(u_1, u_2, \alpha) = (1 - u_1)^{\alpha} + (1 - u_2)^{\alpha} - (1 - u_1)^{\alpha}(1 - u_2)^{\alpha}$. The copula density is

$$c_{\alpha}(u_1, u_2) = A(u_1, u_2, \alpha)^{\frac{1}{\alpha}-2}(1 - u_1)^{\alpha-1}(1 - u_2)^{\alpha-1}\{(\alpha - 1) + A(u_1, u_2, \alpha)\}.$$

The log-copula density is

$$l(u_1, u_2, \alpha) = \log\{\alpha - 1 + A(u_1, u_2, \alpha)\} + (\alpha - 1)\log(1 - u_1) + (\alpha - 1)\log(1 - u_2) \\ + \left(\frac{1}{\alpha} - 2\right)\log A(u_1, u_2, \alpha).$$

One can generate data (U_1, U_2) from the Joe copula as follows:

Step1. Generate $U_1, V \sim U(0, 1)$.

Step2. Set U_2 as the solution to $V = A(U_1, U_2, \alpha)^{\frac{1}{\alpha}-1}\{1 - (1 - U_2)^\alpha\}(1 - U_1)^{\alpha-1}$.

Now we derive the first- and second-order derivatives of the log-likelihood. Let $u_t = \varphi\{(Y_t - \mu)/\sigma\}$, $U_t = \Phi\{(Y_t - \mu)/\sigma\}$, and $h_t = u_t/(1 - U_t)$. Then,

$$\frac{\partial U_t}{\partial \mu} = -\frac{u_t}{\sigma}, \quad \frac{\partial^2 U_t}{\partial \mu^2} = -\frac{Y_t - \mu}{\sigma^3} u_t, \quad \frac{\partial \log(1 - U_t)}{\partial \mu} = \frac{h_t}{\sigma}, \quad \frac{\partial \log(1 - U_t)}{\partial \sigma} = \frac{Y_t - \mu}{\sigma^2} h_t,$$

$$\frac{\partial u_t}{\partial \mu} = \frac{Y_t - \mu}{\sigma^2} u_t, \quad \frac{\partial u_t}{\partial \sigma} = \frac{(Y_t - \mu)^2}{\sigma^3} u_t,$$

$$\frac{\partial h_t}{\partial \mu} = \frac{Y_t - \mu}{\sigma^2} h_t - \frac{h_t^2}{\sigma}, \quad \frac{\partial h_t}{\partial \sigma} = \frac{(Y_t - \mu)^2}{\sigma^3} h_t - \frac{Y_t - \mu}{\sigma^2} h_t^2,$$

$$\frac{\partial U_t}{\partial \sigma} = -\frac{Y_t - \mu}{\sigma^2} u_t, \quad \frac{\partial^2 U_t}{\partial (\sigma)^2} = \frac{Y_t - \mu}{\sigma^3} \left\{ 2 - \frac{(Y_t - \mu)^2}{\sigma^2} \right\} u_t,$$

$$\frac{\partial^2 U_t}{\partial \mu \partial \sigma} = \left\{ 1 - \frac{(Y_t - \mu)^2}{\sigma^2} \right\} \frac{u_t}{\sigma}.$$

Let

$$A(U_{t-1}, U_t, \alpha) = (1 - U_{t-1})^\alpha + (1 - U_t)^\alpha - (1 - U_{t-1})^\alpha (1 - U_t)^\alpha,$$

$$\begin{aligned} A_\alpha(U_{t-1}, U_t, \alpha) &= \partial A_\alpha(U_{t-1}, U_t, \alpha) / \partial \alpha \\ &= (1 - U_{t-1})^\alpha \log(1 - U_{t-1}) + (1 - U_t)^\alpha \log(1 - U_t) \\ &\quad - (1 - U_{t-1})^\alpha (1 - U_t)^\alpha \log\{(1 - U_{t-1})(1 - U_t)\}, \end{aligned}$$

$$\begin{aligned} A_{\alpha\alpha}(U_{t-1}, U_t, \alpha) &= \partial^2 A_\alpha(U_{t-1}, U_t, \alpha) / \partial \alpha^2 \\ &= (1 - U_{t-1})^\alpha \{\log(1 - U_{t-1})\}^2 + (1 - U_t)^\alpha \{\log(1 - U_t)\}^2 \\ &\quad - (1 - U_{t-1})^\alpha (1 - U_t)^\alpha [\log\{(1 - U_{t-1})(1 - U_t)\}]^2, \end{aligned}$$

$$\begin{aligned} A_\mu(U_{t-1}, U_t, \alpha) &= \partial A(U_{t-1}, U_t, \alpha) / \partial \mu = \alpha(1 - U_{t-1})^{\alpha-1} \{1 - (1 - U_t)^\alpha\} u_{t-1} / \sigma \\ &\quad + \alpha(1 - U_t)^{\alpha-1} \{1 - (1 - U_{t-1})^\alpha\} u_t / \sigma, \end{aligned}$$

$$\begin{aligned} A_\sigma(U_{t-1}, U_t, \alpha) &= \partial A(U_{t-1}, U_t, \alpha) / \partial \sigma \\ &= \alpha(1 - U_{t-1})^{\alpha-1} \{1 - (1 - U_t)^\alpha\} (Y_{t-1} - \mu) u_{t-1} / \sigma^2 \\ &\quad + \alpha(1 - U_t)^{\alpha-1} \{1 - (1 - U_{t-1})^\alpha\} (Y_t - \mu) u_t / \sigma^2, \end{aligned}$$

$$\begin{aligned} A_{\mu\mu}(U_{t-1}, U_t, \alpha) &= \partial^2 A(U_{t-1}, U_t, \alpha) / \partial \mu^2 \\ &= \alpha(1 - U_{t-1})^{\alpha-1} [(\alpha - 1)(1 - U_{t-1})^{-1} u_{t-1} \{1 - (1 - U_t)^\alpha\} \\ &\quad - \alpha(1 - U_t)^{\alpha-1} u_t + \{1 - (1 - U_t)^\alpha\} (Y_{t-1} - \mu) / \sigma] u_{t-1} / \sigma^2 \\ &\quad + \alpha(1 - U_t)^{\alpha-1} [(\alpha - 1)(1 - U_t)^{-1} u_t \{1 - (1 - U_{t-1})^\alpha\} \\ &\quad - \alpha(1 - U_{t-1})^{\alpha-1} u_{t-1} + \{1 - (1 - U_{t-1})^\alpha\} (Y_t - \mu) / \sigma] u_t / \sigma^2, \end{aligned}$$

$$A_{\alpha\mu}(U_{t-1}, U_t, \alpha) = \partial^2 A(U_{t-1}, U_t, \alpha) / \partial \mu \partial \alpha$$

$$\begin{aligned}
&= (1 - U_{t-1})^{\alpha-1} [1 - (1 - U_t)^\alpha + \alpha \log(1 - U_{t-1}) \{1 - (1 - U_t)^\alpha\} \\
&\quad - \alpha(1 - U_t)^\alpha \log(1 - U_t)] u_{t-1} / \sigma \\
&\quad + (1 - U_t)^{\alpha-1} [1 - (1 - U_{t-1})^\alpha + \alpha \log(1 - U_t) \{1 - (1 - U_{t-1})^\alpha\} \\
&\quad - \alpha(1 - U_{t-1})^\alpha \log(1 - U_{t-1})] u_t / \sigma,
\end{aligned}$$

$$\begin{aligned}
A_{\alpha\sigma}(U_{t-1}, U_t, \alpha) &= \partial^2 A(U_{t-1}, U_t, \alpha) / \partial \sigma \partial \alpha \\
&= (1 - U_{t-1})^{\alpha-1} [1 - (1 - U_t)^\alpha + \alpha \{1 - (1 - U_t)^\alpha\} \log(1 - U_{t-1}) \\
&\quad - \alpha(1 - U_t)^\alpha \log(1 - U_t)] (Y_{t-1} - \mu) u_{t-1} / \sigma^2 \\
&\quad + (1 - U_t)^{\alpha-1} [1 - (1 - U_{t-1})^\alpha + \alpha \{1 - (1 - U_{t-1})^\alpha\} \log(1 - U_t) \\
&\quad - \alpha(1 - U_{t-1})^\alpha \log(1 - U_{t-1})] (Y_t - \mu) u_t / \sigma^2,
\end{aligned}$$

$$\begin{aligned}
A_{\sigma\sigma}(U_{t-1}, U_t, \alpha) &= \partial^2 A(U_{t-1}, U_t, \alpha) / \partial \sigma^2 \\
&= \alpha(1 - U_{t-1})^{\alpha-1} [(\alpha - 1)(1 - U_{t-1})^{-1} \\
&\quad \times (Y_{t-1} - \mu)^2 u_{t-1} \{1 - (1 - U_t)^\alpha\} \\
&\quad - \alpha(1 - U_t)^{\alpha-1} u_t (Y_t - \mu)(Y_{t-1} - \mu) \\
&\quad + \{1 - (1 - U_t)^\alpha\} (Y_{t-1} - \mu)^3 / \sigma] u_{t-1} / \sigma^4 \\
&\quad + \alpha(1 - U_t)^{\alpha-1} [(\alpha - 1)(1 - U_t)^{-1} (Y_t - \mu)^2 u_t \{1 - (1 - U_{t-1})^\alpha\} \\
&\quad - \alpha(1 - U_{t-1})^{\alpha-1} u_{t-1} (Y_{t-1} - \mu)(Y_t - \mu) \\
&\quad + \{1 - (1 - U_{t-1})^\alpha\} (Y_t - \mu)^3 / \sigma] u_t / \sigma^4,
\end{aligned}$$

$$\begin{aligned}
A_{\mu\sigma}(U_{t-1}, U_t, \alpha) &= \partial^2 A(U_{t-1}, U_t, \alpha) / \partial \sigma \partial \mu \\
&= \alpha(1 - U_{t-1})^{\alpha-1} [(\alpha - 1)(1 - U_{t-1})^{-1} \\
&\quad \times (Y_{t-1} - \mu) u_{t-1} \{1 - (1 - U_t)^\alpha\} / \sigma \\
&\quad - \alpha(1 - U_t)^{\alpha-1} u_t (Y_{t-1} - \mu) / \sigma - 1 + (1 - U_t)^\alpha \\
&\quad + \{1 - (1 - U_t)^\alpha\} (Y_{t-1} - \mu)^2 / \sigma^2] u_{t-1} / \sigma^2 \\
&= \alpha(1 - U_t)^{\alpha-1} [(\alpha - 1)(1 - U_t)^{-1} (Y_t - \mu) u_t \{1 - (1 - U_{t-1})^\alpha\} / \sigma \\
&\quad - \alpha(1 - U_{t-1})^{\alpha-1} u_{t-1} (Y_t - \mu) / \sigma - 1 + (1 - U_{t-1})^\alpha \\
&\quad + \{1 - (1 - U_{t-1})^\alpha\} (Y_t - \mu)^2 / \sigma^2] u_t / \sigma^2,
\end{aligned}$$

Then, the log-likelihood is

$$\begin{aligned}
L(\mu, \sigma, \alpha) &= -\frac{1}{2} \log(2\pi) - \log \sigma - \frac{1}{n} \sum_{t=1}^n \frac{(Y_t - \mu)^2}{2\sigma^2} \\
&\quad + \frac{1}{n} \sum_{t=2}^n [\log\{\alpha - 1 + A(U_{t-1}, U_t, \alpha)\} + (\alpha - 1) \log(1 - U_{t-1}) \\
&\quad + (\alpha - 1) \log(1 - U_t) + (1/\alpha - 2) \log A(U_{t-1}, U_t, \alpha)].
\end{aligned}$$

The first-order derivatives are

$$\begin{aligned} \frac{\partial L(\mu, \sigma, \alpha)}{\partial \alpha} &= \frac{1}{n} \sum_{t=2}^n \left[\frac{1 + A_\alpha(U_{t-1}, U_t, \alpha)}{\alpha - 1 + A(U_{t-1}, U_t, \alpha)} + \log(1 - U_{t-1}) + \log(1 - U_t) \right. \\ &\quad \left. - \frac{\log A(U_{t-1}, U_t, \alpha)}{\alpha^2} + \left(\frac{1}{\alpha} - 2\right) \frac{A_\alpha(U_{t-1}, U_t, \alpha)}{A(U_{t-1}, U_t, \alpha)} \right], \\ \frac{\partial L(\mu, \sigma, \alpha)}{\partial \mu} &= \frac{1}{n} \sum_{t=1}^n \frac{Y_t - \mu}{\sigma^2} \\ &= \frac{1}{n} \sum_{t=2}^n \left[\frac{A_\mu(U_{t-1}, U_t, \alpha)}{\alpha - 1 + A(U_{t-1}, U_t, \alpha)} + \frac{\alpha - 1}{\sigma} (h_{t-1} + h_t) \right. \\ &\quad \left. + \left(\frac{1}{\alpha} - 2\right) \frac{A_\mu(U_{t-1}, U_t, \alpha)}{A(U_{t-1}, U_t, \alpha)} \right], \\ \frac{\partial L(\mu, \sigma, \alpha)}{\partial \sigma} &= -\frac{1}{\sigma} + \frac{1}{n} \sum_{t=1}^n \frac{(Y_t - \mu)^2}{\sigma^3} \\ &\quad + \frac{1}{n} \sum_{t=2}^n \left[\frac{A_\sigma(U_{t-1}, U_t, \alpha)}{\alpha - 1 + A(U_{t-1}, U_t, \alpha)} \right. \\ &\quad \left. + (\alpha - 1) \left(\frac{Y_{t-1} - \mu}{\sigma^2} h_{t-1} + \frac{Y_t - \mu}{\sigma^2} h_t \right) + \left(\frac{1}{\alpha} - 2\right) \frac{A_\sigma(U_{t-1}, U_t, \alpha)}{A(U_{t-1}, U_t, \alpha)} \right]. \end{aligned}$$

The second-order derivatives are

$$\begin{aligned} \frac{\partial^2 L(\mu, \sigma, \alpha)}{\partial \alpha^2} &= \frac{1}{n} \sum_{t=2}^n \left[\frac{A_{\alpha\alpha}(U_{t-1}, U_t, \alpha)}{\alpha - 1 + A(U_{t-1}, U_t, \alpha)} - \left\{ \frac{1 + A_\alpha(U_{t-1}, U_t, \alpha)}{\alpha - 1 + A(U_{t-1}, U_t, \alpha)} \right\}^2 \right. \\ &\quad + \frac{2 \log A(U_{t-1}, U_t, \alpha)}{\alpha^3} - \frac{2A_\alpha(U_{t-1}, U_t, \alpha)}{\alpha^2 A(U_{t-1}, U_t, \alpha)} \\ &\quad \left. + \left(\frac{1}{\alpha} - 2\right) \left(\frac{A_{\alpha\alpha}(U_{t-1}, U_t, \alpha)}{A(U_{t-1}, U_t, \alpha)} - \left\{ \frac{A_\alpha(U_{t-1}, U_t, \alpha)}{A(U_{t-1}, U_t, \alpha)} \right\}^2 \right) \right], \\ \frac{\partial^2 L(\mu, \sigma, \alpha)}{\partial \mu \partial \alpha} &= \frac{1}{n} \sum_{t=2}^n \left[\frac{A_{\alpha\mu}(U_{t-1}, U_t, \alpha)}{\alpha - 1 + A(U_{t-1}, U_t, \alpha)} \right. \\ &\quad - \frac{\{1 + A_\alpha(U_{t-1}, U_t, \alpha)\} A_\mu(U_{t-1}, U_t, \alpha)}{\{\alpha - 1 + A(U_{t-1}, U_t, \alpha)\}^2} \\ &\quad + \frac{1}{\sigma} (h_{t-1} + h_t) - \frac{A_\mu(U_{t-1}, U_t, \alpha)}{\alpha^2 A(U_{t-1}, U_t, \alpha)} \\ &\quad \left. + \left(\frac{1}{\alpha} - 2\right) \left(\frac{A_{\alpha\mu}(U_{t-1}, U_t, \alpha)}{A(U_{t-1}, U_t, \alpha)} - \frac{A_\alpha(U_{t-1}, U_t, \alpha) A_\mu(U_{t-1}, U_t, \alpha)}{A(U_{t-1}, U_t, \alpha)^2} \right) \right], \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 L(\mu, \sigma, \alpha)}{\partial \sigma \partial \alpha} &= \frac{1}{n} \sum_{t=2}^n \left[\frac{A_{\alpha\sigma}(U_{t-1}, U_t, \alpha)}{\alpha - 1 + A(U_{t-1}, U_t, \alpha)} \right. \\ &\quad - \frac{\{1 + A_\alpha(U_{t-1}, U_t, \alpha)\}A_\sigma(U_{t-1}, U_t, \alpha)}{\{\alpha - 1 + A(U_{t-1}, U_t, \alpha)\}^2} \\ &\quad + \frac{(Y_{t-1} - \mu)}{\sigma} h_{t-1} + \frac{(Y_t - \mu)}{\sigma} h_t - \frac{A_\sigma(U_{t-1}, U_t, \alpha)}{\alpha^2 A(U_{t-1}, U_t, \alpha)} \\ &\quad \left. + \left(\frac{1}{\alpha} - 2\right) \left(\frac{A_{\alpha\sigma}(U_{t-1}, U_t, \alpha)}{A(U_{t-1}, U_t, \alpha)} - \frac{A_\alpha(U_{t-1}, U_t, \alpha)A_\sigma(U_{t-1}, U_t, \alpha)}{A(U_{t-1}, U_t, \alpha)^2}\right) \right], \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 L(\mu, \sigma, \alpha)}{\partial \mu^2} &= -\frac{1}{\sigma^2} + \frac{1}{n} \sum_{t=2}^n \left[\frac{A_{\mu\mu}(U_{t-1}, U_t, \alpha)}{\alpha - 1 + A(U_{t-1}, U_t, \alpha)} - \frac{A_\mu(U_{t-1}, U_t, \alpha)^2}{\{\alpha - 1 + A(U_{t-1}, U_t, \alpha)\}^2} \right. \\ &\quad + \frac{\alpha - 1}{\sigma^2} \left\{ \frac{Y_{t-1} - \mu}{\sigma} h_{t-1} - h_{t-1}^2 + \frac{Y_t - \mu}{\sigma} h_t - h_t^2 \right\} \\ &\quad \left. + \left(\frac{1}{\alpha} - 2\right) \left(\frac{A_{\mu\mu}(U_{t-1}, U_t, \alpha)}{A(U_{t-1}, U_t, \alpha)} - \frac{A_\mu(U_{t-1}, U_t, \alpha)^2}{A(U_{t-1}, U_t, \alpha)^2}\right) \right], \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 L(\mu, \sigma, \alpha)}{\partial (\sigma)^2} &= \frac{1}{\sigma^2} - \frac{3}{n} \sum_{t=1}^n \frac{(Y_t - \mu)^2}{\sigma^4} \\ &\quad + \frac{1}{n} \sum_{t=2}^n \left[\frac{A_{\sigma\sigma}(U_{t-1}, U_t, \alpha)}{\alpha - 1 + A(U_{t-1}, U_t, \alpha)} - \frac{A_\sigma(U_{t-1}, U_t, \alpha)^2}{\{\alpha - 1 + A(U_{t-1}, U_t, \alpha)\}^2} \right. \\ &\quad + (\alpha - 1)h_{t-1} \frac{Y_{t-1} - \mu}{\sigma^2} \left[-\frac{1}{2\sigma} + \frac{(Y_{t-1} - \mu)^2}{\sigma^3} - \frac{Y_{t-1} - \mu}{\sigma^2} h_{t-1} \right] \\ &\quad + (\alpha - 1)h_t \frac{Y_t - \mu}{\sigma^2} \left[-\frac{1}{2\sigma} + \frac{(Y_t - \mu)^2}{\sigma^3} - \frac{Y_t - \mu}{\sigma^2} h_t \right] \\ &\quad \left. + \left(\frac{1}{\alpha} - 2\right) \left(\frac{A_{\sigma\sigma}(U_{t-1}, U_t, \alpha)}{A(U_{t-1}, U_t, \alpha)} - \frac{A_\sigma(U_{t-1}, U_t, \alpha)^2}{A(U_{t-1}, U_t, \alpha)^2}\right) \right], \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 L(\mu, \sigma, \alpha)}{\partial \sigma \partial \mu} &= -\frac{2}{n} \sum_{t=1}^n \frac{Y_t - \mu}{\sigma^3} \\ &\quad + \frac{1}{n} \sum_{t=2}^n \left[\frac{A_{\mu\sigma}(U_{t-1}, U_t, \alpha)}{\alpha - 1 + A(U_{t-1}, U_t, \alpha)} - \frac{A_\mu(U_{t-1}, U_t, \alpha)A_\sigma(U_{t-1}, U_t, \alpha)}{\{\alpha - 1 + A(U_{t-1}, U_t, \alpha)\}^2} \right. \\ &\quad - \frac{\alpha - 1}{\sigma^2} (h_{t-1} - h_t) + \frac{\alpha - 1}{\sigma^3} \left\{ (Y_{t-1} - \mu) \left(\frac{Y_{t-1} - \mu}{\sigma} h_{t-1} - h_{t-1}^2\right) \right. \\ &\quad \left. \left. + (Y_t - \mu) \left(\frac{Y_t - \mu}{\sigma^2} h_t - h_t^2\right) \right\} \right] \end{aligned}$$

$$+ \left(\frac{1}{\alpha} - 2 \right) \left(\frac{A_{\mu\sigma}(U_{t-1}, U_t, \alpha)}{A(U_{t-1}, U_t, \alpha)} - \frac{A_{\mu}(U_{t-1}, U_t, \alpha)A_{\sigma}(U_{t-1}, U_t, \alpha)}{A(U_{t-1}, U_t, \alpha)^2} \right) \Big].$$

Acknowledgment

The authors thank the editor and reviewer for their helpful comments that led to improvements of their article.

Funding

The work of T. Emura is supported by the research grant funded by the Ministry of Science and Technology of Taiwan (MOST 103-2118-M-008-MY2).

References

- Abegaz, F., Naik-Nimbalkar, U. V. (2008). Dynamic copula-based Markov time series. *Communications in Statistics – Theory and Method* 37:2447–2460.
- Bisgaard, S., Kulahci, M. (2007). Quality quandaries, practical time series modeling II. *Quality Engineering* 19:393–400.
- Brechmann, E. C., Czado, C. (2014). COPAR-multivariate time series modelling using the copula autoregressive model. *Applied Stochastic Models in Business and Industry* 31:495–514. DOI: [10.1002/asmb.2043](https://doi.org/10.1002/asmb.2043)
- Box, G. E. P., Jenkins, G. M. (1976). *Time Series Analysis: Forecasting and Control*. 2nd ed. Englewood Cliffs, NJ: Prentice Hall.
- Box, G., Narasimhan, S. (2010). Rethinking statistics for quality control. *Quality Engineering* 22:60–72.
- Chen, X., Fan, Y. (2006). Estimation of copula-based semiparametric time series models. *Journal of Econometrics* 130:307–335.
- Darsow, W. F., Nguten, B., Olsen, E. T. (1992). Copulas and Markov processes. *Illinois Journal of Mathematics* 36:600–642.
- de Uña-Álvarez, J., Veraverbeke, N. (2013). Generalized copula-graphic estimator. *Test* 22:343–360.
- Domma, F., Giordano, S., Francesco, P. P. (2009). Statistical modeling of temporal dependence in financial data via a copula function. *Communications in Statistics - Simulation and Computation* 38:703–728.
- Emura, T., Chen, Y. H. (2014). Gene selection for survival data under dependent censoring: A copula-based approach. *Statistical Methods in Medical Research*. DOI: [10.1177/0962280214533378](https://doi.org/10.1177/0962280214533378)
- Emura, T., Long, T. H. (2015). *R Copula.Markov package: Estimation and Statistical Process Control Under Copula-Based Time Series Models, Version 1.0*. Available at: <http://cran.r-project.org/web/packages/Copula.Markov/>
- Emura, T., Nakatochi, M., Murotani, K., Rondeau, V. (2015). A joint frailty-copula model between tumour progression and death for meta-analysis. *Statistical Methods in Medical Research*. DOI: [10.1177/0962280215604510](https://doi.org/10.1177/0962280215604510)
- Hu, Y. H., Emura, T. (2015). Maximum likelihood estimation for a special exponential family under random double-truncation. *Computational Statistics* 30:1199–1229. DOI [10.1007/s00180-015-0564-z](https://doi.org/10.1007/s00180-015-0564-z)
- Huh, K. (2014). *Optimal Monitoring Methods for Univariate and Multivariate EWMA Control Charts*. Ph.D. dissertation, McMaster University, Hamilton, Ontario, Canada.
- Hung, Y. C., Tseng, N. F. (2013). Extracting informative variables in the validation of two-group causal relationship. *Computational Statistics* 28:1151–1167.
- Joe, H. (1993). Parametric families of multivariate distributions with given margins. *Journal of Multivariate Analysis* 46:262–282.
- Joe, H. (1997). *Multivariate Models and Dependence Concepts*. Boca Raton, FL: Chapman & Hall/CRC.
- Khuri, A. I. (2003). *Advanced Calculus with Applications in Statistics*. 2nd ed. New York: Wiley.
- Knuth, S., Schmid, W. (2004). Control charts for time series: A review. In: Lenz, H.-J., Wilrich, P. T., eds. *Frontiers in Statistical Quality Control*. Vol. 7. Berlin Heidelberg: Springer-Verlag, pp. 210–236.

- Long, T. H., Emura, T. (2014). A control chart using copula-based Markov chain models. *Journal of the Chinese Statistical Association* 52(4):466–496.
- Montgomery, D. C. (2009). *Statistical Quality Control*. 6th ed. New York: Wiley.
- Nelsen, R. B. (2006). *An Introduction to Copulas (Springer Series in Statistics)*. 2nd ed. New York: Springer-Verlag.
- Sari, J. K., Newby, M. J., Brombacher, A. C., and Tang, L. C. (2009). Bivariate constant stress degradation model: Led lighting system reliability estimation with two-stage monitoring. *Quality and Reliability Engineering International* 25:1067–1084.
- Schepsmeier, U., Stöber, J. (2014). Derivatives and Fisher information of bivariate copulas. *Statistical Papers* 55:525–542.
- Tseng, S.-T., Tang, J., Lin C.-H. (2007). Sample size determination for achieving stability of double multivariate exponentially weighted moving average controller. *Technometrics* 17:73–80.
- Wieringa, J. E. (1999). *Statistical Process Control for Serially Correlated Data*. New York: Labyrinth Publishing.