# ERASMUS UNIVERSITY ROTTERDAM

## Erasmus School of Economics

Bachelor Thesis [FEB63007|2018 Major Quantitative Finance]



---

# Mixture copulas and the housing crisis

---

**Student:**

E.C. Walrave

450778

**Supervisor:**

Dr. J.W.N. Reuvers

Second assessor: Prof. Dr. R. Paap

Date final version: July 7, 2019

## Abstract

The use of copulas in order to model time series is gaining popularity due to its simplicity. Namely, marginal distributions can be obtained from the joint distribution based on some dependence structure and vice versa. An example of an application is Zimmer (2012), who used the copula approach to model dependencies of housing prices in the states Arizona, California, Florida and Nevada, which is important for risk estimation purposes. Previously this was modeled by a Gaussian copula, but this does not allow to model extreme events, like a crisis. Therefore, Zimmer tested several other copulas on filtered quarterly percentage changes of the Housing Price Index and found that a Clayton-Gumbel copula, which is a convex combination of copulas, i.e. a mixture copula, performed better than the Gaussian one. Since Zimmer (2012) tested only one mixture copula, he suggested that a Clayton-survival Clayton or Gumbel-survival Gumbel mixture might outperform the Clayton-Gumbel one. In this thesis, Zimmer's suggestions are tested by adhering to his approach. It follows from Bayes Information Criterion and the Vuong test statistics that the suggested mixture copulas perform similarly to the Clayton-Gumbel one and that the captured tail-dependencies (or extreme observations) do not differ significantly across the copula mixtures.

# Contents

# 1 Introduction

Using copula-based models for describing time series gained popularity in many research areas, like engineering, see for instance Salvadori et al. (2016), or economics and finance, where Van den Goorbergh et al. (2003) and Grégoire et al. (2008) are just some examples. As said by Patton (2012), the reason for this increasing interest is that copula-based models provide a way to link marginal distributions to a joint one by incorporating a desired dependence structure. In addition, from the joint distribution, of which the dependence structure is known, the marginals can be obtained. Thus copulas allow the data to fit a more flexible model such that the estimation accuracy can be increased. Namely, the researcher is now able to assume and estimate multiple marginal distributions simultaneously for one specific model instead of being stuck to one already existing multivariate distribution. As an illustration, both the student's $t$ and the Gaussian marginal distributions can now be considered while creating a joint distribution rather than just one of them.

Another interesting application of copulas, which is central in this thesis, is in the field of mortgages. Mortgages are asset-backed securities, more specifically, the loan should be payed back or the underlying security, in this case a house, is confiscated. By diversifying all provided mortgages based on the credibility of receiving future cash flows, two types of tranches, senior and junior, can be created, where the latter one implies a higher default risk. Next, the junior tranches are distributed over multiple groups and are again rated as senior or junior tranches. These tranches of 'risky' asset-backed securities are called collaterized debt obligations (CDOs). For more information on CDOs, see the book Corporate Finance by Berk and DeMarzo (2017). As mentioned by Zimmer (2012), CDO ratings are estimated by means of copulas, where usually the Gaussian copula is applied. This is due to the fact that it is easy to implement, but it also has a drawback because it assumes asymptotic independence, which is not that realistic during times of crises. Embrechts et al. (2002) add some explanation to this concept, namely "Regardless of how high a correlation we choose, if we go far enough into the tail, extreme events appear to occur independently in each margin". In other words, the Gaussian copula cannot accurately model tail dependence in the data. That is, relating to the dependence of extreme observations present in the data, i.e. the dependence of observations in the upper-right or bottom-left quadrant of a bivariate distribution (Embrechts et al., 2001). Therefore Zimmer (2012) considered other types of copulas to model the dependence of housing prices in four U.S. states, specifically Arizona, California, Florida and Nevada, which were assumed to be affected the most by the housing crisis (which succeeded the housing bubble of 2002). He found that a convex combination of copulas, in particular the Clayton-Gumbel mixture copula, provided the best fit.

Furthermore, Zimmer (2012) mentioned that other copula mixtures might produce a good (or even better) fit as well, as long as the tail dependence of the CDOs can be incorporated. He suggested to investigate the Clayton-survival Clayton and Gumbel-survival Gumbel copula. Note that a Frank copula is one of the four most studied copulas, but it only suits data if there is weak tail dependence (Caia et al., 2006), which is not the case for the housing prices. Consequently, the central research question is:

*Could the Clayton-Gumbel copula be outperformed by a Clayton-survival Clayton or Gumbel-survival Gumbel copula in describing the dependencies in the housing prices of Arizona, California, Florida and Nevada?*

This central research question is motivated by the fact that a copula that better fits the data allows for an improved estimation of the involved risk in a CDO. That, in turn, is relevant for bankers and investors who want to be exposed to only a certain amount of risk in their investment, i.e. they want to be secured that they receive ‘enough’ future cash flows and that the mortgage seller will not default. In addition, copulas could be used for forecasting purposes, see Patton (2013), and in this context could provide useful insights whether to invest in the CDO or not. Namely if the default risk decreases in the future, it is worth to invest in the underlying asset since future cash flows have a higher probability to be paid, but if the default risk will increase, the investment is worth less or even nothing at all. Note however, that this forecasting perspective is not the aim of Zimmer (2012) nor this thesis.

Following Zimmer's approach, the first step is to convert the data (the Housing Price Index, see Section 3) to quarterly percentage changes for the four states. These time series are then cleaned of autocorrelation and autoregressive conditional heteroskedasticity. Next, the created (normal) marginal probability distributions serve as input for the desired copula specification, where maximum likelihood provides the estimated dependence parameters. These are then converted to Kendall's $\tau$ for ease of comparison between copulas. Also the captured tail-dependencies can be obtained from the estimates of dependence, which differ quite a bit over the mixture copulas, though not significantly due to reasonably big standard errors. Finally, Bayes Information Criterion (BIC) and Vuong test statistics indicate that the Clayton-Gumbel copula cannot be outperformed by the Clayton-survival Clayton and the Gumbel-survival Gumbel copulas.

This thesis starts by elaborating on the rather small amount of literature on this subject in Section 2. Then, in Section 3 the data is discussed in more detail. Next the methodology is explained in Section 4. This includes the general framework of copulas, the specifications of the studied bivariate copulas together with formulas for their tail dependencies and Kendall's $\tau$, the marginal distributions that serve as input for the (mixture) copulas, the estimation method and the selection criteria for the copulas. The corresponding results can be found in Section 5. Lastly, the conclusion is provided in Section 6.

## 2 Literature

Mixture copulas were already used to model dependence structures in financial markets by Hu (2006). Though before working with the copulas, Hu (2006) filters the GARCH components out of the financial data such that possible conditional heteroskedasticity is removed. After this, Hu (2006) used a mixture model consisting of the Gaussian, Gumbel and survival Gumbel copula to be able to capture the extreme events and tail dependencies of the financial data. Additionally, Hu (2006) compares the performance of this convex combination of copulas to the most commonly used Gaussian copula. Similarly, Zimmer (2012) first applied a GARCH filter to the data, though he also removed the autocorrelation that was present. Next, several copulas were compared to the Gaussian one by using Bayes Information Criterion (BIC) and Vuong test statistics. Zimmer (2012) concluded that the Clayton-Gumbel mixture copula performed significantly better than the Gaussian one. Therefore, this thesis focuses on testing whether this mixture copula can be outperformed by other mixture copulas, particularly the Clayton-survival Clayton and Gumbel-survival Gumbel, which are both able to capture lower- and upper-tail dependence. Furthermore, the 'same' housing data and methodology as in Zimmer (2012) is considered.

# 3 Data

Similar to Zimmer (2012), the Housing Price Index (HPI) from the Federal Housing Finance Agency is used for the states Arizona (AZ), California (CA), Florida (FL) and Nevada (NV). This measure includes price movements of single-family houses only and is weighted by a repeat-sales technique. More specifically, it incorporates average price changes of single-family houses when they are refinanced by the mortgage buyer companies Fannie Mae or Freddie Mac. Therefore, the HPI could be seen as an accurate indicator for house price trends over time, starting from January 1975.

Zimmer (2012) considered the quarterly percentage changes from the second quarter of 1975 to the first quarter of 2009 (136 observations). After converting the HPI data into quarterly percentage changes for the specified time period, some small deviations with Zimmer (2012) can be observed. These changes could be explained by the revision of past HPI values by the Federal Housing Finance Agency based on new information of mortgages and housing prices obtained from the companies. For the ease of comparison pairwise scatter plots are shown for each state pair in Figure 1 in Appendix A.

# 4 Methodology

In this section, mainly the methodology of Zimmer (2012) is pursued. In some parts the explanation is extended for clarification. First, the general theory behind copulas is explained in Section 4.1. Next, in Section 4.2, the bivariate copula specifications and their properties are given. Then the process of obtaining the marginal distributions and the estimation of the copulas is provided in Sections 4.3 and 4.4 respectively. Lastly, the criteria or tests leading to the 'best' copula are discussed in Section 4.5.

## 4.1 General framework of copulas

The short explanation that follows, is based on concise summaries of Patton (2012) and Zimmer (2012) on the framework of copulas, which was first introduced by Sklar in 1959. Namely, an $n$-copula could be defined as an $n$-dimensional distribution consisting of $n$ univariate marginal distributions distributed as $U(0,1)$. Thus, mathematically we can define a copula $C$ as $C$: $[0,1]^n \rightarrow [0,1]$.

Consider a continuous $n$-dimensional joint distribution function, $F(y_1,\ldots,y_n)$, where each random variable $y_i$, for $i=1,\ldots,n$, has a univariate marginal cumulative distribution function (cdf) $F_1(y_1),\ldots,F_n(y_n)$ and a corresponding inverse $F_1^{-1},\ldots,F_n^{-1}$. Assuming that variables $u_1,\ldots,u_n$ are $U(0,1)$ distributed, it follows that $y_1=F_1^{-1}(u_1)\sim F_1,\ldots,y_n=F_n^{-1}(u_n)\sim F_n$. In words, this notation implies that the inverse transformation of the marginal distribution, where uniform variables serve as input, are distributed according to the original marginals $F_1,\ldots,F_n$. Therefore,

$$F(y_1,\ldots,y_n)=F(F_1^{-1}(u_1)),\ldots,F(F_n^{-1}(u_n))=C(u_1,\ldots,u_n),$$

where the copula is unique only if the marginal cdfs are continuous. Since the joint distribution is parameterized by the copula and its marginal distributions by means of a dependence structure, a dependence parameter must be estimated as well. This leads to the following representation

$$F(y_1,\ldots,y_n)=C(F_1(y_1),\ldots,F_n(y_n);\theta),$$

where $\theta$ is the dependence parameter. Some copulas restrict this dependence parameter to be in a certain interval. Moreover, notice that due to the specification of a copula, it is easy to obtain a characterization of the joint cdf if the the marginal cdfs are known.

## 4.2 Bivariate copula specifications

The copulas under investigation are the Clayton-Gumbel, Clayton-survival Clayton and the Gumbel-survival Gumbel. These are discussed in more detail in Poulin et al. (2007) and Hu (2006). The former explains the Clayton, survival Clayton and Gumbel copulas, while the latter specifies the survival Gumbel and mixtures of copulas. The specification of the copulas and their dependence parameter restrictions are displayed in Table 1.

Table 1: Copula specifications and $\theta$ domain.

| Copula | $C(u_1,u_2;\theta)$ | $\theta$ domain |
|---|---|---|
| Clayton | $(u_1^{-\theta}+u_2^{-\theta}-1)^{-1/\theta}$ | $(0,\infty)$ |
| Gumbel | $exp\{-[(-ln(u_1))^{\theta}+(-ln(u_2))^{\theta}]^{1/\theta}\}$ | $[1,\infty)$ |
| survival Clayton | $u_1+u_2-1+\{(1-u_1)^{-\theta}+(1-u_2)^{-\theta}-1\}^{-1/\theta}$ | $(0,\infty)$ |
| survival Gumbel | $u_1+u_2-1+exp\{-[(-ln(1-u_1))^{1/\theta}+(-ln(1-u_2))^{1/\theta}]^{\theta}\}$ | $(0,1]$ |
| Mixture | $\sum_i w_iC_i(u_1,u_2;\theta)$<br>where $w_i\in[0,1]$ and $\sum_i w_i=1$ for every copula $i$ | depends on<br>copula $i$ |

As can be seen from Table 1, the mixture copula's $\theta$ domain depends on the involved copulas. For example, a Clayton-Gumbel copula has the $\theta$ domain restriction corresponding to the Clayton copula for the Clayton part, while the Gumbel part adheres to the $\theta$ domain restriction for the Gumbel copula. Furthermore, notice that the Clayton and survival Clayton copulas are mirror images. Therefore, the tail dependence captured by these copulas are exactly opposite, see Table 2. This is also true for the Gumbel and survival Gumbel copulas, though it is harder to see because $\theta$ in the Gumbel specification is defined as $\frac{1}{\theta}$ in the survival Gumbel copula in Table 1.

Table 2: Copula tail dependence and Kendall's $\tau$.

| Copula | Lower-tail dependence | Upper-tail dependence | Kendall's $\tau$ |
|---|---|---|---|
| Clayton | $2^{-\frac{1}{\theta}}$ | 0 | $\frac{\theta}{\theta+2}$ |
| Gumbel | 0 | $2 - 2^{\frac{1}{\theta}}$ | $1 - \frac{1}{\theta}$ |
| survival Clayton | 0 | $2^{-\frac{1}{\theta}}$ | $\frac{\theta}{\theta+2}$ |
| survival Gumbel | $2 - 2^{\theta}$ | 0 | $1 - \theta$ |

Note: the copulas display only lower- or upper-tail dependence.

In the mixture copula the tail dependence per copula can be calculated. So, alike the previous example of a Clayton-Gumbel mixture, the lower-tail dependence can be calculated by inputting the dependence parameter $\theta$ corresponding to the Clayton part in the formula for lower-tail dependence of the Clayton copula, which is displayed in Table 2. This is due to the fact that the Gumbel copula cannot capture lower-tail dependence. However, it is able to incorporate upper-tail dependence, while the Clayton copula cannot. Correspondingly the upper-tail dependence of the Clayton-Gumbel mixture is computed by inserting the dependence parameter $\theta$ corresponding to the Gumbel part in the formula provided for the Gumbel copula in Table 2. Moreover, notice that for the first three copulas shown in Table 2, the larger $\theta$, the more tail dependence is present, while for the last one it is less.

In addition to the tail dependence, Kendall's $\tau$ is displayed in Table 2 for the specified single copulas. This is added, because the estimated dependence parameters cannot directly be compared across copulas (Zimmer, 2012). Consequently, we need to convert them to a measure for comparison, like Kendall's $\tau$. This measure is bounded on the interval $[-1, 1]$, where the interpretation is quite similar to correlations. More specifically, $-1$ corresponds to perfect negative dependence, 0 to independence and 1 to perfect positive dependence. Furthermore, Kendall's $\tau$ could in principle be computed for an entire mixture copula as well by

$$\tau = 4 \iint_{[0,1]^2} C(u_1, u_2)\ \partial C(u_1, u_2) - 1 = 4 \iint_{[0,1]^2} C(u_1, u_2) \cdot c(u_1, u_2)\ \partial u_1 \partial u_2, \tag{1}$$

where $C(u_1, u_2)$ is the copula cdf and $c(u_1, u_2)$ the corresponding copula probability density function (pdf) (Poulin et al., 2007). However, for the copula comparisons in this thesis, it seems more informative to compare Kendall's $\tau$ per copula part of the mixture. Namely, the Clayton $\tau$'s in the Clayton-Gumbel and Clayton-survival Clayton mixture copulas can then be compared directly and differences in value would be easier to perceive. This counts for the Gumbel $\tau$'s in the Clayton-Gumbel and Gumbel-survival Gumbel mixture too. In addition, notice that Equation 1 would lead to a more lengthy calculation for the mixture copulas than for the single ones. Therefore, in Section 5.2 Kendall's $\tau$ is examined only per copula part of the mixture. Thus, in case of the Clayton-Gumbel mixture, Kendall's $\tau$ is computed for the Clayton and Gumbel part separately according to the formulas in Table 2.

### 4.3 Specifying the marginal distributions

In this section, the approach of Zimmer (2012) is followed in order to find the marginal distributions $F_1(y_{1,t})$ and $F_2(y_{2,t})$, where $y_{1,t}$ and $y_{2,t}$ are the quarterly percentage changes in HPI between two states from time $t-1$ to $t$. When analyzing the correlogram of quarterly percentage changes for Arizona, California, Florida and Nevada, it turns out that autocorrelation is present in all of them. In addition, it should not be surprising if conditional heteroskedasticity is present in the times series, because it is found in most financial data. Since both serial correlation and conditional heteroskedasticity could lead to invalid conclusions about the dependence between $y_{1,t}$ and $y_{2,t}$, these AR-GARCH effects must be filtered out of the time series. This can be done by describing each univariate marginal cdf as an AR(1)-GARCH(1,1) model:

$$y_{j,t} = \beta_{j,0} + \beta_{j,1} y_{j,t-1} + \epsilon_{j,t}, \text{ for } j = 1, 2.$$

Error term $\epsilon_{j,t}$ follows a normal distribution with mean zero and conditional variance

$$\sigma^2_{j,t} = \alpha_{j,0} + \alpha_{j,1}\epsilon^2_{j,t-1} + \delta_j \sigma^2_{j,t-1}.$$

Next, the AR(1) and GARCH(1,1) components are taken out of the percentage changes in the HPI by setting

$$\tilde{y}_{j,t} = \frac{\hat{\epsilon}_{j,t}}{\sqrt{\hat{\alpha}_{j,0} + \hat{\alpha}_{j,1}\hat{\epsilon}^2_{j,t-1} + \hat{\delta}_j \hat{\sigma}^2_{j,t-1}}}, \text{ for } j = 1, 2.$$

To check whether there is no autocorrelation present in the filtered time series $\tilde{y}_{j,t}$, the correlogram is studied again. It follows that there is no autocorrelation for the states California and Florida, though for Arizona and Nevada there might still be some doubts about the significance of the third-order autocorrelation. Therefore Breusch-Godfrey tests for serial correlation are performed by means of first regressing the filtered time series on a constant and the first lag of this filtered time series via ordinary least squares (OLS). The residuals of this regression are then regressed on the corresponding first-order lagged residuals and first-order lagged filtered time series. The $R^2$ of this OLS regression multiplied by the amount of observations is the test statistic, which is compared to the critical value $\chi^2(1)$. These Breusch-Godfrey tests confirm that there is no autocorrelation present in any of the filtered time series. Hence the null hypothesis of no autocorrelation cannot be rejected at a 5% significance level.

Then, to see whether the filtered series exhibit conditional heteroskedasticity, ARCH Lagrange Multiplier tests are performed on the ARCH(1)-GARCH(1,1) models for each state. These tests have the null hypothesis that there is no ARCH up to order 1 in the residuals. For all four filtered percentage changes, the null hypothesis cannot be refuted, leading to the conclusion that there is no conditional heteroskedasticity present. Note that in Eviews this test is already pre-programmed and can simply be performed using a command. See the code in Appendix C.

Finally, the Jarque-Bera test for normality concludes that the filtered time series of California and Nevada are not normally distributed, while the ones of Arizona and Florida are. More specific, the test statistics for California, Nevada, Arizona and Florida are respectively 127.9, 18.0, 1.7 and 2.6. Zimmer (2012) had a similar problem and still assumed normal marginals, because ratings agencies were assumed to do this as well. Since this thesis adheres to Zimmer's approach, normal probability distributions are assumed for all filtered percentage changes.

## 4.4 Copula estimation

To obtain the joint distribution, $\tilde{y}_{1,t}$ and $\tilde{y}_{2,t}$ (see Section 4.3 for more information on them) serve as input for the copula:

$$F(\tilde{y}_{1,t}, \tilde{y}_{2,t}) = C(F_1(\tilde{y}_{1,t}), F_2(\tilde{y}_{2,t}); \theta).$$

The pdf version of the copula is

$$c(F_1(\tilde{y}_{1,t}), F_2(\tilde{y}_{2,t}); \theta) = \frac{\partial C(F_1(\tilde{y}_{1,t}), F_2(\tilde{y}_{2,t}); \theta)}{\partial F_1(\tilde{y}_{1,t})\ \partial F_2(\tilde{y}_{2,t})} \cdot f_1(\tilde{y}_{1,t}) \times f_2(\tilde{y}_{2,t}), \tag{2}$$

where $f_1(\tilde{y}_{1,t})$ and $f_2(\tilde{y}_{2,t})$ represent the marginal pdfs corresponding to cdfs $F_1(\tilde{y}_{1,t})$ and $F_2(\tilde{y}_{2,t})$ respectively. Note that the pdf versions of the single copulas in Table 3 can be used in deriving the pdfs of the mixture copulas.

To estimate the parameters in the specification of a copula, maximum likelihood can be applied, see Zimmer (2012). In the first step, the natural logarithm of Equation 2 is taken. This is then summed over all observations. Next, in the final step of the estimation procedure, the obtained expression is maximized with respect to $\theta$ such that maximum likelihood estimates for $\theta$ can be acquired. As discussed in Section 4.2, these estimates for $\theta$ can be transformed to Kendall's $\tau$ such that it is easier to compare copulas.

Due to the GARCH estimations explained in Section 4.3, the obtained maximum likelihood standard errors might be incorrect. That is why they are calculated by means of a block bootstrap approach. In the first step overlapping blocks of 20 consecutive observations are drawn. Then seven blocks are randomly chosen such that a time series as long as the original one can be made. With this sample, the AR(1)-GARCH(1,1) estimation and the copula models are re-estimated. This procedure is iterated 500 times and the standard deviation of the obtained maximum likelihood estimates serves as the standard deviation of the original estimates.

Table 3: The pdf of different copula specifications.

| Copula | $\frac{\partial C}{\partial u_1 \partial u_2}$ |
|---|---|
| Clayton | $(\theta+1)\ (u_1u_2)^{-\theta-1}\ (u_1^{-\theta} + u_2^{-\theta} - 1)^{-\frac{1}{\theta}-2}$ |
| Gumbel | $\exp\Big\{-(\tilde{u}_1{}^{\theta} + \tilde{u}_2{}^{\theta})^{\frac{1}{\theta}}\Big\}\ (u_1u_2)^{-1}\ (\tilde{u}_1\tilde{u}_2)^{\theta-1}\ (\tilde{u}_1{}^{\theta} + \tilde{u}_2{}^{\theta})^{-(2-\frac{1}{\theta})}\ \Big\{(\tilde{u}_1{}^{\theta} + \tilde{u}_2{}^{\theta})^{\frac{1}{\theta}} + \theta - 1\Big\}$ |
| | where $\tilde{u}_j = -\ln(u_j)$ for $j = 1, 2$ |
| survival Clayton | $\theta\ (\frac{1}{\theta}+1)\ \{(1-u_1)(1-u_2)\}^{-\theta-1}\ \{(1-u_1)^{-\theta} + (1-u_2)^{-\theta} - 1\}^{-\frac{1}{\theta}-2}$ |
| survival Gumbel | $\exp\Big\{-(\tilde{u}_1{}^{\frac{1}{\theta}} + \tilde{u}_2{}^{\frac{1}{\theta}})^{\theta}\Big\}\ (u_1-1)^{-1}\ (u_2-1)^{-1}\ (\tilde{u}_1 + \tilde{u}_2)^{\frac{1}{\theta}-1}\ \ldots$ |
| | $\times\Big[({}_1{}^{\frac{1}{\theta}} + \tilde{u}_2{}^{\frac{1}{\theta}})^{2\theta-2}\ -\ \frac{\theta-1}{\theta}\ (\tilde{u}_1{}^{\frac{1}{\theta}} + \tilde{u}_2{}^{\frac{1}{\theta}})^{\theta-2}\Big]$ |
| | where $\tilde{u}_j = -\ln(1-u_j)$ for $j = 1, 2$ |

## 4.5 Selection criteria for the copulas

In order to select the 'best' copula, or at least to see whether the Clayton-Gumbel copula can be outperformed, Bayes information criterion (BIC) and the Vuong test statistic are considered. The former is explained in Section 4.5.1 and the latter in Section 4.5.2. Note that in principle one could compare any type of copula using these selection criteria, so for example, a single copula, like the Clayton, could be compared to a mixture copula, such as the Clayton-Gumbel, without any problem.

### 4.5.1 BIC

BIC indicates which copula provides the best fit of the model. The lower the value, the better the fit. It can be calculated as

$$BIC = -2\ \ln(L) + k\ \ln(n),$$

where $\ln(L)$ is the log-likelihood, $k$ the amount of parameters and $n$ the sample size. Note that in the considered mixture copulas, the number of parameters and the sample size are the same for each, so the last term does not really 'penalize' in this case. Hence, the differences in BIC values can be fully explained by deviations in the log-likelihood.

### 4.5.2 Vuong test statistic

The Vuong test statistic for non-nested models compares each mixture copula to the baseline Clayton-Gumbel copula, where significant negative (positive) values imply that the baseline copula is better (worse). Assume that

$$m_i = \ln\left(\frac{c_1(u_1,u_2;\hat{\theta}_1)}{c_2(u_1,u_2;\hat{\theta}_2)}\right) = \ln(c_1(u_1,u_2;\hat{\theta}_1)) - \ln(c_2(u_1,u_2;\hat{\theta}_2)),$$

where the Clayton-Gumbel mixture pdf is defined as $c_2(u_1,u_2;\hat{\theta}_2)$ and the other mixture pdf (Clayton-survival Clayton or Gumbel-survival Gumbel) as $c_1(u_1,u_2;\hat{\theta}_1)$. Then the test statistic can be calculated according to Vuong (1989), that is

$$Z = \frac{1}{\sqrt{n}}\sum_{i=1}^{n}\frac{m_i}{\hat{w}_n} = \frac{\sum_{i=1}^{n} m_i}{\sqrt{\sum_{i=1}^{n}\left(m_i - \frac{1}{n}\sum_{i=1}^{n} m_i\right)^2}} \xrightarrow{D} N(0,1). \tag{3}$$

In Equation 3 the variable $\hat{w}_n$ can be seen. This term, together with the $\frac{1}{\sqrt{n}}$, represents a normalization factor equal to $\sqrt{\sum_{i=1}^{n}(m_i - \frac{1}{n}\sum_{i=1}^{n} m_i)^2}$. Thus $\hat{w}_n$ could be interpreted as the standard deviation. Notice that a possible arising problem is that the denominator might equal zero. However, this does not occur for the considered copulas such that this case is not of interest in this thesis.

If $Z$ is greater than the critical value, $c_1(u_1,u_2;\hat{\theta}_1)$ is preferred over the Clayton-Gumbel mixture. On the other hand, if $Z$ is smaller than the negative of the critical value, the Clayton-Gumbel copula is favoured over the other copula mixture. When the test statistic lays between $\pm$ the critical value, the two copulas have the same explanatory power.

# 5 Results

In order to find the 'best' copula, the tests described in Sections 4.5.1 and 4.5.2 are used. The results of BIC and the Vuong tests can be found in Section 5.1, while Kendall's $\tau$ and the tail dependencies are shown in Section 5.2. These estimates help to explain possible differences between the considered copula mixtures.

## 5.1 Selecting the 'best' copula

To see which copula mixture fits the data best, the BIC values are calculated. These are presented in Table 4. Remember from Section 4.5.1 that a lower BIC value indicates a better fit.

Table 4: BIC values for the copulas.

| Copula | AZ-CA | AZ-FL | AZ-NV | CA-FL | CA-NV | FL-NV |
|---|---|---|---|---|---|---|
| Clayton-Gumbel | 725.90 | 702.78 | 707.09 | 748.32 | 740.82 | 724.07 |
| Clayton-survival Clayton | 723.41* | 705.76 | 709.72 | 747.44 | 741.02 | 725.04 |
| Gumbel-survival Gumbel | 723.48 | 700.79* | 704.01* | 747.08* | 738.20* | 723.74* |

Note: * indicates the smallest BIC value.

From Table 4 the Gumbel-survival Gumbel displays the lowest BIC values for 5 out of 6 state pairs, while the Clayton-survival Clayton is the best for the remaining pair. Though, when comparing the BIC values, they are quite close for all mixtures. Therefore, to see if the Clayton-Gumbel mixture is outperformed by any of the other mixtures, the Vuong test statistics are determined, as described in Section 4.5.2. These are shown in Table 5.

Table 5: The Vuong test statistics.

| Copula | AZ-CA | AZ-FL | AZ-NV | CA-FL | CA-NV | FL-NV |
|---|---|---|---|---|---|---|
| Clayton-survival Clayton | 1.55 | -1.81* | -1.51 | 0.62 | -0.11 | -0.78 |
| Gumbel-survival Gumbel | 0.63 | 0.76 | 0.60 | 0.99 | 1.08 | 0.32 |

Note: * indicates that the mixture copula is inferior to the Clayton-Gumbel one at a 0.10 level.

The Vuong test statistics of Table 5 show that the null hypothesis of equal explanatory power cannot be rejected at a 5% significance level. Though when considering a 10% significance level, it can be seen that the Clayton-Gumbel mixture performs better than the Clayton-survival Clayton one for the state pair Arizona-Florida. Thus, the Clayton-Gumbel mixture has the same explanatory power as the Clayton-survival Clayton and the Gumbel-survival Gumbel copulas when a 5% significance level is considered. In other words, the Clayton-Gumbel copula is not inferior nor superior to any of the other two mixture copulas at a 5% significance level.

## 5.2 Estimates of Kendall's $\tau$ and tail dependence

Estimates of dependence are shown in Table 6 for the Clayton-Gumbel, Clayton-survival Clayton and Gumbel-survival Gumbel copulas. Note that Kendall's $\tau$ measures the dependence across the whole distribution, while the tail dependence only measures the dependence in the tails (Zimmer (2012)). These estimates are computed according to the formulas in Table 2. Furthermore, the displayed standard errors are calculated by the block bootstrap approach explained in Section 4.4.

Table 6: Estimates of Kendall's $\tau$ and the tail dependence per copula.

| Copula | AZ-CA | AZ-FL | AZ-NV | CA-FL | CA-NV | FL-NV |
|---|---|---|---|---|---|---|
| Clayton-Gumbel | | | | | | |
| Clayton $\tau$ | 0.56 (0.12) | 0.60 (0.28) | 0.64 (0.28) | 0.42 (0.17) | 0.38 (0.20) | 0.33 (0.24) |
| Gumbel $\tau$ | 0.20 (0.12) | 0.38 (0.21) | 0.34 (0.24) | 0.23 (0.15) | 0.30 (0.19) | 0.39 (0.19) |
| Lower-tail dependence | 0.77 (0.14) | 0.79 (0.36) | 0.82 (0.36) | 0.62 (0.20) | 0.57 (0.23) | 0.49 (0.31) |
| Upper-tail dependence | 0.26 (0.14) | 0.47 (0.23) | 0.42 (0.25) | 0.30 (0.17) | 0.38 (0.21) | 0.48 (0.22) |
| Proportion Clayton ($\hat{w}$) | 0.52 (0.16) | 0.23 (0.18) | 0.34 (0.19) | 0.43 (0.19) | 0.37 (0.21) | 0.33 (0.17) |
| Clayton-survival Clayton | | | | | | |
| Clayton $\tau$ | 0.57 (0.11) | 0.53 (0.23) | 0.62 (0.25) | 0.42 (0.16) | 0.40 (0.18) | 0.35 (0.20) |
| survival Clayton $\tau$ | 0.15 (0.13) | 0.34 (0.20) | 0.30 (0.23) | 0.17 (0.13) | 0.22 (0.17) | 0.37 (0.18) |
| Lower-tail dependence | 0.77 (0.12) | 0.73 (0.29) | 0.81 (0.31) | 0.62 (0.17) | 0.59 (0.18) | 0.53 (0.27) |
| Upper-tail dependence | 0.14 (0.21) | 0.51 (0.29) | 0.44 (0.32) | 0.19 (0.21) | 0.29 (0.25) | 0.55 (0.27) |
| Proportion Clayton ($\hat{w}$) | 0.54 (0.16) | 0.40 (0.15) | 0.41 (0.16) | 0.49 (0.19) | 0.45 (0.20) | 0.45 (0.17) |
| Gumbel-survival Gumbel | | | | | | |
| Gumbel $\tau$ | 0.16 (0.17) | 0.29 (0.26) | 0.65 (0.29) | 0.22 (0.19) | 0.30 (0.26) | 0.45 (0.24) |
| survival Gumbel $\tau$ | 0.51 (0.10) | 0.59 (0.24) | 0.24 (0.23) | 0.36 (0.15) | 0.39 (0.14) | 0.28 (0.19) |
| Lower-tail dependence | 0.60 (0.10) | 0.67 (0.26) | 0.31 (0.25) | 0.44 (0.16) | 0.47 (0.15) | 0.36 (0.21) |
| Upper-tail dependence | 0.22 (0.19) | 0.36 (0.29) | 0.72 (0.32) | 0.28 (0.22) | 0.37 (0.28) | 0.53 (0.27) |
| Proportion Gumbel ($\hat{w}$) | 0.29 (0.16) | 0.49 (0.21) | 0.47 (0.18) | 0.31 (0.20) | 0.30 (0.22) | 0.52 (0.20) |

Note: the standard deviation is given in parenthesis.

First, the Clayton-Gumbel and Clayton-survival Clayton copulas are compared by means of Table 6. The value for Kendall's $\tau$ in the Clayton part is quite similar across the two copula mixtures. Though for 3 out of 6 state pairs the Clayton dependence is larger in size for the Clayton-survival Clayton copula, for 1 out of 6 equal and smaller for 2 out of 6 pairs. The minor differences in the dependence captured by the Clayton part across the copulas can also be seen from the estimates of lower-tail dependence. Namely, the higher the value of Kendall's $\tau$ in the Clayton part, the bigger is the lower-tail dependence, which is a logical result from the formulas described in Section 4.2. Furthermore, a striking difference between the mixture copulas seems to be the proportion given to the Clayton part, because the Clayton-survival Clayton gives more weight to the Clayton copula than the Clayton-Gumbel mixture for all state pairs. However, due to large standard deviations, the estimates of the two mixtures do not differ significantly. Moreover, the value given to Kendall's $\tau$ corresponding to the Gumbel part is bigger than the one belonging to the survival-Clayton part.

This might suggest that the Gumbel copula is relatively ‘better’ at capturing the upper-tail dependence of the data than the survival Clayton copula, which can also be seen for 4 out of 6 state pairs when considering the BIC values and Vuong test statistics displayed in Tables 4 and 5 respectively. Nevertheless, due to large standard deviations, the Gumbel and survival-Clayton estimates do not differ significantly.

Adding the Gumbel-survival Gumbel to this discussion, the most striking result is probably the difference in captured tail dependence across the different mixtures. According to the BIC values in Table 4, the Gumbel-survival Gumbel mixture always outperforms the Clayton-Gumbel one and in 5 out of 6 cases the Clayton-survival Clayton copula, although not by a huge amount. Therefore small deviations in the captured tail dependencies were expected, but for the state pair Arizona-Nevada for example, the magnitude of the upper-tail dependence explained by the Gumbel-survival Gumbel copula is between 64% and 71% bigger, while the lower tail dependence is about 62% smaller. Actually the lower tail dependence captured by the survival Gumbel is lower than in any other mixture. Thus it might seem that the survival Gumbel copula is not that suitable for describing lower tail dependence, but again the large standard errors do not make this result significant.

Since the survival Clayton and survival Gumbel specification seemed to perform worse than the Gumbel and Clayton copula respectively, but were ‘saved’ by the large standard errors, it seems interesting to check how the survival Clayton-survival Gumbel copula competes with the Clayton-Gumbel one. This is done in Appendix B.

# 6 Conclusion

Lately, the use of copulas became more and more popular. One example of an application is Zimmer (2012), who thought of using different copulas to model housing prices in four U.S. states that were affected the most by the housing crisis, namely Arizona, California, Florida and Nevada. His research led to the conclusion that the commonly used Gaussian copula could be outperformed by a convex combination of copulas, the Clayton-Gumbel mixture. Therefore, one could question whether other mixture copulas would perform even better. This idea and Zimmer’s suggestions guided towards the research question, which is whether the Clayton-Gumbel copula could be outperformed by a Clayton-survival Clayton or Gumbel-survival Gumbel copula in describing the dependencies in the housing prices of Arizona, California, Florida and Nevada.

In order to see whether the aforementioned mixtures can outperform the Clayton-Gumbel copula, the approach of Zimmer (2012) was pursued. When comparing the used data, some small differences can be observed due to the fact that the housing price index is updated every time new information is available. The quarterly percentage changes of this index exhibited serial correlation. Therefore, autocorrelation and autoregressive conditional heteroskedasticity (usually present in financial data) were filtered out by means of an AR(1)-GARCH(1,1) model. Testing for normality led to the conclusion that the filtered time series for

California and Nevada were not normally distributed, while the ones for Arizona and Florida were. Though, adhering to Zimmer (2012), it was assumed that a normal distribution could accurately describe them. These normal marginal distributions were then substituted in the Clayton-Gumbel, Clayton-survival Clayton and Gumbel-survival Gumbel mixture copula specifications via which estimates of the dependence parameter were obtained by maximum likelihood. For comparison reasons, these estimates were converted to Kendall's $\tau$ and the amount of lower- and upper-tail dependence were calculated. Their standard deviations were found by means of a block bootstrap approach. In addition to this, BIC values and Vuong test statistics were computed.

From the BIC values followed that the Gumbel-survival Gumbel mixture fits the data best for 5 out of 6 state pairs, while the other state pair is best described by the Clayton-survival Clayton. However, this did not imply that these mixture copulas outperform the Clayton-Gumbel one. Namely, the Vuong test statistics indicated that none of these mixtures was better in describing the data than the Clayton-Gumbel one. This seemed quite remarkable for several state pairs when looking at the differences in Kendall's $\tau$ and the tail dependencies, but the deviations were not significant due to large standard errors. In conclusion, the Clayton-Gumbel mixture cannot be outperformed by the Clayton-survival Clayton or Gumbel-survival Gumbel copulas in explaining the dependencies of housing prices for the states Arizona, California, Florida and Nevada.

For further research there are several suggestions. First of all, one could search for a different distribution for the states California and Nevada, since they are not normally distributed. Note however, that Zimmer (2012) mentioned that the student's $t$-distribution with different degrees of freedom led to nearly identical results. Secondly, there are many other copulas or copula mixtures that could be considered. For example, a Clayton-Gaussian-Gumbel mixture. Though, this would imply that the BIC values are penalized more, because the number of estimated parameters increases, which makes it harder to get a better fit. Lastly, the predictive accuracy of the different copulas could be studied, which is an important aspect for risk estimation and forecasting.

# References


Berk, J. B. and DeMarzo, P. M. (2017). *Corporate finance*. Pearson Education Limited.

Caia, Z., Chen, X., Fan, Y., and Wang, X. (2006). Selection of copulas with applications in finance. Technical report, Citeseer.

Embrechts, P., Lindskog, F., and McNeil, A. (2001). Modelling dependence with copulas. *Rapport technique, Département de mathématiques, Institut Fédéral de Technologie de Zurich, Zurich.*

Embrechts, P., McNeil, A., and Straumann, D. (2002). Correlation and dependence in risk management: properties and pitfalls. *Risk management: value at risk and beyond*, 1:176–223.

Goorbergh, van den, R. W., Genest, C., and Werker, B. J. (2003). Multivariate option pricing using dynamic copula models.

Grégoire, V., Genest, C., and Gendron, M. (2008). Using copulas to model price dependence in energy markets. *Energy risk*, 5(5):58–64.

Hu, L. (2006). Dependence patterns across financial markets: a mixed copula approach. *Applied financial economics*, 16(10):717–729.

Patton, A. (2013). Copula methods for forecasting multivariate time series. In *Handbook of economic forecasting*, volume 2, pages 899–960. Elsevier.

Patton, A. J. (2012). A review of copula models for economic time series. *Journal of Multivariate Analysis*, 110:4–18.

Poulin, A., Huard, D., Favre, A.-C., and Pugin, S. (2007). Importance of tail dependence in bivariate frequency analysis. *Journal of Hydrologic Engineering*, 12(4):394–403.

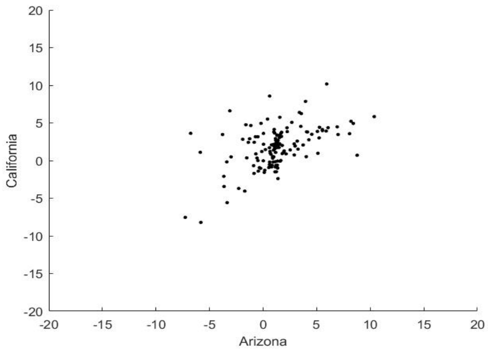
Salvadori, G., Durante, F., De Michele, C., Bernardi, M., and Petrella, L. (2016). A multivariate copula-based framework for dealing with hazard scenarios and failure probabilities. *Water Resources Research*, 52(5):3701–3721.

Sklar, A. (1959). Fonction de répartition dont les marges sont données. *Inst. Stat. Univ. Paris*, 8:229–231.

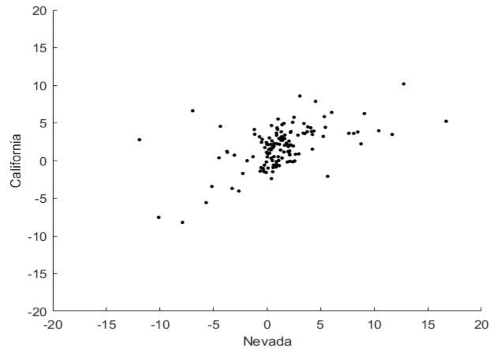
Vuong, Q. H. (1989). Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica: Journal of the Econometric Society*, pages 307–333.

Zimmer, D. M. (2012). The role of copulas in the housing crisis. *Review of Economics and Statistics*, 94(2):607–620.

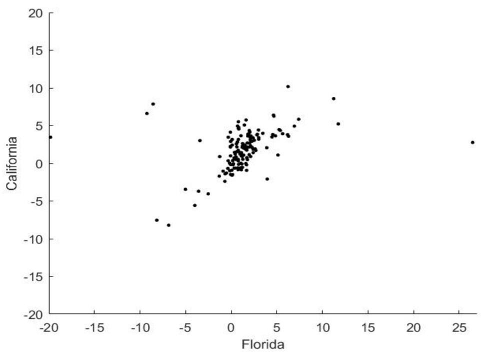
# A Pairwise scatter plots of the used data



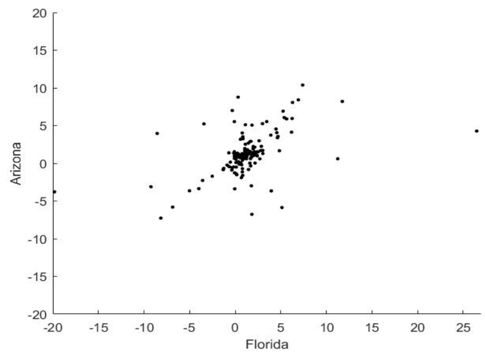

(a) Arizona vs California.



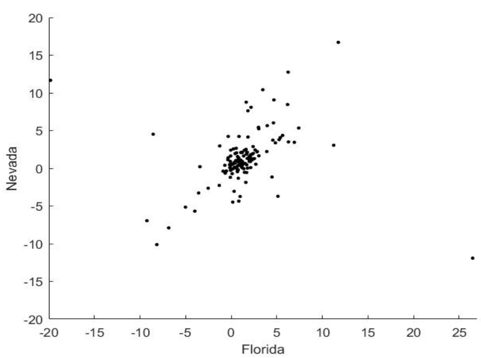

(b) Nevada vs California.



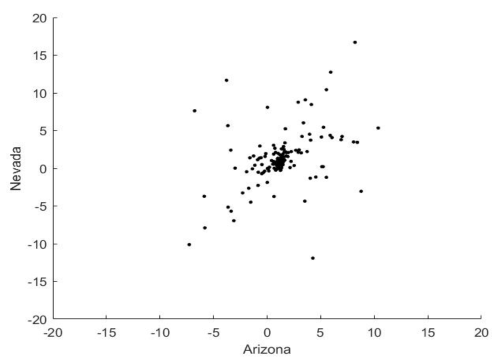

(c) Florida vs California.




(d) Florida vs Arizona.




(e) Florida vs Nevada.




(f) Arizona vs Nevada.

Figure 1: Scatter plots of quarterly percentage changes in the HPI for the different state pairs from the second quarter of 1975 up to the first quarter of 2009.

# B Survival Clayton-survival Gumbel copula

As mentioned in Section 5.2 the survival Clayton-survival Gumbel copula is evaluated for the sake of completeness. Remember that this copula is expected to perform somewhat worse than the Clayton-Gumbel one. The selection criteria and estimates of dependence are displayed in Table 7.

Table 7: Selection criteria and estimates of dependence for the survival Clayton-survival Gumbel copula.

| | AZ-CA | AZ-FL | AZ-NV | CA-FL | CA-NV | FL-NV |
|---|---|---|---|---|---|---|
| BIC | 721.94* | 701.60 | 706.92 | 746.24* | 737.84* | 724.53 |
| Vuong | 0.99 | 0.31 | 0.03 | 1.46 | 1.16 | -0.31 |
| survival Clayton $\tau$ | 0.14 (0.17) | 0.22 (0.25) | 0.70 (0.28) | 0.15 (0.17) | 0.23 (0.23) | 0.40 (0.23) |
| survival Gumbel $\tau$ | 0.52 (0.09) | 0.58 (0.20) | 0.35 (0.20) | 0.37 (0.15) | 0.40 (0.14) | 0.33 (0.17) |
| Lower-tail dependence | 0.61 (0.09) | 0.66 (0.22) | 0.43 (0.21) | 0.45 (0.15) | 0.48 (0.14) | 0.41 (0.18) |
| Upper-tail dependence | 0.11 (0.24) | 0.30 (0.34) | 0.86 (0.37) | 0.15 (0.25) | 0.31 (0.31) | 0.60 (0.32) |
| Proportion survival Clayton ($\hat{w}$) | 0.32 (0.16) | 0.38 (0.16) | 0.23 (0.16) | 0.32 (0.20) | 0.28 (0.21) | 0.39 (0.19) |

Note: * indicates that the BIC value is smaller than in any of the other considered mixtures (Clayton-Gumbel, Clayton-survival Clayton and Gumbel-survival Gumbel). The standard deviation of the estimates is given in parenthesis.

In Table 7 the BIC value indicates that in 3 out of 6 cases the survival Clayton-survival Gumbel copula outperforms all the other copula mixtures (Clayton-Gumbel, Clayton-survival Clayton and Gumbel-survival Gumbel) for which the values can be found in Table 4. This result is remarkable considering that the captured lower- and upper-tail dependence is in general lower for the survival Clayton-survival Gumbel copula than in the other copula mixtures displayed in Table 6. In addition, it is clear from the estimation that lower-tail dependence is higher valued than upper-tail due to the larger weight given to the survival-Gumbel part, just as in the Gumbel-survival Gumbel mixture which does this for 5 out of 6 state pairs. The other two mixtures favour upper-tail dependence in 5 out of 6 cases. Also this fact makes it seem extraordinary that the performance of the survival Clayton-survival Gumbel mixture does not significantly differ in describing the data from the Clayton-Gumbel copula, which can be concluded from the Vuong test statistic. Though, note the word ‘seem’, which is used because the standard errors are large enough to make these results insignificant. Hence the conclusion from the Vuong test statistic is in line with the obtained estimates of dependence, namely the survival Clayton-survival Gumbel copula mixture performs the same in describing the data as the Clayton-Gumbel mixture.

# C Eviews Code

## General code

```
'Set the sample the same as in Zimmer (2012)'
smpl 1975Q2 2009Q1

'Creating the quarterly percentage changes for the HPI index'
series az_percent = (az_index - az_index(-1)) / az_index(-1) * 100
series ca_percent = (ca_index - ca_index(-1)) / ca_index(-1) * 100
series fl_percent = (fl_index - fl_index(-1)) / fl_index(-1) * 100
series nv_percent = (nv_index - nv_index(-1)) / nv_index(-1) * 100

'Check whether autocorrelation is present in the data'
az_percent.correl(20)
ca_percent.correl(20)
fl_percent.correl(20)
nv_percent.correl(20)

'Since autocorrelation is present, an AR(1)-GARCH(1,1) model is modelled. '
'The corresponding conditional variance of the error term are calculated. Then the
    eventual corrected filtered price changes are determined'

'Arizona'
equation eqAZ.arch az_percent az_percent(-1) AR(1)
series az_resid = resid
eqAZ.makegarch AZcvar  'retrieve the conditional variance'
series az_filter = (az_resid) / @sqrt(AZcvar)

'California'
equation eqCA.arch ca_percent ca_percent(-1) AR(1)
series ca_resid = resid
eqAZ.makegarch CAcvar  'retrieve the conditional variance'
series ca_filter = (ca_resid) / @sqrt(CAcvar)

'Florida'
equation eqFL.arch fl_percent fl_percent(-1) AR(1)
```

```
series fl_resid = resid
eqFL.makegarch FLcvar  'retrieve the conditional variance'
series fl_filter = (fl_resid) / @sqrt(FLcvar)

'Nevada'
equation eqNV.arch nv_percent nv_percent(-1) AR(1)
series nv_resid = resid
eqNV.makegarch NVcvar 'retrieve the conditional variance'
series nv_filter = (nv_resid) / @sqrt(NVcvar)

'Show the correlogram up to 20 lags to see whether autocorrelation is present'
az_filter.correl(20)
ca_filter.correl(20)
fl_filter.correl(20)
nv_filter.correl(20)

'Formal autocorrelation test: Breusch Godfrey'
ls az_filter c az_filter(-1)
series zzz = resid
equation zzz1.ls zzz c zzz(-1) az_filter(-1)
scalar zzz21=zzz1.@r2 * 132 'Since 132 observations are left in the estimation'

ls ca_filter c ca_filter(-1)
series zzz = resid
equation zzz1.ls zzz c zzz(-1) ca_filter(-1)
scalar zzz22=zzz1.@r2 * 132 'Since 132 observations are left in the estimation'

ls fl_filter c fl_filter(-1)
series zzz = resid
equation zzz1.ls zzz c zzz(-1) fl_filter(-1)
scalar zzz23=zzz1.@r2 * 132 'Since 132 observations are left in the estimation'

ls nv_filter c nv_filter(-1)
series zzz = resid
equation zzz1.ls zzz c zzz(-1) nv_filter(-1)
scalar zzz24=zzz1.@r2 * 132 'Since 132 observations are left in the estimation'
```

```
'corresponding probabilities'
scalar zzzprob21 = @chisq(zzz21,1)
scalar zzzprob22 = @chisq(zzz22,1)
scalar zzzprob23 = @chisq(zzz23,1)
scalar zzzprob24 = @chisq(zzz24,1)

'ARCH LM test to test for autoregressive conditional heteroskedasticity up to order 1 in
    the AR(1)-GARCH(1,1) estimation'
eqaz.archtest(1)
eqca.archtest(1)
eqfl.archtest(1)
eqnv.archtest(1)

'To test for normality, click the filtered series -> view -> Descriptive statistics and
    tests -> Stats table, to see the Jarque-Bera test statistic'

'CDFs of the marginals'
series cdf_az = @cnorm(az_filter)
series cdf_ca = @cnorm(ca_filter)
series cdf_fl = @cnorm(fl_filter)
series cdf_nv = @cnorm(nv_filter)

'PDFs of the marginals'
series pdf_az = @dnorm(az_filter)
series pdf_ca = @dnorm(ca_filter)
series pdf_fl = @dnorm(fl_filter)
series pdf_nv = @dnorm(nv_filter)

'-------------------------------------------------------------------------------------'
'This program calculates the starting values for the mixture copulas for each state pair.
     Thus (survival) Clayton and (suvival) Gumbel will be evaluated. First a maximum
    likelihood object is created, then the desired sample size is selected (amount of
    observations) and maximum likelihood is performed.'

'note in the Clayton and survival Clayton ((c(1)^2) makes sure that theta will be between
     0 and infinity. In the Gumbel (exp(c(1))+1) creates a theta between 1 and infinity
    and in the survival Gumbel (exp(c(1))/(exp(c(1))+1) makes theta between 0 and 1'
```

```
'clayton AZ-CA'
logl logl_clayton_az_ca
logl_clayton_az_ca.append @logl logl11
logl_clayton_az_ca.append @param c(1) 0.6
logl_clayton_az_ca.append f1 = pdf_az
logl_clayton_az_ca.append f2 = pdf_ca
logl_clayton_az_ca.append term1 = ((c(1)^2)+1)*(cdf_az*cdf_ca)^(-(c(1)^2)-1) * ( cdf_ca
     ^(-(c(1)^2)) + cdf_az^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2)*f1*f2
logl_clayton_az_ca.append logl11 = log(term1)
smpl 1976q1 2009q1
logl_clayton_az_ca.ml(showstart)
scalar clayton_az_ca = logl_clayton_az_ca.@coef(1)

'clayton AZ-FL'
logl logl_clayton_az_fl
logl_clayton_az_fl.append @logl logl12
logl_clayton_az_fl.append @param c(1) 0.6
logl_clayton_az_fl.append f1 = pdf_az
logl_clayton_az_fl.append f2 = pdf_fl
logl_clayton_az_fl.append term1 = ((c(1)^2)+1)*(cdf_az*cdf_fl)^(-(c(1)^2)-1) * ( cdf_fl
     ^(-(c(1)^2)) + cdf_az^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2)*f1*f2
logl_clayton_az_fl.append logl12 = log(term1)
smpl 1976q1 2009q1
logl_clayton_az_fl.ml(showstart)
scalar clayton_az_fl = logl_clayton_az_fl.@coef(1)

'clayton AZ-NV'
logl logl_clayton_az_nv
logl_clayton_az_nv.append @logl logl13
logl_clayton_az_nv.append @param c(1) 0.6
logl_clayton_az_nv.append f1 = pdf_az
logl_clayton_az_nv.append f2 = pdf_nv
logl_clayton_az_nv.append term1 = ((c(1)^2)+1)*(cdf_az*cdf_nv)^(-(c(1)^2)-1) * ( cdf_nv
     ^(-(c(1)^2)) + cdf_az^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2)*f1*f2
logl_clayton_az_nv.append logl13 = log(term1)
smpl 1976q1 2009q1
```

```
logl_clayton_az_nv.ml(showstart)
scalar clayton_az_nv = logl_clayton_az_nv.@coef(1)

'clayton CA-FL'
logl logl_clayton_ca_fl
logl_clayton_ca_fl.append @logl logl14
logl_clayton_ca_fl.append @param c(1) 0.6
logl_clayton_ca_fl.append f1 = pdf_ca
logl_clayton_ca_fl.append f2 = pdf_fl
logl_clayton_ca_fl.append term1 = ((c(1)^2)+1)*(cdf_ca*cdf_fl)^(-(c(1)^2)-1) * ( cdf_fl
     ^(-(c(1)^2)) + cdf_ca^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2)*f1*f2
logl_clayton_ca_fl.append logl14 = log(term1)
smpl 1976q1 2009q1
logl_clayton_ca_fl.ml(showstart)
scalar clayton_ca_fl = logl_clayton_ca_fl.@coef(1)

'clayton CA-NV'
logl logl_clayton_ca_nv
logl_clayton_ca_nv.append @logl logl15
logl_clayton_ca_nv.append @param c(1) 0.6
logl_clayton_ca_nv.append f1 = pdf_ca
logl_clayton_ca_nv.append f2 = pdf_nv
logl_clayton_ca_nv.append term1 = ((c(1)^2)+1)*(cdf_ca*cdf_nv)^(-(c(1)^2)-1) * ( cdf_nv
     ^(-(c(1)^2)) + cdf_ca^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2)*f1*f2
logl_clayton_ca_nv.append logl15 = log(term1)
smpl 1976q1 2009q1
logl_clayton_ca_nv.ml(showstart)
scalar clayton_ca_nv = logl_clayton_ca_nv.@coef(1)

'clayton FL-NV'
logl logl_clayton_fl_nv
logl_clayton_fl_nv.append @logl logl16
logl_clayton_fl_nv.append @param c(1) 0.6
logl_clayton_fl_nv.append f1 = pdf_fl
logl_clayton_fl_nv.append f2 = pdf_nv
logl_clayton_fl_nv.append term1 = ((c(1)^2)+1)*(cdf_fl*cdf_nv)^(-(c(1)^2)-1) * ( cdf_nv
     ^(-(c(1)^2)) + cdf_fl^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2)*f1*f2
```

```
logl_clayton_fl_nv.append logl16 = log(term1)
smpl 1976q1 2009q1
logl_clayton_fl_nv.ml(showstart)
scalar clayton_fl_nv = logl_clayton_fl_nv.@coef(1)

'gumbel AZ-CA'
logl logl_gumbel_az_ca
logl_gumbel_az_ca.append @logl logl21
logl_gumbel_az_ca.append @param c(1) 1.1
logl_gumbel_az_ca.append f1=pdf_az
logl_gumbel_az_ca.append f2= pdf_ca
logl_gumbel_az_ca.append term1 = exp(-((-log(cdf_az))^(exp(c(1))+1)+(-log(cdf_ca))^(exp(c
    (1))+1))^(1/(exp(c(1))+1))) * (cdf_az*cdf_ca)^-1 * (-log(cdf_az)*-log(cdf_ca))^((exp(
    c(1))+1)-1) * ((-log(cdf_az))^(exp(c(1))+1) + (-log(cdf_ca))^(exp(c(1))+1))^(-2+1/(
    exp(c(1))+1)) * (((-log(cdf_az))^(exp(c(1))+1)+(-log(cdf_ca))^(exp(c(1))+1))^(1/(exp(
    c(1))+1))+(exp(c(1))+1)-1) *f1 *f2
logl_gumbel_az_ca.append logl21=log(term1)
smpl 1976q1 2009q1
logl_gumbel_az_ca.ml(showstart)
scalar gumbel_az_ca = logl_gumbel_az_ca.@coef(1)

'gumbel AZ-FL'
logl logl_gumbel_az_fl
logl_gumbel_az_fl.append @logl logl22
logl_gumbel_az_fl.append @param c(1) 1.1
logl_gumbel_az_fl.append f1=pdf_az
logl_gumbel_az_fl.append f2= pdf_fl
logl_gumbel_az_fl.append term1 = exp(-((-log(cdf_az))^(exp(c(1))+1)+(-log(cdf_fl))^(exp(c
    (1))+1))^(1/(exp(c(1))+1))) * (cdf_az*cdf_fl)^-1 * (-log(cdf_az)*-log(cdf_fl))^((exp(
    c(1))+1)-1) * ((-log(cdf_az))^(exp(c(1))+1) + (-log(cdf_fl))^(exp(c(1))+1))^(-2+1/(
    exp(c(1))+1)) * (((-log(cdf_az))^(exp(c(1))+1)+(-log(cdf_fl))^(exp(c(1))+1))^(1/(exp(
    c(1))+1))+(exp(c(1))+1)-1) *f1 *f2
logl_gumbel_az_fl.append logl22=log(term1)
smpl 1976q1 2009q1
logl_gumbel_az_fl.ml(showstart)
scalar gumbel_az_fl = logl_gumbel_az_fl.@coef(1)
```

```
'gumbel AZ-NV'
logl logl_gumbel_az_nv
logl_gumbel_az_nv.append @logl logl23
logl_gumbel_az_nv.append @param c(1) 1.1
logl_gumbel_az_nv.append f1=pdf_az
logl_gumbel_az_nv.append f2= pdf_nv
logl_gumbel_az_nv.append term1 = exp(-((-log(cdf_az))^(exp(c(1))+1)+(-log(cdf_nv))^(exp(c
    (1))+1))^(1/(exp(c(1))+1))) * (cdf_az*cdf_nv)^-1 * (-log(cdf_az)*-log(cdf_nv))^((exp(
    c(1))+1)-1) * ((-log(cdf_az))^(exp(c(1))+1) + (-log(cdf_nv))^(exp(c(1))+1))^(-2+1/(
    exp(c(1))+1)) * (((-log(cdf_az))^(exp(c(1))+1)+(-log(cdf_nv))^(exp(c(1))+1))^(1/(exp(
    c(1))+1))+(exp(c(1))+1)-1) *f1 *f2
logl_gumbel_az_nv.append logl23=log(term1)
smpl 1976q1 2009q1
logl_gumbel_az_nv.ml(showstart)
scalar gumbel_az_nv = logl_gumbel_az_nv.@coef(1)


'gumbel CA-FL'
logl logl_gumbel_ca_fl
logl_gumbel_ca_fl.append @logl logl24
logl_gumbel_ca_fl.append @param c(1) 1.1
logl_gumbel_ca_fl.append f1=pdf_ca
logl_gumbel_ca_fl.append f2= pdf_fl
logl_gumbel_ca_fl.append term1 = exp(-((-log(cdf_ca))^(exp(c(1))+1)+(-log(cdf_fl))^(exp(c
    (1))+1))^(1/(exp(c(1))+1))) * (cdf_ca*cdf_fl)^-1 * (-log(cdf_ca)*-log(cdf_fl))^((exp(
    c(1))+1)-1) * ((-log(cdf_ca))^(exp(c(1))+1) + (-log(cdf_fl))^(exp(c(1))+1))^(-2+1/(
    exp(c(1))+1)) * (((-log(cdf_ca))^(exp(c(1))+1)+(-log(cdf_fl))^(exp(c(1))+1))^(1/(exp(
    c(1))+1))+(exp(c(1))+1)-1) *f1 *f2
logl_gumbel_ca_fl.append logl24=log(term1)
smpl 1976q1 2009q1
logl_gumbel_ca_fl.ml(showstart)
scalar gumbel_ca_fl = logl_gumbel_ca_fl.@coef(1)

'gumbel CA-NV'
logl logl_gumbel_ca_nv
logl_gumbel_ca_nv.append @logl logl25
logl_gumbel_ca_nv.append @param c(1) 1.1
```

```
logl_gumbel_ca_nv.append f1=pdf_ca
logl_gumbel_ca_nv.append f2= pdf_nv
logl_gumbel_ca_nv.append term1 = exp(-((-log(cdf_ca))^(exp(c(1))+1)+(-log(cdf_nv))^(exp(c
    (1))+1))^(1/(exp(c(1))+1))) * (cdf_ca*cdf_nv)^-1 * (-log(cdf_ca)*-log(cdf_nv))^((exp(
    c(1))+1)-1) * ((-log(cdf_ca))^(exp(c(1))+1) + (-log(cdf_nv))^(exp(c(1))+1))^(-2+1/(
    exp(c(1))+1)) * (((-log(cdf_ca))^(exp(c(1))+1)+(-log(cdf_nv))^(exp(c(1))+1))^(1/(exp(
    c(1))+1))+(exp(c(1))+1)-1) *f1 *f2
logl_gumbel_ca_nv.append logl25=log(term1)
smpl 1976q1 2009q1
logl_gumbel_ca_nv.ml(showstart)
scalar gumbel_ca_nv = logl_gumbel_ca_nv.@coef(1)

'gumbel FL-NV'
logl logl_gumbel_fl_nv
logl_gumbel_fl_nv.append @logl logl26
logl_gumbel_fl_nv.append @param c(1) 1.1
logl_gumbel_fl_nv.append f1=pdf_fl
logl_gumbel_fl_nv.append f2= pdf_nv
logl_gumbel_fl_nv.append term1 = exp(-((-log(cdf_fl))^(exp(c(1))+1)+(-log(cdf_nv))^(exp(c
    (1))+1))^(1/(exp(c(1))+1))) * (cdf_fl*cdf_nv)^-1 * (-log(cdf_fl)*-log(cdf_nv))^((exp(
    c(1))+1)-1) * ((-log(cdf_fl))^(exp(c(1))+1) + (-log(cdf_nv))^(exp(c(1))+1))^(-2+1/(
    exp(c(1))+1)) * (((-log(cdf_fl))^(exp(c(1))+1)+(-log(cdf_nv))^(exp(c(1))+1))^(1/(exp(
    c(1))+1))+(exp(c(1))+1)-1) *f1 *f2
logl_gumbel_fl_nv.append logl26=log(term1)
smpl 1976q1 2009q1
logl_gumbel_fl_nv.ml(showstart)
scalar gumbel_fl_nv = logl_gumbel_fl_nv.@coef(1)

'survival clayton AZ-CA'
logl logl_survival_clayton_az_ca
logl_survival_clayton_az_ca.append @logl logl41
logl_survival_clayton_az_ca.append @param c(1) 0.6
logl_survival_clayton_az_ca.append f1=pdf_az
logl_survival_clayton_az_ca.append f2=pdf_ca
logl_survival_clayton_az_ca.append term1 = (c(1)^2)*(1/(c(1)^2)+1) * ((1-cdf_az)*(1-
    cdf_ca))^(-(c(1)^2)-1) * ( (1-cdf_az)^(-(c(1)^2)) + (1-cdf_ca)^(-(c(1)^2))-1 )^(-1/(c
    (1)^2)-2) *f1*f2
```

```
logl_survival_clayton_az_ca.append logl41=log(term1)
smpl 1976q1 2009q1
logl_survival_clayton_az_ca.ml(showstart)
scalar survival_clayton_az_ca = logl_survival_clayton_az_ca.@coef(1)

'survival clayton AZ-FL'
logl logl_survival_clayton_az_fl
logl_survival_clayton_az_fl.append @logl logl42
logl_survival_clayton_az_fl.append @param c(1) 0.6
logl_survival_clayton_az_fl.append f1=pdf_az
logl_survival_clayton_az_fl.append f2=pdf_fl
logl_survival_clayton_az_fl.append term1 = (c(1)^2)*(1/(c(1)^2)+1) * ((1-cdf_az)*(1-
    cdf_fl))^(-(c(1)^2)-1) * ( (1-cdf_az)^(-(c(1)^2)) + (1-cdf_fl)^(-(c(1)^2))-1 )^(-1/(c
    (1)^2)-2) *f1*f2
logl_survival_clayton_az_fl.append logl42=log(term1)
smpl 1976q1 2009q1
logl_survival_clayton_az_fl.ml(showstart)
scalar survival_clayton_az_fl = logl_survival_clayton_az_fl.@coef(1)

'survival clayton AZ-NV'
logl logl_survival_clayton_az_nv
logl_survival_clayton_az_nv.append @logl logl43
logl_survival_clayton_az_nv.append @param c(1) 0.6
logl_survival_clayton_az_nv.append f1=pdf_az
logl_survival_clayton_az_nv.append f2=pdf_nv
logl_survival_clayton_az_nv.append term1 = (c(1)^2)*(1/(c(1)^2)+1) * ((1-cdf_az)*(1-
    cdf_nv))^(-(c(1)^2)-1) * ( (1-cdf_az)^(-(c(1)^2)) + (1-cdf_nv)^(-(c(1)^2))-1 )^(-1/(c
    (1)^2)-2) *f1*f2
logl_survival_clayton_az_nv.append logl43=log(term1)
smpl 1976q1 2009q1
logl_survival_clayton_az_nv.ml(showstart)
scalar survival_clayton_az_nv = logl_survival_clayton_az_nv.@coef(1)

'survival clayton CA-FL'
logl logl_survival_clayton_ca_fl
logl_survival_clayton_ca_fl.append @logl logl44
logl_survival_clayton_ca_fl.append @param c(1) 0.6
```

```
logl_survival_clayton_ca_fl.append f1=pdf_ca
logl_survival_clayton_ca_fl.append f2=pdf_fl
logl_survival_clayton_ca_fl.append term1 = (c(1)^2)*(1/(c(1)^2)+1) * ((1-cdf_ca)*(1-
    cdf_fl))^(-(c(1)^2)-1) * ( (1-cdf_ca)^(-(c(1)^2)) + (1-cdf_fl)^(-(c(1)^2))-1 )^(-1/(c
    (1)^2)-2) *f1*f2
logl_survival_clayton_ca_fl.append logl44=log(term1)
smpl 1976q1 2009q1
logl_survival_clayton_ca_fl.ml(showstart)
scalar survival_clayton_ca_fl = logl_survival_clayton_ca_fl.@coef(1)

'survival clayton CA-NV'
logl logl_survival_clayton_ca_nv
logl_survival_clayton_ca_nv.append @logl logl45
logl_survival_clayton_ca_nv.append @param c(1) 0.6
logl_survival_clayton_ca_nv.append f1=pdf_ca
logl_survival_clayton_ca_nv.append f2=pdf_nv
logl_survival_clayton_ca_nv.append term1 = (c(1)^2)*(1/(c(1)^2)+1) * ((1-cdf_ca)*(1-
    cdf_nv))^(-(c(1)^2)-1) * ( (1-cdf_ca)^(-(c(1)^2)) + (1-cdf_nv)^(-(c(1)^2))-1 )^(-1/(c
    (1)^2)-2) *f1*f2
logl_survival_clayton_ca_nv.append logl45=log(term1)
smpl 1976q1 2009q1
logl_survival_clayton_ca_nv.ml(showstart)
scalar survival_clayton_ca_nv = logl_survival_clayton_ca_nv.@coef(1)

'survival clayton FL-NV'
logl logl_survival_clayton_fl_nv
logl_survival_clayton_fl_nv.append @logl logl46
logl_survival_clayton_fl_nv.append @param c(1) 0.6
logl_survival_clayton_fl_nv.append f1=pdf_fl
logl_survival_clayton_fl_nv.append f2=pdf_nv
logl_survival_clayton_fl_nv.append term1 = (c(1)^2)*(1/(c(1)^2)+1) * ((1-cdf_fl)*(1-
    cdf_nv))^(-(c(1)^2)-1) * ( (1-cdf_fl)^(-(c(1)^2)) + (1-cdf_nv)^(-(c(1)^2))-1 )^(-1/(c
    (1)^2)-2) *f1*f2
logl_survival_clayton_fl_nv.append logl46=log(term1)
smpl 1976q1 2009q1
logl_survival_clayton_fl_nv.ml(showstart)
scalar survival_clayton_fl_nv = logl_survival_clayton_fl_nv.@coef(1)
```

```
'survival gumbel AZ-CA'
logl logl_survival_gumbel_az_ca
logl_survival_gumbel_az_ca.append @logl logl61
logl_survival_gumbel_az_ca.append @param c(1) 0.6
logl_survival_gumbel_az_ca.append f1=pdf_az
logl_survival_gumbel_az_ca.append f2=pdf_ca
logl_survival_gumbel_az_ca.append term1 = ((exp(-((-log(1 - cdf_az))^(1/(exp(c(1))/(exp(c
    (1))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(1))/(exp(c(1))+1))))^(exp(c(1))/(exp(c(1))
    +1)))*((-log(1 - cdf_az))^(1/(exp(c(1))/(exp(c(1))+1))) + (-log(1 - cdf_ca))^(1/(exp(
    c(1))/(exp(c(1))+1))))^(2*(exp(c(1))/(exp(c(1))+1)) - 2)*(-log(1 - cdf_az))^(1/(exp(c
    (1))/(exp(c(1))+1)) - 1)*(-log(1 - cdf_ca))^(1/(exp(c(1))/(exp(c(1))+1)) - 1))/((
    cdf_az - 1)*(cdf_ca - 1)) - (exp(-((-log(1 - cdf_az))^(1/(exp(c(1))/(exp(c(1))+1))) +
     (-log(1 - cdf_ca))^(1/(exp(c(1))/(exp(c(1))+1))))^(exp(c(1))/(exp(c(1))+1)))*((exp(c
    (1))/(exp(c(1))+1)) - 1)*((-log(1 - cdf_az))^(1/(exp(c(1))/(exp(c(1))+1))) + (-log(1
    - cdf_ca))^(1/(exp(c(1))/(exp(c(1))+1))))^((exp(c(1))/(exp(c(1))+1)) - 2)*(-log(1 -
    cdf_az))^(1/(exp(c(1))/(exp(c(1))+1)) - 1)*(-log(1 - cdf_ca))^(1/(exp(c(1))/(exp(c(1)
    )+1)) - 1))/((exp(c(1))/(exp(c(1))+1))*(cdf_az - 1)*(cdf_ca - 1)) )*f1*f2
logl_survival_gumbel_az_ca.append logl61=log(term1)
smpl 1976q1 2009q1
logl_survival_gumbel_az_ca.ml(showstart)
scalar survival_gumbel_az_ca = logl_survival_gumbel_az_ca.@coef(1)

'survival gumbel AZ-FL'
logl logl_survival_gumbel_az_fl
logl_survival_gumbel_az_fl.append @logl logl62
logl_survival_gumbel_az_fl.append @param c(1) 0.6
logl_survival_gumbel_az_fl.append f1=pdf_az
logl_survival_gumbel_az_fl.append f2=pdf_fl
logl_survival_gumbel_az_fl.append term1 = ((exp(-((-log(1 - cdf_az))^(1/(exp(c(1))/(exp(c
    (1))+1))) + (-log(1 - cdf_fl))^(1/(exp(c(1))/(exp(c(1))+1))))^(exp(c(1))/(exp(c(1))
    +1)))*((-log(1 - cdf_az))^(1/(exp(c(1))/(exp(c(1))+1))) + (-log(1 - cdf_fl))^(1/(exp(
    c(1))/(exp(c(1))+1))))^(2*(exp(c(1))/(exp(c(1))+1)) - 2)*(-log(1 - cdf_az))^(1/(exp(c
    (1))/(exp(c(1))+1)) - 1)*(-log(1 - cdf_fl))^(1/(exp(c(1))/(exp(c(1))+1)) - 1))/((
    cdf_az - 1)*(cdf_fl - 1)) - (exp(-((-log(1 - cdf_az))^(1/(exp(c(1))/(exp(c(1))+1))) +
     (-log(1 - cdf_fl))^(1/(exp(c(1))/(exp(c(1))+1))))^(exp(c(1))/(exp(c(1))+1)))*((exp(c
    (1))/(exp(c(1))+1)) - 1)*((-log(1 - cdf_az))^(1/(exp(c(1))/(exp(c(1))+1))) + (-log(1
```

```
    - cdf_fl))^(1/(exp(c(1))/(exp(c(1))+1))))^((exp(c(1))/(exp(c(1))+1)) - 2)*(-log(1 -
    cdf_az))^(1/(exp(c(1))/(exp(c(1))+1)) - 1)*(-log(1 - cdf_fl))^(1/(exp(c(1))/(exp(c(1)
    )+1)) - 1))/((exp(c(1))/(exp(c(1))+1))*(cdf_az - 1)*(cdf_fl - 1)) )*f1*f2
logl_survival_gumbel_az_fl.append logl62=log(term1)
smpl 1976q1 2009q1
logl_survival_gumbel_az_fl.ml(showstart)
scalar survival_gumbel_az_fl = logl_survival_gumbel_az_fl.@coef(1)

'survival gumbel AZ-NV'
logl logl_survival_gumbel_az_nv
logl_survival_gumbel_az_nv.append @logl logl63
logl_survival_gumbel_az_nv.append @param c(1) 0.6
logl_survival_gumbel_az_nv.append f1=pdf_az
logl_survival_gumbel_az_nv.append f2=pdf_nv
logl_survival_gumbel_az_nv.append term1 = ((exp(-((-log(1 - cdf_az))^(1/(exp(c(1))/(exp(c
    (1))+1))) + (-log(1 - cdf_nv))^(1/(exp(c(1))/(exp(c(1))+1))))^(exp(c(1))/(exp(c(1))
    +1)))*((-log(1 - cdf_az))^(1/(exp(c(1))/(exp(c(1))+1))) + (-log(1 - cdf_nv))^(1/(exp(
    c(1))/(exp(c(1))+1))))^(2*(exp(c(1))/(exp(c(1))+1)) - 2)*(-log(1 - cdf_az))^(1/(exp(c
    (1))/(exp(c(1))+1)) - 1)*(-log(1 - cdf_nv))^(1/(exp(c(1))/(exp(c(1))+1)) - 1))/((
    cdf_az - 1)*(cdf_nv - 1)) - (exp(-((-log(1 - cdf_az))^(1/(exp(c(1))/(exp(c(1))+1))) +
     (-log(1 - cdf_nv))^(1/(exp(c(1))/(exp(c(1))+1))))^(exp(c(1))/(exp(c(1))+1)))*((exp(c
    (1))/(exp(c(1))+1)) - 1)*((-log(1 - cdf_az))^(1/(exp(c(1))/(exp(c(1))+1))) + (-log(1
    - cdf_nv))^(1/(exp(c(1))/(exp(c(1))+1))))^((exp(c(1))/(exp(c(1))+1)) - 2)*(-log(1 -
    cdf_az))^(1/(exp(c(1))/(exp(c(1))+1)) - 1)*(-log(1 - cdf_nv))^(1/(exp(c(1))/(exp(c(1)
    )+1)) - 1))/((exp(c(1))/(exp(c(1))+1))*(cdf_az - 1)*(cdf_nv - 1)) )*f1*f2
logl_survival_gumbel_az_nv.append logl63=log(term1)
smpl 1976q1 2009q1
logl_survival_gumbel_az_nv.ml(showstart)
scalar survival_gumbel_az_nv = logl_survival_gumbel_az_nv.@coef(1)

'survival gumbel CA-FL'
logl logl_survival_gumbel_ca_fl
logl_survival_gumbel_ca_fl.append @logl logl64
logl_survival_gumbel_ca_fl.append @param c(1) 0.6
logl_survival_gumbel_ca_fl.append f1=pdf_ca
logl_survival_gumbel_ca_fl.append f2=pdf_fl
```

```
logl_survival_gumbel_ca_fl.append term1 = ((exp(-((-log(1 - cdf_ca))^(1/(exp(c(1))/(exp(c
    (1))+1))) + (-log(1 - cdf_fl))^(1/(exp(c(1))/(exp(c(1))+1))))^(exp(c(1))/(exp(c(1))
    +1)))*((-log(1 - cdf_ca))^(1/(exp(c(1))/(exp(c(1))+1))) + (-log(1 - cdf_fl))^(1/(exp(
    c(1))/(exp(c(1))+1))))^(2*(exp(c(1))/(exp(c(1))+1)) - 2)*(-log(1 - cdf_ca))^(1/(exp(c
    (1))/(exp(c(1))+1)) - 1)*(-log(1 - cdf_fl))^(1/(exp(c(1))/(exp(c(1))+1)) - 1))/((
    cdf_ca - 1)*(cdf_fl - 1)) - (exp(-((-log(1 - cdf_ca))^(1/(exp(c(1))/(exp(c(1))+1))) +
     (-log(1 - cdf_fl))^(1/(exp(c(1))/(exp(c(1))+1))))^(exp(c(1))/(exp(c(1))+1)))*((exp(c
    (1))/(exp(c(1))+1)) - 1)*((-log(1 - cdf_ca))^(1/(exp(c(1))/(exp(c(1))+1))) + (-log(1
    - cdf_fl))^(1/(exp(c(1))/(exp(c(1))+1))))^((exp(c(1))/(exp(c(1))+1)) - 2)*(-log(1 -
    cdf_ca))^(1/(exp(c(1))/(exp(c(1))+1)) - 1)*(-log(1 - cdf_fl))^(1/(exp(c(1))/(exp(c(1)
    )+1)) - 1))/((exp(c(1))/(exp(c(1))+1))*(cdf_ca - 1)*(cdf_fl - 1)) )*f1*f2
logl_survival_gumbel_ca_fl.append logl64=log(term1)
smpl 1976q1 2009q1
logl_survival_gumbel_ca_fl.ml(showstart)
scalar survival_gumbel_ca_fl = logl_survival_gumbel_ca_fl.@coef(1)

'survival gumbel CA-NV'
logl logl_survival_gumbel_ca_nv
logl_survival_gumbel_ca_nv.append @logl logl65
logl_survival_gumbel_ca_nv.append @param c(1) 0.6
logl_survival_gumbel_ca_nv.append f1=pdf_ca
logl_survival_gumbel_ca_nv.append f2=pdf_nv
logl_survival_gumbel_ca_nv.append term1 = ((exp(-((-log(1 - cdf_ca))^(1/(exp(c(1))/(exp(c
    (1))+1))) + (-log(1 - cdf_nv))^(1/(exp(c(1))/(exp(c(1))+1))))^(exp(c(1))/(exp(c(1))
    +1)))*((-log(1 - cdf_ca))^(1/(exp(c(1))/(exp(c(1))+1))) + (-log(1 - cdf_nv))^(1/(exp(
    c(1))/(exp(c(1))+1))))^(2*(exp(c(1))/(exp(c(1))+1)) - 2)*(-log(1 - cdf_ca))^(1/(exp(c
    (1))/(exp(c(1))+1)) - 1)*(-log(1 - cdf_nv))^(1/(exp(c(1))/(exp(c(1))+1)) - 1))/((
    cdf_ca - 1)*(cdf_nv - 1)) - (exp(-((-log(1 - cdf_ca))^(1/(exp(c(1))/(exp(c(1))+1))) +
     (-log(1 - cdf_nv))^(1/(exp(c(1))/(exp(c(1))+1))))^(exp(c(1))/(exp(c(1))+1)))*((exp(c
    (1))/(exp(c(1))+1)) - 1)*((-log(1 - cdf_ca))^(1/(exp(c(1))/(exp(c(1))+1))) + (-log(1
    - cdf_nv))^(1/(exp(c(1))/(exp(c(1))+1))))^((exp(c(1))/(exp(c(1))+1)) - 2)*(-log(1 -
    cdf_ca))^(1/(exp(c(1))/(exp(c(1))+1)) - 1)*(-log(1 - cdf_nv))^(1/(exp(c(1))/(exp(c(1)
    )+1)) - 1))/((exp(c(1))/(exp(c(1))+1))*(cdf_ca - 1)*(cdf_nv - 1)) )*f1*f2
logl_survival_gumbel_ca_nv.append logl65=log(term1)
smpl 1976q1 2009q1
logl_survival_gumbel_ca_nv.ml(showstart)
scalar survival_gumbel_ca_nv = logl_survival_gumbel_ca_nv.@coef(1)
```

```
'survival gumbel FL-NV'
logl logl_survival_gumbel_fl_nv
logl_survival_gumbel_fl_nv.append @logl logl66
logl_survival_gumbel_fl_nv.append @param c(1) 0.6
logl_survival_gumbel_fl_nv.append f1=pdf_fl
logl_survival_gumbel_fl_nv.append f2=pdf_nv
logl_survival_gumbel_fl_nv.append term1 = ((exp(-((-log(1 - cdf_fl))^(1/(exp(c(1))/(exp(c
    (1))+1))) + (-log(1 - cdf_nv))^(1/(exp(c(1))/(exp(c(1))+1))))^(exp(c(1))/(exp(c(1))
    +1)))*((-log(1 - cdf_fl))^(1/(exp(c(1))/(exp(c(1))+1))) + (-log(1 - cdf_nv))^(1/(exp(
    c(1))/(exp(c(1))+1))))^(2*(exp(c(1))/(exp(c(1))+1)) - 2)*(-log(1 - cdf_fl))^(1/(exp(c
    (1))/(exp(c(1))+1)) - 1)*(-log(1 - cdf_nv))^(1/(exp(c(1))/(exp(c(1))+1)) - 1))/((
    cdf_fl - 1)*(cdf_nv - 1)) - (exp(-((-log(1 - cdf_fl))^(1/(exp(c(1))/(exp(c(1))+1))) +
     (-log(1 - cdf_nv))^(1/(exp(c(1))/(exp(c(1))+1))))^(exp(c(1))/(exp(c(1))+1)))*((exp(c
    (1))/(exp(c(1))+1)) - 1)*((-log(1 - cdf_fl))^(1/(exp(c(1))/(exp(c(1))+1))) + (-log(1
    - cdf_nv))^(1/(exp(c(1))/(exp(c(1))+1))))^((exp(c(1))/(exp(c(1))+1)) - 2)*(-log(1 -
    cdf_fl))^(1/(exp(c(1))/(exp(c(1))+1)) - 1)*(-log(1 - cdf_nv))^(1/(exp(c(1))/(exp(c(1)
    )+1)) - 1))/((exp(c(1))/(exp(c(1))+1))*(cdf_fl - 1)*(cdf_nv - 1)) )*f1*f2
logl_survival_gumbel_fl_nv.append logl66=log(term1)
smpl 1976q1 2009q1
logl_survival_gumbel_fl_nv.ml(showstart)
scalar survival_gumbel_fl_nv = logl_survival_gumbel_fl_nv.@coef(1)

'-------------------------------------------------------------------------------------'
'This program calculates the restricted coefficients of the mixture copulas for each
    state pair, which are then converted to theta and then to Kendall‘s Tau.'
'to avoid errors in the parameter statement, eviews starts the estimation with the values
     originally present in coefficient vector c, which we will choose according to the
    previous estimations. Note that the weight is set as (exp(c(3))/(1+exp(c(3))) such
    that it will be between 0 and 1'
'-------------------------------------------------------------------------------------'
'Clayton-Gumbel mixture AZ-CA'
c(1) = clayton_az_ca
c(2) = gumbel_az_ca
c(3) = -1

logl logl_Clay_Gum_az_ca
```

```
logl_Clay_Gum_az_ca.append @logl logl31
logl_Clay_Gum_az_ca.append f1=pdf_az
logl_Clay_Gum_az_ca.append f2=pdf_ca
logl_Clay_Gum_az_ca.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_az*cdf_ca)
    ^(-(c(1)^2)-1) * ( cdf_ca^(-(c(1)^2)) + cdf_az^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2) + (1-
    exp(c(3))/(1+exp(c(3)))) * exp(-((-log(cdf_az))^(exp(c(2))+1)+(-log(cdf_ca))^(exp(c
    (2))+1))^(1/(exp(c(2))+1))) * (cdf_az*cdf_ca)^-1 * (-log(cdf_az)*-log(cdf_ca))^((exp(
    c(2))+1)-1) * ((-log(cdf_az))^(exp(c(2))+1) + (-log(cdf_ca))^(exp(c(2))+1))^(-2+1/(
    exp(c(2))+1)) * (((-log(cdf_az))^(exp(c(2))+1)+(-log(cdf_ca))^(exp(c(2))+1))^(1/(exp(
    c(2))+1))+(exp(c(2))+1)-1) )*f1 *f2
logl_Clay_Gum_az_ca.append logl31 = log(term1)
smpl 1976q1 2009q1
logl_Clay_Gum_az_ca.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_Gum_az_ca_CLAY = logl_Clay_Gum_az_ca.@coef(2)
scalar Clay_Gum_az_ca_GUM = logl_Clay_Gum_az_ca.@coef(3)
scalar Clay_Gum_az_ca_PI = logl_Clay_Gum_az_ca.@coef(1)


'Clayton-Gumbel mixture AZ-FL'
c(1) = clayton_az_fl
c(2) = gumbel_az_fl
c(3) = -1

logl logl_Clay_Gum_az_fl
logl_Clay_Gum_az_fl.append @logl logl32
logl_Clay_Gum_az_fl.append f1=pdf_az
logl_Clay_Gum_az_fl.append f2=pdf_fl
logl_Clay_Gum_az_fl.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_az*cdf_fl)
    ^(-(c(1)^2)-1) * ( cdf_fl^(-(c(1)^2)) + cdf_az^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2) + (1-
    exp(c(3))/(1+exp(c(3)))) * exp(-((-log(cdf_az))^(exp(c(2))+1)+(-log(cdf_fl))^(exp(c
    (2))+1))^(1/(exp(c(2))+1))) * (cdf_az*cdf_fl)^-1 * (-log(cdf_az)*-log(cdf_fl))^((exp(
    c(2))+1)-1) * ((-log(cdf_az))^(exp(c(2))+1) + (-log(cdf_fl))^(exp(c(2))+1))^(-2+1/(
    exp(c(2))+1)) * (((-log(cdf_az))^(exp(c(2))+1)+(-log(cdf_fl))^(exp(c(2))+1))^(1/(exp(
    c(2))+1))+(exp(c(2))+1)-1) )*f1 *f2
```

```
logl_Clay_Gum_az_fl.append logl32 = log(term1)
smpl 1976q1 2009q1
logl_Clay_Gum_az_fl.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_Gum_az_fl_CLAY = logl_Clay_Gum_az_fl.@coef(2)
scalar Clay_Gum_az_fl_GUM = logl_Clay_Gum_az_fl.@coef(3)
scalar Clay_Gum_az_fl_PI = logl_Clay_Gum_az_fl.@coef(1)


'Clayton-Gumbel mixture AZ-NV'
c(1) = clayton_az_nv
c(2) = gumbel_az_nv
c(3) = -1

logl logl_Clay_Gum_az_nv
logl_Clay_Gum_az_nv.append @logl logl33
logl_Clay_Gum_az_nv.append f1=pdf_az
logl_Clay_Gum_az_nv.append f2=pdf_nv
logl_Clay_Gum_az_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_az*cdf_nv)
    ^(-(c(1)^2)-1) * ( cdf_nv^(-(c(1)^2)) + cdf_az^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2) + (1-
    exp(c(3))/(1+exp(c(3)))) * exp(-((-log(cdf_az))^(exp(c(2))+1)+(-log(cdf_nv))^(exp(c
    (2))+1))^(1/(exp(c(2))+1))) * (cdf_az*cdf_nv)^-1 * (-log(cdf_az)*-log(cdf_nv))^((exp(
    c(2))+1)-1) * ((-log(cdf_az))^(exp(c(2))+1) + (-log(cdf_nv))^(exp(c(2))+1))^(-2+1/(
    exp(c(2))+1)) * (((-log(cdf_az))^(exp(c(2))+1)+(-log(cdf_nv))^(exp(c(2))+1))^(1/(exp(
    c(2))+1))+(exp(c(2))+1)-1) )*f1 *f2
logl_Clay_Gum_az_nv.append logl33 = log(term1)
smpl 1976q1 2009q1
logl_Clay_Gum_az_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_Gum_az_nv_CLAY = logl_Clay_Gum_az_nv.@coef(2)
scalar Clay_Gum_az_nv_GUM = logl_Clay_Gum_az_nv.@coef(3)
scalar Clay_Gum_az_nv_PI = logl_Clay_Gum_az_nv.@coef(1)
```

```
'Clayton-Gumbel mixture CA-FL'
c(1) = clayton_ca_fl
c(2) = gumbel_ca_fl
c(3) = -1

logl logl_Clay_Gum_ca_fl
logl_Clay_Gum_ca_fl.append @logl logl34
logl_Clay_Gum_ca_fl.append f1=pdf_ca
logl_Clay_Gum_ca_fl.append f2=pdf_fl
logl_Clay_Gum_ca_fl.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_ca*cdf_fl)
    ^(-(c(1)^2)-1) * ( cdf_fl^(-(c(1)^2)) + cdf_ca^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2) + (1-
    exp(c(3))/(1+exp(c(3)))) * exp(-((-log(cdf_ca))^(exp(c(2))+1)+(-log(cdf_fl))^(exp(c
    (2))+1))^(1/(exp(c(2))+1))) * (cdf_ca*cdf_fl)^-1 * (-log(cdf_ca)*-log(cdf_fl))^((exp(
    c(2))+1)-1) * ((-log(cdf_ca))^(exp(c(2))+1) + (-log(cdf_fl))^(exp(c(2))+1))^(-2+1/(
    exp(c(2))+1)) * (((-log(cdf_ca))^(exp(c(2))+1)+(-log(cdf_fl))^(exp(c(2))+1))^(1/(exp(
    c(2))+1))+(exp(c(2))+1)-1) )*f1 *f2
logl_Clay_Gum_ca_fl.append logl34 = log(term1)
smpl 1976q1 2009q1
logl_Clay_Gum_ca_fl.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_Gum_ca_fl_CLAY = logl_Clay_Gum_ca_fl.@coef(2)
scalar Clay_Gum_ca_fl_GUM = logl_Clay_Gum_ca_fl.@coef(3)
scalar Clay_Gum_ca_fl_PI = logl_Clay_Gum_ca_fl.@coef(1)


'Clayton-Gumbel mixture CA-NV'
c(1) = clayton_ca_nv
c(2) = gumbel_ca_nv
c(3) = -1

logl logl_Clay_Gum_ca_nv
logl_Clay_Gum_ca_nv.append @logl logl35
logl_Clay_Gum_ca_nv.append f1=pdf_ca
logl_Clay_Gum_ca_nv.append f2=pdf_nv
```

```
logl_Clay_Gum_ca_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_ca*cdf_nv)
     ^(-(c(1)^2)-1) * ( cdf_nv^(-(c(1)^2)) + cdf_ca^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2) + (1-
     exp(c(3))/(1+exp(c(3)))) * exp(-((-log(cdf_ca))^(exp(c(2))+1)+(-log(cdf_nv))^(exp(c
     (2))+1))^(1/(exp(c(2))+1))) * (cdf_ca*cdf_nv)^-1 * (-log(cdf_ca)*-log(cdf_nv))^((exp(
     c(2))+1)-1) * ((-log(cdf_ca))^(exp(c(2))+1) + (-log(cdf_nv))^(exp(c(2))+1))^(-2+1/(
     exp(c(2))+1)) * (((-log(cdf_ca))^(exp(c(2))+1)+(-log(cdf_nv))^(exp(c(2))+1))^(1/(exp(
     c(2))+1))+(exp(c(2))+1)-1) )*f1 *f2
logl_Clay_Gum_ca_nv.append logl35 = log(term1)
smpl 1976q1 2009q1
logl_Clay_Gum_ca_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
     in the code.'
scalar Clay_Gum_ca_nv_CLAY = logl_Clay_Gum_ca_nv.@coef(2)
scalar Clay_Gum_ca_nv_GUM = logl_Clay_Gum_ca_nv.@coef(3)
scalar Clay_Gum_ca_nv_PI = logl_Clay_Gum_ca_nv.@coef(1)


'Clayton-Gumbel mixture FL-NV'
c(1) = clayton_fl_nv
c(2) = gumbel_fl_nv
c(3) = -1

logl logl_Clay_Gum_fl_nv
logl_Clay_Gum_fl_nv.append @logl logl36
logl_Clay_Gum_fl_nv.append f1=pdf_fl
logl_Clay_Gum_fl_nv.append f2=pdf_nv
logl_Clay_Gum_fl_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_fl*cdf_nv)
     ^(-(c(1)^2)-1) * ( cdf_nv^(-(c(1)^2)) + cdf_fl^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2) + (1-
     exp(c(3))/(1+exp(c(3)))) * exp(-((-log(cdf_fl))^(exp(c(2))+1)+(-log(cdf_nv))^(exp(c
     (2))+1))^(1/(exp(c(2))+1))) * (cdf_fl*cdf_nv)^-1 * (-log(cdf_fl)*-log(cdf_nv))^((exp(
     c(2))+1)-1) * ((-log(cdf_fl))^(exp(c(2))+1) + (-log(cdf_nv))^(exp(c(2))+1))^(-2+1/(
     exp(c(2))+1)) * (((-log(cdf_fl))^(exp(c(2))+1)+(-log(cdf_nv))^(exp(c(2))+1))^(1/(exp(
     c(2))+1))+(exp(c(2))+1)-1) )*f1 *f2
logl_Clay_Gum_fl_nv.append logl36 = log(term1)
smpl 1976q1 2009q1
logl_Clay_Gum_fl_nv.ml(showstart)
```

```
'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_Gum_fl_nv_CLAY = logl_Clay_Gum_fl_nv.@coef(2)
scalar Clay_Gum_fl_nv_GUM = logl_Clay_Gum_fl_nv.@coef(3)
scalar Clay_Gum_fl_nv_PI = logl_Clay_Gum_fl_nv.@coef(1)

'-------------------------------------------------------------------------------------
    '
'Clayton-survival Clayton mixture AZ-CA'
c(1) = clayton_az_ca
c(2) = survival_clayton_az_ca
c(3) = -1

logl logl_Clay_surv_Clay_az_ca
logl_Clay_surv_Clay_az_ca.append @logl logl51
logl_Clay_surv_Clay_az_ca.append f1=pdf_az
logl_Clay_surv_Clay_az_ca.append f2=pdf_ca
logl_Clay_surv_Clay_az_ca.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_ca*
    cdf_az)^(-(c(1)^2)-1) * ( cdf_az^(-(c(1)^2)) + cdf_ca^(-(c(1)^2)) -1)^(-1/(c(1)^2)
    -2) + (1-exp(c(3))/(1+exp(c(3)))) * (c(2)^2)*(1/(c(2)^2)+1) * ((1-cdf_ca)*(1-cdf_az))
    ^(-(c(2)^2)-1) * ( (1-cdf_ca)^(-(c(2)^2)) + (1-cdf_az)^(-(c(2)^2))-1 )^(-1/(c(2)^2)
    -2) )*f1*f2
logl_Clay_surv_Clay_az_ca.append logl51 = log(term1)
smpl 1976q1 2009q1
logl_Clay_surv_Clay_az_ca.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_surv_clay_az_ca_CLAY = logl_Clay_surv_Clay_az_ca.@coef(2)
scalar Clay_surv_clay_az_ca_surv_clay = logl_Clay_surv_Clay_az_ca.@coef(3)
scalar Clay_surv_clay_az_ca_PI = logl_Clay_surv_Clay_az_ca.@coef(1)


'Clayton-survival Clayton mixture AZ-FL'
c(1) = clayton_az_fl
c(2) = survival_clayton_az_fl
```

```
c(3) = -1

logl logl_Clay_surv_Clay_az_fl
logl_Clay_surv_Clay_az_fl.append @logl logl52
logl_Clay_surv_Clay_az_fl.append f1=pdf_az
logl_Clay_surv_Clay_az_fl.append f2=pdf_fl
logl_Clay_surv_Clay_az_fl.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_fl*
    cdf_az)^(-(c(1)^2)-1) * ( cdf_az^(-(c(1)^2)) + cdf_fl^(-(c(1)^2)) -1)^(-1/(c(1)^2)
    -2) + (1-exp(c(3))/(1+exp(c(3)))) * (c(2)^2)*(1/(c(2)^2)+1) * ((1-cdf_fl)*(1-cdf_az))
    ^(-(c(2)^2)-1) * ( (1-cdf_fl)^(-(c(2)^2)) + (1-cdf_az)^(-(c(2)^2))-1 )^(-1/(c(2)^2)
    -2) )*f1*f2
logl_Clay_surv_Clay_az_fl.append logl52 = log(term1)
smpl 1976q1 2009q1
logl_Clay_surv_Clay_az_fl.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_surv_clay_az_fl_CLAY = logl_Clay_surv_Clay_az_fl.@coef(2)
scalar Clay_surv_clay_az_fl_surv_clay = logl_Clay_surv_Clay_az_fl.@coef(3)
scalar Clay_surv_clay_az_fl_PI = logl_Clay_surv_Clay_az_fl.@coef(1)



'Clayton-survival Clayton mixture AZ-NV'
c(1) = clayton_az_nv
c(2) = survival_clayton_az_nv
c(3) = -1

logl logl_Clay_surv_Clay_az_nv
logl_Clay_surv_Clay_az_nv.append @logl logl53
logl_Clay_surv_Clay_az_nv.append f1=pdf_az
logl_Clay_surv_Clay_az_nv.append f2=pdf_nv
logl_Clay_surv_Clay_az_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_nv*
    cdf_az)^(-(c(1)^2)-1) * ( cdf_az^(-(c(1)^2)) + cdf_nv^(-(c(1)^2)) -1)^(-1/(c(1)^2)
    -2) + (1-exp(c(3))/(1+exp(c(3)))) * (c(2)^2)*(1/(c(2)^2)+1) * ((1-cdf_nv)*(1-cdf_az))
    ^(-(c(2)^2)-1) * ( (1-cdf_nv)^(-(c(2)^2)) + (1-cdf_az)^(-(c(2)^2))-1 )^(-1/(c(2)^2)
    -2) )*f1*f2
logl_Clay_surv_Clay_az_nv.append logl53 = log(term1)
```

```
smpl 1976q1 2009q1
logl_Clay_surv_Clay_az_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_surv_clay_az_nv_CLAY = logl_Clay_surv_Clay_az_nv.@coef(2)
scalar Clay_surv_clay_az_nv_surv_clay = logl_Clay_surv_Clay_az_nv.@coef(3)
scalar Clay_surv_clay_az_nv_PI = logl_Clay_surv_Clay_az_nv.@coef(1)



'Clayton-survival Clayton mixture CA-FL'
c(1) = clayton_ca_fl
c(2) = survival_clayton_ca_fl
c(3) = -1

logl logl_Clay_surv_Clay_ca_fl
logl_Clay_surv_Clay_ca_fl.append @logl logl54
logl_Clay_surv_Clay_ca_fl.append f1=pdf_ca
logl_Clay_surv_Clay_ca_fl.append f2=pdf_fl
logl_Clay_surv_Clay_ca_fl.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_fl*
    cdf_ca)^(-(c(1)^2)-1) * ( cdf_ca^(-(c(1)^2)) + cdf_fl^(-(c(1)^2)) -1)^(-1/(c(1)^2)
    -2) + (1-exp(c(3))/(1+exp(c(3)))) * (c(2)^2)*(1/(c(2)^2)+1) * ((1-cdf_fl)*(1-cdf_ca))
    ^(-(c(2)^2)-1) * ( (1-cdf_fl)^(-(c(2)^2)) + (1-cdf_ca)^(-(c(2)^2))-1 )^(-1/(c(2)^2)
    -2) )*f1*f2
logl_Clay_surv_Clay_ca_fl.append logl54 = log(term1)
smpl 1976q1 2009q1
logl_Clay_surv_Clay_ca_fl.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_surv_clay_ca_fl_CLAY = logl_Clay_surv_Clay_ca_fl.@coef(2)
scalar Clay_surv_clay_ca_fl_surv_clay = logl_Clay_surv_Clay_ca_fl.@coef(3)
scalar Clay_surv_clay_ca_fl_PI = logl_Clay_surv_Clay_ca_fl.@coef(1)



'Clayton-survival Clayton mixture CA-NV'
c(1) = clayton_ca_nv
```

```
c(2) = survival_clayton_ca_nv
c(3) = -1

logl logl_Clay_surv_Clay_ca_nv
logl_Clay_surv_Clay_ca_nv.append @logl logl55
logl_Clay_surv_Clay_ca_nv.append f1=pdf_ca
logl_Clay_surv_Clay_ca_nv.append f2=pdf_nv
logl_Clay_surv_Clay_ca_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_nv*
     cdf_ca)^(-(c(1)^2)-1) * ( cdf_ca^(-(c(1)^2)) + cdf_nv^(-(c(1)^2)) -1)^(-1/(c(1)^2)
     -2) + (1-exp(c(3))/(1+exp(c(3)))) * (c(2)^2)*(1/(c(2)^2)+1) * ((1-cdf_nv)*(1-cdf_ca))
     ^(-(c(2)^2)-1) * ( (1-cdf_nv)^(-(c(2)^2)) + (1-cdf_ca)^(-(c(2)^2))-1 )^(-1/(c(2)^2)
     -2) )*f1*f2
logl_Clay_surv_Clay_ca_nv.append logl55 = log(term1)
smpl 1976q1 2009q1
logl_Clay_surv_Clay_ca_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
     in the code.'
scalar Clay_surv_clay_ca_nv_CLAY = logl_Clay_surv_Clay_ca_nv.@coef(2)
scalar Clay_surv_clay_ca_nv_surv_clay = logl_Clay_surv_Clay_ca_nv.@coef(3)
scalar Clay_surv_clay_ca_nv_PI = logl_Clay_surv_Clay_ca_nv.@coef(1)


'Clayton-survival Clayton mixture FL-NV'
c(1) = clayton_fl_nv
c(2) = survival_clayton_fl_nv
c(3) = -1

logl logl_Clay_surv_Clay_fl_nv
logl_Clay_surv_Clay_fl_nv.append @logl logl56
logl_Clay_surv_Clay_fl_nv.append f1=pdf_fl
logl_Clay_surv_Clay_fl_nv.append f2=pdf_nv
logl_Clay_surv_Clay_fl_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_nv*
     cdf_fl)^(-(c(1)^2)-1) * ( cdf_fl^(-(c(1)^2)) + cdf_nv^(-(c(1)^2)) -1)^(-1/(c(1)^2)
     -2) + (1-exp(c(3))/(1+exp(c(3)))) * (c(2)^2)*(1/(c(2)^2)+1) * ((1-cdf_nv)*(1-cdf_fl))
     ^(-(c(2)^2)-1) * ( (1-cdf_nv)^(-(c(2)^2)) + (1-cdf_fl)^(-(c(2)^2))-1 )^(-1/(c(2)^2)
     -2) )*f1*f2
```

```
logl_Clay_surv_Clay_fl_nv.append logl56 = log(term1)
smpl 1976q1 2009q1
logl_Clay_surv_Clay_fl_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_surv_clay_fl_nv_CLAY = logl_Clay_surv_Clay_fl_nv.@coef(2)
scalar Clay_surv_clay_fl_nv_surv_clay = logl_Clay_surv_Clay_fl_nv.@coef(3)
scalar Clay_surv_clay_fl_nv_PI = logl_Clay_surv_Clay_fl_nv.@coef(1)


'-------------------------------------------------------------------------------------------'
'Gumbel-survival Gumbel mixture AZ-CA'
c(1) = Gumbel_az_ca
c(2) = survival_Gumbel_az_ca
c(3) = -1

logl logl_Gum_surv_Gum_az_ca
logl_Gum_surv_Gum_az_ca.append @logl logl71
logl_Gum_surv_Gum_az_ca.append f1=pdf_az
logl_Gum_surv_Gum_az_ca.append f2=pdf_ca
logl_Gum_surv_Gum_az_ca.append term1 = ( exp(c(3))/(1+exp(c(3)))* exp(-((-log(cdf_az))^(
    exp(c(1))+1)+(-log(cdf_ca))^(exp(c(1))+1))^(1/(exp(c(1))+1))) * (cdf_az*cdf_ca)^-1 *
    (-log(cdf_az)*-log(cdf_ca))^((exp(c(1))+1)-1) * ((-log(cdf_az))^(exp(c(1))+1) + (-log
    (cdf_ca))^(exp(c(1))+1))^(-2+1/(exp(c(1))+1)) * (((-log(cdf_az))^(exp(c(1))+1)+(-log(
    cdf_ca))^(exp(c(1))+1))^(1/(exp(c(1))+1))+(exp(c(1))+1)-1) + (1-exp(c(3))/(1+exp(c(3)
    ))) * ((exp(-((-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))
    ^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_ca))^(1/(
    exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp
    (c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log
    (1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((cdf_ca - 1)*(cdf_az - 1)) - (exp
    (-((-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2)
    )/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log
    (1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c
    (2))+1))))^((exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2)
    )+1)) - 1)*(-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c
    (2))+1))*(cdf_ca - 1)*(cdf_az - 1))) )*f1 *f2
```

```
logl_Gum_surv_Gum_az_ca.append logl71 = log(term1)
smpl 1976q1 2009q1
logl_Gum_surv_Gum_az_ca.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Gum_surv_Gum_az_ca_GUM = logl_Gum_surv_Gum_az_ca.@coef(2)
scalar Gum_surv_Gum_az_ca_surv_GUM = logl_Gum_surv_Gum_az_ca.@coef(3)
scalar Gum_surv_Gum_az_ca_PI = logl_Gum_surv_Gum_az_ca.@coef(1)



'Gumbel-survival Gumbel mixture AZ-FL'
c(1) = Gumbel_az_fl
c(2) = survival_Gumbel_az_fl
c(3) = -1

logl logl_Gum_surv_Gum_az_fl
logl_Gum_surv_Gum_az_fl.append @logl logl72
logl_Gum_surv_Gum_az_fl.append f1=pdf_az
logl_Gum_surv_Gum_az_fl.append f2=pdf_fl
logl_Gum_surv_Gum_az_fl.append term1 = ( exp(c(3))/(1+exp(c(3)))* exp(-((-log(cdf_az))^(
    exp(c(1))+1)+(-log(cdf_fl))^(exp(c(1))+1))^(1/(exp(c(1))+1))) * (cdf_az*cdf_fl)^-1 *
    (-log(cdf_az)*-log(cdf_fl))^((exp(c(1))+1)-1) * ((-log(cdf_az))^(exp(c(1))+1) + (-log
    (cdf_fl))^(exp(c(1))+1))^(-2+1/(exp(c(1))+1)) * (((-log(cdf_az))^(exp(c(1))+1)+(-log(
    cdf_fl))^(exp(c(1))+1))^(1/(exp(c(1))+1))+(exp(c(1))+1)-1) + (1-exp(c(3))/(1+exp(c(3)
    ))) * ((exp(-((-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))
    ^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_fl))^(1/(
    exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp
    (c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log
    (1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((cdf_fl - 1)*(cdf_az - 1)) - (exp
    (-((-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2)
    )/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log
    (1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c
    (2))+1))))^((exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2)
    )+1)) - 1)*(-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c
    (2))+1))*(cdf_fl - 1)*(cdf_az - 1))) )*f1 *f2
logl_Gum_surv_Gum_az_fl.append logl72 = log(term1)
```

```
smpl 1976q1 2009q1
logl_Gum_surv_Gum_az_fl.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Gum_surv_Gum_az_fl_GUM = logl_Gum_surv_Gum_az_fl.@coef(2)
scalar Gum_surv_Gum_az_fl_surv_GUM = logl_Gum_surv_Gum_az_fl.@coef(3)
scalar Gum_surv_Gum_az_fl_PI = logl_Gum_surv_Gum_az_fl.@coef(1)



'Gumbel-survival Gumbel mixture AZ-NV'
c(1) = Gumbel_az_nv
c(2) = survival_Gumbel_az_nv
c(3) = -1

logl logl_Gum_surv_Gum_az_nv
logl_Gum_surv_Gum_az_nv.append @logl logl73
logl_Gum_surv_Gum_az_nv.append f1=pdf_az
logl_Gum_surv_Gum_az_nv.append f2=pdf_nv
logl_Gum_surv_Gum_az_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))* exp(-((-log(cdf_az))^(
    exp(c(1))+1)+(-log(cdf_nv))^(exp(c(1))+1))^(1/(exp(c(1))+1))) * (cdf_az*cdf_nv)^-1 *
    (-log(cdf_az)*-log(cdf_nv))^((exp(c(1))+1)-1) * ((-log(cdf_az))^(exp(c(1))+1) + (-log
    (cdf_nv))^(exp(c(1))+1))^(-2+1/(exp(c(1))+1)) * (((-log(cdf_az))^(exp(c(1))+1)+(-log(
    cdf_nv))^(exp(c(1))+1))^(1/(exp(c(1))+1))+(exp(c(1))+1)-1) + (1-exp(c(3))/(1+exp(c(3)
    ))) * ((exp(-((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))
    ^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_nv))^(1/(
    exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp
    (c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log
    (1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((cdf_nv - 1)*(cdf_az - 1)) - (exp
    (-((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2)
    )/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log
    (1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c
    (2))+1))))^((exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2)
    )+1)) - 1)*(-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c
    (2))+1))*(cdf_nv - 1)*(cdf_az - 1))) )*f1 *f2
logl_Gum_surv_Gum_az_nv.append logl73 = log(term1)
smpl 1976q1 2009q1
```

```
logl_Gum_surv_Gum_az_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Gum_surv_Gum_az_nv_GUM = logl_Gum_surv_Gum_az_nv.@coef(2)
scalar Gum_surv_Gum_az_nv_surv_GUM = logl_Gum_surv_Gum_az_nv.@coef(3)
scalar Gum_surv_Gum_az_nv_PI = logl_Gum_surv_Gum_az_nv.@coef(1)


'Gumbel-survival Gumbel mixture CA-FL'
c(1) = Gumbel_ca_fl
c(2) = survival_Gumbel_ca_fl
c(3) = -1

logl logl_Gum_surv_Gum_ca_fl
logl_Gum_surv_Gum_ca_fl.append @logl logl74
logl_Gum_surv_Gum_ca_fl.append f1=pdf_ca
logl_Gum_surv_Gum_ca_fl.append f2=pdf_fl
logl_Gum_surv_Gum_ca_fl.append term1 = ( exp(c(3))/(1+exp(c(3)))* exp(-((-log(cdf_ca))^(
    exp(c(1))+1)+(-log(cdf_fl))^(exp(c(1))+1))^(1/(exp(c(1))+1))) * (cdf_ca*cdf_fl)^-1 *
    (-log(cdf_ca)*-log(cdf_fl))^((exp(c(1))+1)-1) * ((-log(cdf_ca))^(exp(c(1))+1) + (-log
    (cdf_fl))^(exp(c(1))+1))^(-2+1/(exp(c(1))+1)) * (((-log(cdf_ca))^(exp(c(1))+1)+(-log(
    cdf_fl))^(exp(c(1))+1))^(1/(exp(c(1))+1))+(exp(c(1))+1)-1) + (1-exp(c(3))/(1+exp(c(3)
    ))) * ((exp(-((-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))
    ^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_fl))^(1/(
    exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp
    (c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log
    (1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((cdf_fl - 1)*(cdf_ca - 1)) - (exp
    (-((-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2)
    )/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log
    (1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c
    (2))+1))))^((exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2)
    )+1)) - 1)*(-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c
    (2))+1))*(cdf_fl - 1)*(cdf_ca - 1))) )*f1 *f2
logl_Gum_surv_Gum_ca_fl.append logl74 = log(term1)
smpl 1976q1 2009q1
logl_Gum_surv_Gum_ca_fl.ml(showstart)
```

```
'Note that the output first gives the coefficient for pi, even though it is called c(3)
     in the code.'
scalar Gum_surv_Gum_ca_fl_GUM = logl_Gum_surv_Gum_ca_fl.@coef(2)
scalar Gum_surv_Gum_ca_fl_surv_GUM = logl_Gum_surv_Gum_ca_fl.@coef(3)
scalar Gum_surv_Gum_ca_fl_PI = logl_Gum_surv_Gum_ca_fl.@coef(1)


'Gumbel-survival Gumbel mixture CA-NV'
c(1) = Gumbel_ca_nv
c(2) = survival_Gumbel_ca_nv
c(3) = -1

logl logl_Gum_surv_Gum_ca_nv
logl_Gum_surv_Gum_ca_nv.append @logl logl75
logl_Gum_surv_Gum_ca_nv.append f1=pdf_ca
logl_Gum_surv_Gum_ca_nv.append f2=pdf_nv
logl_Gum_surv_Gum_ca_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))* exp(-((-log(cdf_ca))^(
     exp(c(1))+1)+(-log(cdf_nv))^(exp(c(1))+1))^(1/(exp(c(1))+1))) * (cdf_ca*cdf_nv)^-1 *
     (-log(cdf_ca)*-log(cdf_nv))^((exp(c(1))+1)-1) * ((-log(cdf_ca))^(exp(c(1))+1) + (-log
     (cdf_nv))^(exp(c(1))+1))^(-2+1/(exp(c(1))+1)) * (((-log(cdf_ca))^(exp(c(1))+1)+(-log(
     cdf_nv))^(exp(c(1))+1))^(1/(exp(c(1))+1))+(exp(c(1))+1)-1) + (1-exp(c(3))/(1+exp(c(3)
     ))) * ((exp(-((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))
     ^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_nv))^(1/(
     exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp
     (c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log
     (1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((cdf_nv - 1)*(cdf_ca - 1)) - (exp
     (-((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2)
     )/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log
     (1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c
     (2))+1))))^((exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2)
     )+1)) - 1)*(-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c
     (2))+1))*(cdf_nv - 1)*(cdf_ca - 1))) )*f1 *f2
logl_Gum_surv_Gum_ca_nv.append logl75 = log(term1)
smpl 1976q1 2009q1
logl_Gum_surv_Gum_ca_nv.ml(showstart)
```

```
'Note that the output first gives the coefficient for pi, even though it is called c(3)
     in the code.'
scalar Gum_surv_Gum_ca_nv_GUM = logl_Gum_surv_Gum_ca_nv.@coef(2)
scalar Gum_surv_Gum_ca_nv_surv_GUM = logl_Gum_surv_Gum_ca_nv.@coef(3)
scalar Gum_surv_Gum_ca_nv_PI = logl_Gum_surv_Gum_ca_nv.@coef(1)


'Gumbel-survival Gumbel mixture FL-NV'
c(1) = Gumbel_fl_nv
c(2) = survival_Gumbel_fl_nv
c(3) = -1

logl logl_Gum_surv_Gum_fl_nv
logl_Gum_surv_Gum_fl_nv.append @logl logl76
logl_Gum_surv_Gum_fl_nv.append f1=pdf_fl
logl_Gum_surv_Gum_fl_nv.append f2=pdf_nv
logl_Gum_surv_Gum_fl_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))* exp(-((-log(cdf_fl))^(
     exp(c(1))+1)+(-log(cdf_nv))^(exp(c(1))+1))^(1/(exp(c(1))+1))) * (cdf_fl*cdf_nv)^-1 *
     (-log(cdf_fl)*-log(cdf_nv))^((exp(c(1))+1)-1) * ((-log(cdf_fl))^(exp(c(1))+1) + (-log
     (cdf_nv))^(exp(c(1))+1))^(-2+1/(exp(c(1))+1)) * (((-log(cdf_fl))^(exp(c(1))+1)+(-log(
     cdf_nv))^(exp(c(1))+1))^(1/(exp(c(1))+1))+(exp(c(1))+1)-1) + (1-exp(c(3))/(1+exp(c(3)
     ))) * ((exp(-((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_fl))
     ^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_nv))^(1/(
     exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp
     (c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log
     (1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((cdf_nv - 1)*(cdf_fl - 1)) - (exp
     (-((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_fl))^(1/(exp(c(2)
     )/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log
     (1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c
     (2))+1))))^((exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2)
     )+1)) - 1)*(-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c
     (2))+1))*(cdf_nv - 1)*(cdf_fl - 1))) )*f1 *f2
logl_Gum_surv_Gum_fl_nv.append logl76 = log(term1)
smpl 1976q1 2009q1
logl_Gum_surv_Gum_fl_nv.ml(showstart)
```

```
'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Gum_surv_Gum_fl_nv_GUM = logl_Gum_surv_Gum_fl_nv.@coef(2)
scalar Gum_surv_Gum_fl_nv_surv_GUM = logl_Gum_surv_Gum_fl_nv.@coef(3)
scalar Gum_surv_Gum_fl_nv_PI = logl_Gum_surv_Gum_fl_nv.@coef(1)

'-------------------------------------------------------------------------------------'
'Since the obtained coefficient values are made by incorporating the constraint on theta,
     we still have to obtain the actual theta'

'Clayton-Gumbel mixture'
scalar theta_clay_CG_AZCA = Clay_Gum_az_ca_CLAY^2
scalar theta_gum_CG_AZCA = exp(Clay_Gum_az_ca_GUM)+1
scalar pi_CG_AZCA = exp(Clay_Gum_az_ca_PI)/(1+exp(Clay_Gum_az_ca_PI))

scalar theta_clay_CG_AZFL = Clay_Gum_az_fl_CLAY^2
scalar theta_gum_CG_AZFL = exp(Clay_Gum_az_fl_GUM)+1
scalar pi_CG_AZFL = exp(Clay_Gum_az_fl_PI)/(1+exp(Clay_Gum_az_fl_PI))

scalar theta_clay_CG_AZNV = Clay_Gum_az_nv_CLAY^2
scalar theta_gum_CG_AZNV = exp(Clay_Gum_az_nv_GUM)+1
scalar pi_CG_AZNV = exp(Clay_Gum_az_nv_PI)/(1+exp(Clay_Gum_az_nv_PI))

scalar theta_clay_CG_CAFL = Clay_Gum_ca_fl_CLAY^2
scalar theta_gum_CG_CAFL = exp(Clay_Gum_ca_fl_GUM)+1
scalar pi_CG_CAFL = exp(Clay_Gum_ca_fl_PI)/(1+exp(Clay_Gum_ca_fl_PI))

scalar theta_clay_CG_CANV = Clay_Gum_ca_nv_CLAY^2
scalar theta_gum_CG_CANV = exp(Clay_Gum_ca_nv_GUM)+1
scalar pi_CG_CANV = exp(Clay_Gum_ca_nv_PI)/(1+exp(Clay_Gum_ca_nv_PI))

scalar theta_clay_CG_FLNV = Clay_Gum_fl_nv_CLAY^2
scalar theta_gum_CG_FLNV = exp(Clay_Gum_fl_nv_GUM)+1
scalar pi_CG_FLNV = exp(Clay_Gum_fl_nv_PI)/(1+exp(Clay_Gum_fl_nv_PI))

'Clayton-survival Clayton mixture'
scalar theta_clay_CSC_AZCA = Clay_surv_clay_az_ca_CLAY^2
```

```
scalar theta_survClay_CSC_AZCA = Clay_surv_clay_az_ca_surv_clay^2
scalar pi_CSC_AZCA = exp(Clay_surv_clay_az_ca_PI)/(1+exp(Clay_surv_clay_az_ca_PI))

scalar theta_clay_CSC_AZFL = Clay_surv_clay_az_fl_CLAY^2
scalar theta_survClay_CSC_AZFL = Clay_surv_clay_az_fl_surv_clay^2
scalar pi_CSC_AZFL = exp(Clay_surv_clay_az_fl_PI)/(1+exp(Clay_surv_clay_az_fl_PI))

scalar theta_clay_CSC_AZNV = Clay_surv_clay_az_nv_CLAY^2
scalar theta_survClay_CSC_AZNV = Clay_surv_clay_az_nv_surv_clay^2
scalar pi_CSC_AZNV = exp(Clay_surv_clay_az_nv_PI)/(1+exp(Clay_surv_clay_az_nv_PI))

scalar theta_clay_CSC_CAFL = Clay_surv_clay_ca_fl_CLAY^2
scalar theta_survClay_CSC_CAFL = Clay_surv_clay_ca_fl_surv_clay^2
scalar pi_CSC_CAFL = exp(Clay_surv_clay_ca_fl_PI)/(1+exp(Clay_surv_clay_ca_fl_PI))

scalar theta_clay_CSC_CANV = Clay_surv_clay_ca_nv_CLAY^2
scalar theta_survClay_CSC_CANV = Clay_surv_clay_ca_nv_surv_clay^2
scalar pi_CSC_CANV = exp(Clay_surv_clay_ca_nv_PI)/(1+exp(Clay_surv_clay_ca_nv_PI))

scalar theta_clay_CSC_FLNV = Clay_surv_clay_fl_nv_CLAY^2
scalar theta_survClay_CSC_FLNV = Clay_surv_clay_fl_nv_surv_clay^2
scalar pi_CSC_FLNV = exp(Clay_surv_clay_fl_nv_PI)/(1+exp(Clay_surv_clay_fl_nv_PI))

'gumbel-survival gumbel mixture'
scalar theta_gum_GSG_AZCA = exp(Gum_surv_Gum_az_ca_GUM)+1
scalar theta_survGum_GSG_AZCA = exp(Gum_surv_Gum_az_ca_surv_GUM)/(1+exp(
    Gum_surv_Gum_az_ca_surv_GUM ))
scalar pi_GSG_AZCA = exp(Gum_surv_Gum_az_ca_PI)/(1+exp(Gum_surv_Gum_az_ca_PI ))

scalar theta_gum_GSG_AZFL = exp(Gum_surv_Gum_az_fl_GUM)+1
scalar theta_survGum_GSG_AZFL = exp(Gum_surv_Gum_az_fl_surv_GUM)/(1+exp(
    Gum_surv_Gum_az_fl_surv_GUM ))
scalar pi_GSG_AZFL = exp(Gum_surv_Gum_az_fl_PI)/(1+exp(Gum_surv_Gum_az_fl_PI ))

scalar theta_gum_GSG_AZNV = exp(Gum_surv_Gum_az_nv_GUM)+1
scalar theta_survGum_GSG_AZNV = exp(Gum_surv_Gum_az_nv_surv_GUM)/(1+exp(
    Gum_surv_Gum_az_nv_surv_GUM ))
```

```
scalar pi_GSG_AZNV = exp(Gum_surv_Gum_az_nv_PI)/(1+exp(Gum_surv_Gum_az_nv_PI ))

scalar theta_gum_GSG_CAFL = exp(Gum_surv_Gum_ca_fl_GUM)+1
scalar theta_survGum_GSG_CAFL = exp(Gum_surv_Gum_ca_fl_surv_GUM)/(1+exp(
    Gum_surv_Gum_ca_fl_surv_GUM ))
scalar pi_GSG_CAFL = exp(Gum_surv_Gum_ca_fl_PI)/(1+exp(Gum_surv_Gum_ca_fl_PI ))

scalar theta_gum_GSG_CANV = exp(Gum_surv_Gum_ca_nv_GUM)+1
scalar theta_survGum_GSG_CANV = exp(Gum_surv_Gum_ca_nv_surv_GUM)/(1+exp(
    Gum_surv_Gum_ca_nv_surv_GUM ))
scalar pi_GSG_CANV = exp(Gum_surv_Gum_ca_nv_PI)/(1+exp(Gum_surv_Gum_ca_nv_PI ))

scalar theta_gum_GSG_FLNV = exp(Gum_surv_Gum_fl_nv_GUM)+1
scalar theta_survGum_GSG_FLNV = exp(Gum_surv_Gum_fl_nv_surv_GUM)/(1+exp(
    Gum_surv_Gum_fl_nv_surv_GUM ))
scalar pi_GSG_FLNV = exp(Gum_surv_Gum_fl_nv_PI)/(1+exp(Gum_surv_Gum_fl_nv_PI ))

'-----------------------------------------------------------------------------------------'

'Calculating Kendall‘s tau'
scalar TAU_CLAY_CG_AZCA = THETA_CLAY_CG_AZCA/(THETA_CLAY_CG_AZCA+2)
scalar TAU_CLAY_CG_AZFL = THETA_CLAY_CG_AZFL /(THETA_CLAY_CG_AZFL +2)
scalar TAU_CLAY_CG_AZNV = THETA_CLAY_CG_AZNV / (THETA_CLAY_CG_AZNV +2)
scalar TAU_CLAY_CG_CAFL = THETA_CLAY_CG_CAFL / (THETA_CLAY_CG_CAFL +2)
scalar TAU_CLAY_CG_CANV = THETA_CLAY_CG_CANV / (THETA_CLAY_CG_CANV +2)
scalar TAU_CLAY_CG_FLNV = THETA_CLAY_CG_FLNV / (THETA_CLAY_CG_FLNV +2)
scalar TAU_CLAY_CSC_AZCA = THETA_CLAY_CSC_AZCA / (THETA_CLAY_CSC_AZCA +2)
scalar TAU_CLAY_CSC_AZFL = THETA_CLAY_CSC_AZFL / (THETA_CLAY_CSC_AZFL +2)
scalar TAU_CLAY_CSC_AZNV = THETA_CLAY_CSC_AZNV / (THETA_CLAY_CSC_AZNV+2)
scalar TAU_CLAY_CSC_CAFL = THETA_CLAY_CSC_CAFL / (THETA_CLAY_CSC_CAFL +2)
scalar TAU_CLAY_CSC_CANV = THETA_CLAY_CSC_CANV / (THETA_CLAY_CSC_CANV +2)
scalar TAU_CLAY_CSC_FLNV =THETA_CLAY_CSC_FLNV / (THETA_CLAY_CSC_FLNV +2)
scalar TAU_GUM_CG_AZCA = 1-1/THETA_GUM_CG_AZCA
scalar TAU_GUM_CG_AZFL = 1-1/THETA_GUM_CG_AZFL
scalar TAU_GUM_CG_AZNV = 1-1/THETA_GUM_CG_AZNV
scalar TAU_GUM_CG_CAFL = 1-1/THETA_GUM_CG_CAFL
scalar TAU_GUM_CG_CANV = 1-1/THETA_GUM_CG_CANV
```

```
scalar TAU_GUM_CG_FLNV = 1-1/THETA_GUM_CG_FLNV
scalar TAU_GUM_GSG_AZCA = 1-1/THETA_GUM_GSG_AZCA
scalar TAU_GUM_GSG_AZFL = 1-1/THETA_GUM_GSG_AZFL
scalar TAU_GUM_GSG_AZNV = 1-1/THETA_GUM_GSG_AZNV
scalar TAU_GUM_GSG_CAFL = 1-1/THETA_GUM_GSG_CAFL
scalar TAU_GUM_GSG_CANV = 1-1/THETA_GUM_GSG_CANV
scalar TAU_GUM_GSG_FLNV = 1-1/THETA_GUM_GSG_FLNV
scalar TAU_SURVCLAY_CSC_AZCA = THETA_SURVCLAY_CSC_AZCA/(2+THETA_SURVCLAY_CSC_AZCA)
scalar TAU_SURVCLAY_CSC_AZFL = THETA_SURVCLAY_CSC_AZFL /(2+THETA_SURVCLAY_CSC_AZFL)
scalar TAU_SURVCLAY_CSC_AZNV = THETA_SURVCLAY_CSC_AZNV /(2+THETA_SURVCLAY_CSC_AZNV)
scalar TAU_SURVCLAY_CSC_CAFL = THETA_SURVCLAY_CSC_CAFL /(2+THETA_SURVCLAY_CSC_CAFL)
scalar TAU_SURVCLAY_CSC_CANV = THETA_SURVCLAY_CSC_CANV /(2+THETA_SURVCLAY_CSC_CANV)
scalar TAU_SURVCLAY_CSC_FLNV = THETA_SURVCLAY_CSC_FLNV /(2+THETA_SURVCLAY_CSC_FLNV)
scalar TAU_SURVGUM_GSG_AZCA = 1-THETA_SURVGUM_GSG_AZCA
scalar TAU_SURVGUM_GSG_AZFL = 1-THETA_SURVGUM_GSG_AZFL
scalar TAU_SURVGUM_GSG_AZNV = 1-THETA_SURVGUM_GSG_AZNV
scalar TAU_SURVGUM_GSG_CAFL = 1-THETA_SURVGUM_GSG_CAFL
scalar TAU_SURVGUM_GSG_CANV = 1-THETA_SURVGUM_GSG_CANV
scalar TAU_SURVGUM_GSG_FLNV = 1-THETA_SURVGUM_GSG_FLNV

'-------------------------------------------------------------------------------------'
'Calculation of upper and lower tail dependence'

'Clayton-Gumbel'
scalar dep_clay_cg_azca = 2^(-1/THETA_CLAY_CG_AZCA)
scalar dep_clay_cg_azfl = 2^(-1/THETA_CLAY_CG_AZFL)
scalar dep_clay_cg_aznv = 2^(-1/THETA_CLAY_CG_AZNV)
scalar dep_clay_cg_cafl = 2^(-1/THETA_CLAY_CG_CAFL)
scalar dep_clay_cg_canv = 2^(-1/THETA_CLAY_CG_CANV)
scalar dep_clay_cg_flnv = 2^(-1/THETA_CLAY_CG_FLNV)

scalar dep_gum_cg_azca = 2-2^(1/THETA_GUM_CG_AZCA)
scalar dep_gum_cg_azfl = 2-2^(1/THETA_GUM_CG_AZFL)
scalar dep_gum_cg_aznv = 2-2^(1/THETA_GUM_CG_AZNV)
scalar dep_gum_cg_cafl = 2-2^(1/THETA_GUM_CG_CAFL)
scalar dep_gum_cg_canv = 2-2^(1/THETA_GUM_CG_CANV)
scalar dep_gum_cg_flnv = 2-2^(1/THETA_GUM_CG_FLNV)
```

```
'Clayton-survival Clayton'
scalar dep_clay_csc_azca = 2^(-1/THETA_CLAY_CSC_AZCA)
scalar dep_clay_csc_azfl = 2^(-1/THETA_CLAY_CSC_AZFL)
scalar dep_clay_csc_aznv = 2^(-1/THETA_CLAY_CSC_AZNV)
scalar dep_clay_csc_cafl = 2^(-1/THETA_CLAY_CSC_CAFL)
scalar dep_clay_csc_canv = 2^(-1/THETA_CLAY_CSC_CANV)
scalar dep_clay_csc_flnv = 2^(-1/THETA_CLAY_CSC_FLNV)

scalar dep_survclay_csc_azca = 2^(-1/THETA_SURVCLAY_CSC_AZCA)
scalar dep_survclay_csc_azfl = 2^(-1/THETA_SURVCLAY_CSC_AZFL)
scalar dep_survclay_csc_aznv = 2^(-1/THETA_SURVCLAY_CSC_AZNV)
scalar dep_survclay_csc_cafl = 2^(-1/THETA_SURVCLAY_CSC_CAFL)
scalar dep_survclay_csc_canv = 2^(-1/THETA_SURVCLAY_CSC_CANV)
scalar dep_survclay_csc_flnv = 2^(-1/THETA_SURVCLAY_CSC_FLNV)

'Gumbel-survival Gumbel'
scalar dep_gum_gsg_azca = 2-2^(1/THETA_GUM_GSG_AZCA)
scalar dep_gum_gsg_azfl = 2-2^(1/THETA_GUM_GSG_AZFL)
scalar dep_gum_gsg_aznv = 2-2^(1/THETA_GUM_GSG_AZNV)
scalar dep_gum_gsg_cafl = 2-2^(1/THETA_GUM_GSG_CAFL)
scalar dep_gum_gsg_canv = 2-2^(1/THETA_GUM_GSG_CANV)
scalar dep_gum_gsg_flnv = 2-2^(1/THETA_GUM_GSG_FLNV)

scalar dep_survgum_gsg_azca = 2-2^(THETA_SURVGUM_GSG_AZCA)
scalar dep_survgum_gsg_azfl = 2-2^(THETA_SURVGUM_GSG_AZFL)
scalar dep_survgum_gsg_aznv = 2-2^(THETA_SURVGUM_GSG_AZNV)
scalar dep_survgum_gsg_cafl = 2-2^(THETA_SURVGUM_GSG_CAFL)
scalar dep_survgum_gsg_canv = 2-2^(THETA_SURVGUM_GSG_CANV)
scalar dep_survgum_gsg_flnv = 2-2^(THETA_SURVGUM_GSG_FLNV)

'-----------------------------------------------------------------------------------'
'number of observations used in the maximum likelihood estimation'
scalar numofobs = @obssmpl

'bayes information criterion is in eviews calculated as -2ln(L)/n+k*ln(n)/n, so we need
    to multiply the value by n (= number of observations)'
```

```
scalar BIC_CG_AZCA = logl_clay_gum_az_ca.@sc * numofobs
scalar BIC_CG_AZFL = logl_clay_gum_az_fl.@sc * numofobs
scalar BIC_CG_AZNV = logl_clay_gum_az_nv.@sc * numofobs
scalar BIC_CG_CAFL = logl_clay_gum_ca_fl.@sc * numofobs
scalar BIC_CG_CANV = logl_clay_gum_ca_nv.@sc * numofobs
scalar BIC_CG_FLNV = logl_clay_gum_fl_nv.@sc * numofobs

scalar BIC_CSC_AZCA = logl_clay_surv_clay_az_ca.@sc * numofobs
scalar BIC_CSC_AZFL = logl_clay_surv_clay_az_fl.@sc * numofobs
scalar BIC_CSC_AZNV = logl_clay_surv_clay_az_nv.@sc * numofobs
scalar BIC_CSC_CAFL = logl_clay_surv_clay_ca_fl.@sc * numofobs
scalar BIC_CSC_CANV = logl_clay_surv_clay_ca_nv.@sc * numofobs
scalar BIC_CSC_FLNV = logl_clay_surv_clay_fl_nv.@sc * numofobs

scalar BIC_GSG_AZCA = logl_gum_surv_gum_az_ca.@sc * numofobs
scalar BIC_GSG_AZFL = logl_gum_surv_gum_az_fl.@sc * numofobs
scalar BIC_GSG_AZNV = logl_gum_surv_gum_az_nv.@sc * numofobs
scalar BIC_GSG_CAFL = logl_gum_surv_gum_ca_fl.@sc * numofobs
scalar BIC_GSG_CANV = logl_gum_surv_gum_ca_nv.@sc * numofobs
scalar BIC_GSG_FLNV = logl_gum_surv_gum_fl_nv.@sc * numofobs

'-------------------------------------------------------------------------------------'
'Vuong test statistic'
'Note that it is a 2-sided test!'

'Clayton-gumbel vs Clayton-survival Clayton az-ca'
series m = logl51 - logl31
scalar z_cg_csc_azca = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
scalar Vuong_cg_csc_azca = 2*@cnorm(-@abs(z_cg_csc_azca))

'Clayton-gumbel vs Clayton-survival Clayton az-fl'
series m = logl52 - logl32
scalar z_cg_csc_azfl = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
scalar Vuong_cg_csc_azfl = 2*@cnorm(-@abs(z_cg_csc_azfl))

'Clayton-gumbel vs Clayton-survival Clayton az-nv'
series m = logl53 - logl33
```

```
scalar z_cg_csc_aznv = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
scalar Vuong_cg_csc_aznv = 2*@cnorm(-@abs(z_cg_csc_aznv))

'Clayton-gumbel vs Clayton-survival Clayton ca-fl'
series m = logl54 - logl34
scalar z_cg_csc_cafl = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
scalar Vuong_cg_csc_cafl = 2*@cnorm(-@abs(z_cg_csc_cafl))

'Clayton-gumbel vs Clayton-survival Clayton ca-nv'
series m = logl55 - logl35
scalar z_cg_csc_canv = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
scalar Vuong_cg_csc_canv = 2*@cnorm(-@abs(z_cg_csc_canv))

'Clayton-gumbel vs Clayton-survival Clayton fl-nv'
series m = logl56 - logl36
scalar z_cg_csc_flnv = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
scalar Vuong_cg_csc_flnv = 2*@cnorm(-@abs(z_cg_csc_flnv))

'-------------------------------------------------------------------------------------'
'Clayton-gumbel vs Gumbel-survival Gumbel az-ca'
series m = logl71 - logl31
scalar z_cg_gsg_azca = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
scalar Vuong_cg_gsg_azca = 2*@cnorm(-@abs(z_cg_gsg_azca))

'Clayton-gumbel vs Gumbel-survival Gumbel az-fl'
series m = logl72 - logl32
scalar z_cg_gsg_azfl = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
scalar Vuong_cg_gsg_azfl = 2*@cnorm(-@abs(z_cg_gsg_azfl))

'Clayton-gumbel vs Gumbel-survival Gumbel az-nv'
series m = logl73 - logl33
scalar z_cg_gsg_aznv = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
scalar Vuong_cg_gsg_aznv = 2*@cnorm(-@abs(z_cg_gsg_aznv))

'Clayton-gumbel vs Gumbel-survival Gumbel ca-fl'
series m = logl74 - logl34
scalar z_cg_gsg_cafl = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
```

```
scalar Vuong_cg_gsg_cafl = 2*@cnorm(-@abs(z_cg_gsg_cafl))

'Clayton-gumbel vs Gumbel-survival Gumbel ca-nv'
series m = logl75 - logl35
scalar z_cg_gsg_canv = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
scalar Vuong_cg_gsg_canv = 2*@cnorm(-@abs(z_cg_gsg_canv))

'Clayton-gumbel vs Gumbel-survival Gumbel fl-nv'
series m = logl76 - logl36
scalar z_cg_gsg_flnv = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
scalar Vuong_cg_gsg_flnv = 2*@cnorm(-@abs(z_cg_gsg_flnv))
```

### Block bootstrap approach code

```
'Program for the block bootstrap approach of Clayton-Gumbel'

'Set the sample the same as in Zimmer (2012)'
smpl 1975Q2 2009Q1

'create a group such that we can resample the input all at the same time'
group input az_percent ca_percent fl_percent nv_percent

'set seed for random number generator'
rndseed 12344

scalar rep=500 'number of replications'

'Initialise the vectors'
'theta'
vector(rep) theta_clay_CG_AZCA
vector(rep) theta_gum_CG_AZCA
vector(rep) pi_CG_AZCA

vector(rep) theta_clay_CG_AZFL
vector(rep) theta_gum_CG_AZFL
vector(rep) pi_CG_AZFL
```

```
vector(rep) theta_clay_CG_AZNV
vector(rep) theta_gum_CG_AZNV
vector(rep) pi_CG_AZNV

vector(rep) theta_clay_CG_CAFL
vector(rep) theta_gum_CG_CAFL
vector(rep) pi_CG_CAFL

vector(rep) theta_clay_CG_CANV
vector(rep) theta_gum_CG_CANV
vector(rep) pi_CG_CANV

vector(rep) theta_clay_CG_FLNV
vector(rep) theta_gum_CG_FLNV
vector(rep) pi_CG_FLNV

'Kendall`s tau'
vector(rep) TAU_CLAY_CG_AZCA
vector(rep) TAU_CLAY_CG_AZFL
vector(rep) TAU_CLAY_CG_AZNV
vector(rep) TAU_CLAY_CG_CAFL
vector(rep) TAU_CLAY_CG_CANV
vector(rep) TAU_CLAY_CG_FLNV
vector(rep) TAU_GUM_CG_AZCA
vector(rep) TAU_GUM_CG_AZFL
vector(rep) TAU_GUM_CG_AZNV
vector(rep) TAU_GUM_CG_CAFL
vector(rep) TAU_GUM_CG_CANV
vector(rep) TAU_GUM_CG_FLNV

'Tail dependence'
vector(rep) dep_clay_cg_azca
vector(rep) dep_clay_cg_azfl
vector(rep) dep_clay_cg_aznv
vector(rep) dep_clay_cg_cafl
vector(rep) dep_clay_cg_canv
vector(rep) dep_clay_cg_flnv
```

```
vector(rep) dep_gum_cg_azca
vector(rep) dep_gum_cg_azfl
vector(rep) dep_gum_cg_aznv
vector(rep) dep_gum_cg_cafl
vector(rep) dep_gum_cg_canv
vector(rep) dep_gum_cg_flnv

'block bootstrap approach'
For !i = 1 to rep
'make sure the sample is back to the original sample size'
smpl 1975Q2 2009Q1

input.resample(block=20) az_percent_B ca_percent_B fl_percent_B nv_percent_B

'AR(1)-GARCH(1,1) model and the corresponding conditional variance of the error term are
    calculated. Then the eventual corrected filtered price changes are determined'

'Arizona'
equation eqAZ.arch az_percent_B az_percent_B(-1) AR(1)
series az_resid = resid
eqAZ.makegarch AZcvar
series az_filter = (az_resid) / @sqrt(AZcvar)

'California'
equation eqCA.arch ca_percent_B ca_percent_B(-1) AR(1)
series ca_resid = resid
eqAZ.makegarch CAcvar
series ca_filter = (ca_resid) / @sqrt(CAcvar)

'Florida'
equation eqFL.arch fl_percent_B fl_percent_B(-1) AR(1)
series fl_resid = resid
eqFL.makegarch FLcvar
series fl_filter = (fl_resid) / @sqrt(FLcvar)

'Nevada'
equation eqNV.arch nv_percent_B nv_percent_B(-1) AR(1)
```

```
series nv_resid = resid
eqNV.makegarch NVcvar
series nv_filter = (nv_resid) / @sqrt(NVcvar)

'CDFs of the marginals'
series cdf_az = @cnorm(az_filter)
series cdf_ca = @cnorm(ca_filter)
series cdf_fl = @cnorm(fl_filter)
series cdf_nv = @cnorm(nv_filter)

'PDFs of the marginals'
series pdf_az = @dnorm(az_filter)
series pdf_ca = @dnorm(ca_filter)
series pdf_fl = @dnorm(fl_filter)
series pdf_nv = @dnorm(nv_filter)

'-------------------------------------------------------------------------------------'

'This program calculates the restricted coefficients of the mixture copulas for each
    state pair, which are then converted to the theta and then to Kendall`s Tau. To avoid
     errors in the parameter statement, eviews starts the estimation with the values
    specified in coefficient vector c'

'-------------------------------------------------------------------------------------'
'Clayton-Gumbel mixture AZ-CA'
c(1) = clayton_az_ca
c(2) = gumbel_az_ca
c(3) = -1

logl logl_Clay_Gum_az_ca
logl_Clay_Gum_az_ca.append @logl logl31
logl_Clay_Gum_az_ca.append f1=pdf_az
logl_Clay_Gum_az_ca.append f2=pdf_ca
logl_Clay_Gum_az_ca.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_az*cdf_ca)
    ^(-(c(1)^2)-1) * ( cdf_ca^(-(c(1)^2)) + cdf_az^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2) + (1-
    exp(c(3))/(1+exp(c(3)))) * exp(-((-log(cdf_az))^(exp(c(2))+1)+(-log(cdf_ca))^(exp(c
    (2))+1))^(1/(exp(c(2))+1))) * (cdf_az*cdf_ca)^-1 * (-log(cdf_az)*-log(cdf_ca))^((exp(
```

```
    c(2))+1)-1) * ((-log(cdf_az))^(exp(c(2))+1) + (-log(cdf_ca))^(exp(c(2))+1))^(-2+1/(
    exp(c(2))+1)) * (((-log(cdf_az))^(exp(c(2))+1)+(-log(cdf_ca))^(exp(c(2))+1))^(1/(exp(
    c(2))+1))+(exp(c(2))+1)-1) )*f1 *f2
logl_Clay_Gum_az_ca.append logl31 = log(term1)
smpl 1976q1 2009q1
logl_Clay_Gum_az_ca.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_Gum_az_ca_CLAY = logl_Clay_Gum_az_ca.@coef(2)
scalar Clay_Gum_az_ca_GUM = logl_Clay_Gum_az_ca.@coef(3)
scalar Clay_Gum_az_ca_PI = logl_Clay_Gum_az_ca.@coef(1)


'Clayton-Gumbel mixture AZ-FL'
c(1) = clayton_az_fl
c(2) = gumbel_az_fl
c(3) = -1

logl logl_Clay_Gum_az_fl
logl_Clay_Gum_az_fl.append @logl logl32
logl_Clay_Gum_az_fl.append f1=pdf_az
logl_Clay_Gum_az_fl.append f2=pdf_fl
logl_Clay_Gum_az_fl.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_az*cdf_fl)
    ^(-(c(1)^2)-1) * ( cdf_fl^(-(c(1)^2)) + cdf_az^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2) + (1-
    exp(c(3))/(1+exp(c(3)))) * exp(-((-log(cdf_az))^(exp(c(2))+1)+(-log(cdf_fl))^(exp(c
    (2))+1))^(1/(exp(c(2))+1))) * (cdf_az*cdf_fl)^-1 * (-log(cdf_az)*-log(cdf_fl))^((exp(
    c(2))+1)-1) * ((-log(cdf_az))^(exp(c(2))+1) + (-log(cdf_fl))^(exp(c(2))+1))^(-2+1/(
    exp(c(2))+1)) * (((-log(cdf_az))^(exp(c(2))+1)+(-log(cdf_fl))^(exp(c(2))+1))^(1/(exp(
    c(2))+1))+(exp(c(2))+1)-1) )*f1 *f2
logl_Clay_Gum_az_fl.append logl32 = log(term1)
smpl 1976q1 2009q1
logl_Clay_Gum_az_fl.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_Gum_az_fl_CLAY = logl_Clay_Gum_az_fl.@coef(2)
```

```
scalar Clay_Gum_az_fl_GUM = logl_Clay_Gum_az_fl.@coef(3)
scalar Clay_Gum_az_fl_PI = logl_Clay_Gum_az_fl.@coef(1)


'Clayton-Gumbel mixture AZ-NV'
c(1) = clayton_az_nv
c(2) = gumbel_az_nv
c(3) = -1

logl logl_Clay_Gum_az_nv
logl_Clay_Gum_az_nv.append @logl logl33
logl_Clay_Gum_az_nv.append f1=pdf_az
logl_Clay_Gum_az_nv.append f2=pdf_nv
logl_Clay_Gum_az_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_az*cdf_nv)
    ^(-(c(1)^2)-1) * ( cdf_nv^(-(c(1)^2)) + cdf_az^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2) + (1-
    exp(c(3))/(1+exp(c(3)))) * exp(-((-log(cdf_az))^(exp(c(2))+1)+(-log(cdf_nv))^(exp(c
    (2))+1))^(1/(exp(c(2))+1))) * (cdf_az*cdf_nv)^-1 * (-log(cdf_az)*-log(cdf_nv))^((exp(
    c(2))+1)-1) * ((-log(cdf_az))^(exp(c(2))+1) + (-log(cdf_nv))^(exp(c(2))+1))^(-2+1/(
    exp(c(2))+1)) * (((-log(cdf_az))^(exp(c(2))+1)+(-log(cdf_nv))^(exp(c(2))+1))^(1/(exp(
    c(2))+1))+(exp(c(2))+1)-1) )*f1 *f2
logl_Clay_Gum_az_nv.append logl33 = log(term1)
smpl 1976q1 2009q1
logl_Clay_Gum_az_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_Gum_az_nv_CLAY = logl_Clay_Gum_az_nv.@coef(2)
scalar Clay_Gum_az_nv_GUM = logl_Clay_Gum_az_nv.@coef(3)
scalar Clay_Gum_az_nv_PI = logl_Clay_Gum_az_nv.@coef(1)


'Clayton-Gumbel mixture CA-FL'
c(1) = clayton_ca_fl
c(2) = gumbel_ca_fl
c(3) = -1

logl logl_Clay_Gum_ca_fl
```

```
logl_Clay_Gum_ca_fl.append @logl logl34
logl_Clay_Gum_ca_fl.append f1=pdf_ca
logl_Clay_Gum_ca_fl.append f2=pdf_fl
logl_Clay_Gum_ca_fl.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_ca*cdf_fl)
    ^(-(c(1)^2)-1) * ( cdf_fl^(-(c(1)^2)) + cdf_ca^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2) + (1-
    exp(c(3))/(1+exp(c(3)))) * exp(-((-log(cdf_ca))^(exp(c(2))+1)+(-log(cdf_fl))^(exp(c
    (2))+1))^(1/(exp(c(2))+1))) * (cdf_ca*cdf_fl)^-1 * (-log(cdf_ca)*-log(cdf_fl))^((exp(
    c(2))+1)-1) * ((-log(cdf_ca))^(exp(c(2))+1) + (-log(cdf_fl))^(exp(c(2))+1))^(-2+1/(
    exp(c(2))+1)) * (((-log(cdf_ca))^(exp(c(2))+1)+(-log(cdf_fl))^(exp(c(2))+1))^(1/(exp(
    c(2))+1))+(exp(c(2))+1)-1) )*f1 *f2
logl_Clay_Gum_ca_fl.append logl34 = log(term1)
smpl 1976q1 2009q1
logl_Clay_Gum_ca_fl.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_Gum_ca_fl_CLAY = logl_Clay_Gum_ca_fl.@coef(2)
scalar Clay_Gum_ca_fl_GUM = logl_Clay_Gum_ca_fl.@coef(3)
scalar Clay_Gum_ca_fl_PI = logl_Clay_Gum_ca_fl.@coef(1)


'Clayton-Gumbel mixture CA-NV'
c(1) = clayton_ca_nv
c(2) = gumbel_ca_nv
c(3) = -1

logl logl_Clay_Gum_ca_nv
logl_Clay_Gum_ca_nv.append @logl logl35
logl_Clay_Gum_ca_nv.append f1=pdf_ca
logl_Clay_Gum_ca_nv.append f2=pdf_nv
logl_Clay_Gum_ca_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_ca*cdf_nv)
    ^(-(c(1)^2)-1) * ( cdf_nv^(-(c(1)^2)) + cdf_ca^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2) + (1-
    exp(c(3))/(1+exp(c(3)))) * exp(-((-log(cdf_ca))^(exp(c(2))+1)+(-log(cdf_nv))^(exp(c
    (2))+1))^(1/(exp(c(2))+1))) * (cdf_ca*cdf_nv)^-1 * (-log(cdf_ca)*-log(cdf_nv))^((exp(
    c(2))+1)-1) * ((-log(cdf_ca))^(exp(c(2))+1) + (-log(cdf_nv))^(exp(c(2))+1))^(-2+1/(
    exp(c(2))+1)) * (((-log(cdf_ca))^(exp(c(2))+1)+(-log(cdf_nv))^(exp(c(2))+1))^(1/(exp(
    c(2))+1))+(exp(c(2))+1)-1) )*f1 *f2
```

```
logl_Clay_Gum_ca_nv.append logl35 = log(term1)
smpl 1976q1 2009q1
logl_Clay_Gum_ca_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_Gum_ca_nv_CLAY = logl_Clay_Gum_ca_nv.@coef(2)
scalar Clay_Gum_ca_nv_GUM = logl_Clay_Gum_ca_nv.@coef(3)
scalar Clay_Gum_ca_nv_PI = logl_Clay_Gum_ca_nv.@coef(1)


'Clayton-Gumbel mixture FL-NV'
c(1) = clayton_fl_nv
c(2) = gumbel_fl_nv
c(3) = -1

logl logl_Clay_Gum_fl_nv
logl_Clay_Gum_fl_nv.append @logl logl36
logl_Clay_Gum_fl_nv.append f1=pdf_fl
logl_Clay_Gum_fl_nv.append f2=pdf_nv
logl_Clay_Gum_fl_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_fl*cdf_nv)
    ^(-(c(1)^2)-1) * ( cdf_nv^(-(c(1)^2)) + cdf_fl^(-(c(1)^2)) -1)^(-1/(c(1)^2) -2) + (1-
    exp(c(3))/(1+exp(c(3)))) * exp(-((-log(cdf_fl))^(exp(c(2))+1)+(-log(cdf_nv))^(exp(c
    (2))+1))^(1/(exp(c(2))+1))) * (cdf_fl*cdf_nv)^-1 * (-log(cdf_fl)*-log(cdf_nv))^((exp(
    c(2))+1)-1) * ((-log(cdf_fl))^(exp(c(2))+1) + (-log(cdf_nv))^(exp(c(2))+1))^(-2+1/(
    exp(c(2))+1)) * (((-log(cdf_fl))^(exp(c(2))+1)+(-log(cdf_nv))^(exp(c(2))+1))^(1/(exp(
    c(2))+1))+(exp(c(2))+1)-1) )*f1 *f2
logl_Clay_Gum_fl_nv.append logl36 = log(term1)
smpl 1976q1 2009q1
logl_Clay_Gum_fl_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_Gum_fl_nv_CLAY = logl_Clay_Gum_fl_nv.@coef(2)
scalar Clay_Gum_fl_nv_GUM = logl_Clay_Gum_fl_nv.@coef(3)
scalar Clay_Gum_fl_nv_PI = logl_Clay_Gum_fl_nv.@coef(1)
```

```
'-------------------------------------------------------------------------------------'
'Since the obtained coefficient values are made by incorporating the constraint on theta,
     we still have to obtain the actual theta'

'Clayton-Gumbel mixture'
theta_clay_CG_AZCA(!i) = Clay_Gum_az_ca_CLAY^2
theta_gum_CG_AZCA(!i) = exp(Clay_Gum_az_ca_GUM)+1
pi_CG_AZCA(!i) = exp(Clay_Gum_az_ca_PI)/(1+exp(Clay_Gum_az_ca_PI))

theta_clay_CG_AZFL(!i) = Clay_Gum_az_fl_CLAY^2
theta_gum_CG_AZFL(!i) = exp(Clay_Gum_az_fl_GUM)+1
pi_CG_AZFL(!i) = exp(Clay_Gum_az_fl_PI)/(1+exp(Clay_Gum_az_fl_PI))

theta_clay_CG_AZNV(!i) = Clay_Gum_az_nv_CLAY^2
theta_gum_CG_AZNV(!i) = exp(Clay_Gum_az_nv_GUM)+1
pi_CG_AZNV(!i) = exp(Clay_Gum_az_nv_PI)/(1+exp(Clay_Gum_az_nv_PI))

theta_clay_CG_CAFL(!i) = Clay_Gum_ca_fl_CLAY^2
theta_gum_CG_CAFL(!i) = exp(Clay_Gum_ca_fl_GUM)+1
pi_CG_CAFL(!i) = exp(Clay_Gum_ca_fl_PI)/(1+exp(Clay_Gum_ca_fl_PI))

theta_clay_CG_CANV(!i) = Clay_Gum_ca_nv_CLAY^2
theta_gum_CG_CANV(!i) = exp(Clay_Gum_ca_nv_GUM)+1
pi_CG_CANV(!i) = exp(Clay_Gum_ca_nv_PI)/(1+exp(Clay_Gum_ca_nv_PI))

theta_clay_CG_FLNV(!i) = Clay_Gum_fl_nv_CLAY^2
theta_gum_CG_FLNV(!i) = exp(Clay_Gum_fl_nv_GUM)+1
pi_CG_FLNV(!i) = exp(Clay_Gum_fl_nv_PI)/(1+exp(Clay_Gum_fl_nv_PI))

'-------------------------------------------------------------------------------------'

'Calculating Kendall`s tau'
TAU_CLAY_CG_AZCA(!i) = THETA_CLAY_CG_AZCA(!i)/(THETA_CLAY_CG_AZCA(!i)+2)
TAU_CLAY_CG_AZFL(!i) = THETA_CLAY_CG_AZFL(!i) /(THETA_CLAY_CG_AZFL(!i) +2)
TAU_CLAY_CG_AZNV(!i) = THETA_CLAY_CG_AZNV(!i) / (THETA_CLAY_CG_AZNV(!i) +2)
TAU_CLAY_CG_CAFL(!i) = THETA_CLAY_CG_CAFL(!i) / (THETA_CLAY_CG_CAFL(!i) +2)
TAU_CLAY_CG_CANV(!i) = THETA_CLAY_CG_CANV(!i) / (THETA_CLAY_CG_CANV(!i) +2)
```

```
TAU_CLAY_CG_FLNV(!i) = THETA_CLAY_CG_FLNV(!i) / (THETA_CLAY_CG_FLNV(!i) +2)
TAU_GUM_CG_AZCA(!i) = 1-1/THETA_GUM_CG_AZCA(!i)
TAU_GUM_CG_AZFL(!i) = 1-1/THETA_GUM_CG_AZFL(!i)
TAU_GUM_CG_AZNV(!i) = 1-1/THETA_GUM_CG_AZNV(!i)
TAU_GUM_CG_CAFL(!i) = 1-1/THETA_GUM_CG_CAFL(!i)
TAU_GUM_CG_CANV(!i) = 1-1/THETA_GUM_CG_CANV(!i)
TAU_GUM_CG_FLNV (!i) = 1-1/THETA_GUM_CG_FLNV(!i)

'Calculating the tail dependence'
dep_clay_cg_azca(!i) = 2^(-1/THETA_CLAY_CG_AZCA(!i))
dep_clay_cg_azfl(!i) = 2^(-1/THETA_CLAY_CG_AZFL(!i))
dep_clay_cg_aznv(!i) = 2^(-1/THETA_CLAY_CG_AZNV(!i))
dep_clay_cg_cafl(!i) = 2^(-1/THETA_CLAY_CG_CAFL(!i))
dep_clay_cg_canv(!i) = 2^(-1/THETA_CLAY_CG_CANV(!i))
dep_clay_cg_flnv(!i) = 2^(-1/THETA_CLAY_CG_FLNV(!i))
dep_gum_cg_azca(!i) = 2-2^(1/THETA_GUM_CG_AZCA(!i))
dep_gum_cg_azfl(!i) = 2-2^(1/THETA_GUM_CG_AZFL(!i))
dep_gum_cg_aznv(!i) = 2-2^(1/THETA_GUM_CG_AZNV(!i))
dep_gum_cg_cafl(!i) = 2-2^(1/THETA_GUM_CG_CAFL(!i))
dep_gum_cg_canv(!i) = 2-2^(1/THETA_GUM_CG_CANV(!i))
dep_gum_cg_flnv(!i) = 2-2^(1/THETA_GUM_CG_FLNV(!i))

Next

'Calculating the standard deviations of the variables'
scalar STD_tau_clay_cg_azca = @stdev(tau_clay_cg_azca)
scalar STD_tau_clay_cg_azfl = @stdev(tau_clay_cg_azfl)
scalar STD_tau_clay_cg_aznv = @stdev(tau_clay_cg_aznv)
scalar STD_tau_clay_cg_cafl = @stdev(tau_clay_cg_cafl)
scalar STD_tau_clay_cg_canv = @stdev(tau_clay_cg_canv)
scalar STD_tau_clay_cg_flnv = @stdev(tau_clay_cg_flnv)

scalar STD_tau_gum_cg_azca = @stdev(tau_gum_cg_azca)
scalar STD_tau_gum_cg_azfl = @stdev(tau_gum_cg_azfl)
scalar STD_tau_gum_cg_aznv = @stdev(tau_gum_cg_aznv)
scalar STD_tau_gum_cg_cafl = @stdev(tau_gum_cg_cafl)
scalar STD_tau_gum_cg_canv = @stdev(tau_gum_cg_canv)
```

```
scalar STD_tau_gum_cg_flnv = @stdev(tau_gum_cg_flnv)

scalar STD_pi_CG_azca = @stdev(pi_CG_azca)
scalar STD_pi_CG_azfl = @stdev(pi_CG_azfl)
scalar STD_pi_CG_aznv = @stdev(pi_CG_aznv)
scalar STD_pi_CG_cafl = @stdev(pi_CG_cafl)
scalar STD_pi_CG_canv = @stdev(pi_CG_canv)
scalar STD_pi_CG_flnv = @stdev(pi_CG_flnv)

scalar STD_dep_clay_cg_azca = @stdev(dep_clay_cg_azca)
scalar STD_dep_clay_cg_azfl = @stdev(dep_clay_cg_azfl)
scalar STD_dep_clay_cg_aznv = @stdev(dep_clay_cg_aznv)
scalar STD_dep_clay_cg_cafl = @stdev(dep_clay_cg_cafl)
scalar STD_dep_clay_cg_canv = @stdev(dep_clay_cg_canv)
scalar STD_dep_clay_cg_flnv = @stdev(dep_clay_cg_flnv)
scalar STD_dep_gum_cg_azca = @stdev(dep_gum_cg_azca)
scalar STD_dep_gum_cg_azfl = @stdev(dep_gum_cg_azfl)
scalar STD_dep_gum_cg_aznv = @stdev(dep_gum_cg_aznv)
scalar STD_dep_gum_cg_cafl = @stdev(dep_gum_cg_cafl)
scalar STD_dep_gum_cg_canv = @stdev(dep_gum_cg_canv)
scalar STD_dep_gum_cg_flnv = @stdev(dep_gum_cg_flnv)
'-------------------------------------------------------------------------------------'

'Block bootstrap approach Clayton-survival Clayton'

'Set the sample the same as in Zimmer (2012)'
smpl 1975Q2 2009Q1

'create a group such that we can resample the input all at the same time'
group input az_percent ca_percent fl_percent nv_percent

'set seed for random number generator'
rndseed 10006

scalar rep=500 'number of replications'

'Initialise the vectors'
```

```
'theta'
vector(rep) theta_clay_CSC_AZCA
vector(rep) theta_survClay_CSC_AZCA
vector(rep) pi_CSC_AZCA

vector(rep) theta_clay_CSC_AZFL
vector(rep) theta_survClay_CSC_AZFL
vector(rep) pi_CSC_AZFL

vector(rep) theta_clay_CSC_AZNV
vector(rep) theta_survClay_CSC_AZNV
vector(rep) pi_CSC_AZNV

vector(rep) theta_clay_CSC_CAFL
vector(rep) theta_survClay_CSC_CAFL
vector(rep) pi_CSC_CAFL

vector(rep) theta_clay_CSC_CANV
vector(rep) theta_survClay_CSC_CANV
vector(rep) pi_CSC_CANV

vector(rep) theta_clay_CSC_FLNV
vector(rep) theta_survClay_CSC_FLNV
vector(rep) pi_CSC_FLNV

'---------------------------------------------------------------------------------------'

'Kendall`s tau'
vector(rep) TAU_CLAY_CSC_AZCA
vector(rep) TAU_CLAY_CSC_AZFL
vector(rep) TAU_CLAY_CSC_AZNV
vector(rep) TAU_CLAY_CSC_CAFL
vector(rep) TAU_CLAY_CSC_CANV
vector(rep) TAU_CLAY_CSC_FLNV
vector(rep) TAU_SURVCLAY_CSC_AZCA
vector(rep) TAU_SURVCLAY_CSC_AZFL
vector(rep) TAU_SURVCLAY_CSC_AZNV
```

```
vector(rep) TAU_SURVCLAY_CSC_CAFL
vector(rep) TAU_SURVCLAY_CSC_CANV
vector(rep) TAU_SURVCLAY_CSC_FLNV

'Tail dependence'
vector(rep) dep_clay_csc_azca
vector(rep) dep_clay_csc_azfl
vector(rep) dep_clay_csc_aznv
vector(rep) dep_clay_csc_cafl
vector(rep) dep_clay_csc_canv
vector(rep) dep_clay_csc_flnv
vector(rep) dep_survclay_csc_azca
vector(rep) dep_survclay_csc_azfl
vector(rep) dep_survclay_csc_aznv
vector(rep) dep_survclay_csc_cafl
vector(rep) dep_survclay_csc_canv
vector(rep) dep_survclay_csc_flnv

'block bootstrap approach'
For !i = 1 to rep
'make sure the sample is back to the original sample size'
smpl 1975Q2 2009Q1

input.resample(block=20) az_percent_B ca_percent_B fl_percent_B nv_percent_B

'AR(1)-GARCH(1,1) model and the corresponding conditional variance of the error term are
    calculated. Then the eventual corrected filtered price changes are determined'

'Arizona'
equation eqAZ.arch az_percent_B az_percent_B(-1) AR(1)
series az_resid = resid
eqAZ.makegarch AZcvar
series az_filter = (az_resid) / @sqrt(AZcvar)

'California'
equation eqCA.arch ca_percent_B ca_percent_B(-1) AR(1)
series ca_resid = resid
```

```
eqAZ.makegarch CAcvar
series ca_filter = (ca_resid) / @sqrt(CAcvar)

'Florida'
equation eqFL.arch fl_percent_B fl_percent_B(-1) AR(1)
series fl_resid = resid
eqFL.makegarch FLcvar
series fl_filter = (fl_resid) / @sqrt(FLcvar)

'Nevada'
equation eqNV.arch nv_percent_B nv_percent_B(-1) AR(1)
series nv_resid = resid
eqNV.makegarch NVcvar
series nv_filter = (nv_resid) / @sqrt(NVcvar)

'CDFs of the marginals'
series cdf_az = @cnorm(az_filter)
series cdf_ca = @cnorm(ca_filter)
series cdf_fl = @cnorm(fl_filter)
series cdf_nv = @cnorm(nv_filter)

'PDFs of the marginals'
series pdf_az = @dnorm(az_filter)
series pdf_ca = @dnorm(ca_filter)
series pdf_fl = @dnorm(fl_filter)
series pdf_nv = @dnorm(nv_filter)
'--------------------------------------------------------------------------------------'
'to avoid errors in the parameter statement, eviews starts the estimation with the values
      specified in coefficient vector c'

'Clayton-survival Clayton mixture AZ-CA'
c(1) = clayton_az_ca
c(2) = survival_clayton_az_ca
c(3) = -1

logl logl_Clay_surv_Clay_az_ca
logl_Clay_surv_Clay_az_ca.append @logl logl51
```

```
logl_Clay_surv_Clay_az_ca.append f1=pdf_az
logl_Clay_surv_Clay_az_ca.append f2=pdf_ca
logl_Clay_surv_Clay_az_ca.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_ca*
     cdf_az)^(-(c(1)^2)-1) * ( cdf_az^(-(c(1)^2)) + cdf_ca^(-(c(1)^2)) -1)^(-1/(c(1)^2)
     -2) + (1-exp(c(3))/(1+exp(c(3)))) * (c(2)^2)*(1/(c(2)^2)+1) * ((1-cdf_ca)*(1-cdf_az))
     ^(-(c(2)^2)-1) * ( (1-cdf_ca)^(-(c(2)^2)) + (1-cdf_az)^(-(c(2)^2))-1 )^(-1/(c(2)^2)
     -2) )*f1*f2
logl_Clay_surv_Clay_az_ca.append logl51 = log(term1)
smpl 1976q1 2009q1
logl_Clay_surv_Clay_az_ca.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
     in the code.'
scalar Clay_surv_clay_az_ca_CLAY = logl_Clay_surv_Clay_az_ca.@coef(2)
scalar Clay_surv_clay_az_ca_surv_clay = logl_Clay_surv_Clay_az_ca.@coef(3)
scalar Clay_surv_clay_az_ca_PI = logl_Clay_surv_Clay_az_ca.@coef(1)


'Clayton-survival Clayton mixture AZ-FL'
c(1) = clayton_az_fl
c(2) = survival_clayton_az_fl
c(3) = -1

logl logl_Clay_surv_Clay_az_fl
logl_Clay_surv_Clay_az_fl.append @logl logl52
logl_Clay_surv_Clay_az_fl.append f1=pdf_az
logl_Clay_surv_Clay_az_fl.append f2=pdf_fl
logl_Clay_surv_Clay_az_fl.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_fl*
     cdf_az)^(-(c(1)^2)-1) * ( cdf_az^(-(c(1)^2)) + cdf_fl^(-(c(1)^2)) -1)^(-1/(c(1)^2)
     -2) + (1-exp(c(3))/(1+exp(c(3)))) * (c(2)^2)*(1/(c(2)^2)+1) * ((1-cdf_fl)*(1-cdf_az))
     ^(-(c(2)^2)-1) * ( (1-cdf_fl)^(-(c(2)^2)) + (1-cdf_az)^(-(c(2)^2))-1 )^(-1/(c(2)^2)
     -2) )*f1*f2
logl_Clay_surv_Clay_az_fl.append logl52 = log(term1)
smpl 1976q1 2009q1
logl_Clay_surv_Clay_az_fl.ml(showstart)
```

```
'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_surv_clay_az_fl_CLAY = logl_Clay_surv_Clay_az_fl.@coef(2)
scalar Clay_surv_clay_az_fl_surv_clay = logl_Clay_surv_Clay_az_fl.@coef(3)
scalar Clay_surv_clay_az_fl_PI = logl_Clay_surv_Clay_az_fl.@coef(1)


'Clayton-survival Clayton mixture AZ-NV'
c(1) = clayton_az_nv
c(2) = survival_clayton_az_nv
c(3) = -1

logl logl_Clay_surv_Clay_az_nv
logl_Clay_surv_Clay_az_nv.append @logl logl53
logl_Clay_surv_Clay_az_nv.append f1=pdf_az
logl_Clay_surv_Clay_az_nv.append f2=pdf_nv
logl_Clay_surv_Clay_az_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_nv*
    cdf_az)^(-(c(1)^2)-1) * ( cdf_az^(-(c(1)^2)) + cdf_nv^(-(c(1)^2)) -1)^(-1/(c(1)^2)
    -2) + (1-exp(c(3))/(1+exp(c(3)))) * (c(2)^2)*(1/(c(2)^2)+1) * ((1-cdf_nv)*(1-cdf_az))
    ^(-(c(2)^2)-1) * ( (1-cdf_nv)^(-(c(2)^2)) + (1-cdf_az)^(-(c(2)^2))-1 )^(-1/(c(2)^2)
    -2) )*f1*f2
logl_Clay_surv_Clay_az_nv.append logl53 = log(term1)
smpl 1976q1 2009q1
logl_Clay_surv_Clay_az_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_surv_clay_az_nv_CLAY = logl_Clay_surv_Clay_az_nv.@coef(2)
scalar Clay_surv_clay_az_nv_surv_clay = logl_Clay_surv_Clay_az_nv.@coef(3)
scalar Clay_surv_clay_az_nv_PI = logl_Clay_surv_Clay_az_nv.@coef(1)


'Clayton-survival Clayton mixture CA-FL'
c(1) = clayton_ca_fl
c(2) = survival_clayton_ca_fl
c(3) = -1
```

```
logl logl_Clay_surv_Clay_ca_fl
logl_Clay_surv_Clay_ca_fl.append @logl logl54
logl_Clay_surv_Clay_ca_fl.append f1=pdf_ca
logl_Clay_surv_Clay_ca_fl.append f2=pdf_fl
logl_Clay_surv_Clay_ca_fl.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_fl*
     cdf_ca)^(-(c(1)^2)-1) * ( cdf_ca^(-(c(1)^2)) + cdf_fl^(-(c(1)^2)) -1)^(-1/(c(1)^2)
     -2) + (1-exp(c(3))/(1+exp(c(3)))) * (c(2)^2)*(1/(c(2)^2)+1) * ((1-cdf_fl)*(1-cdf_ca))
     ^(-(c(2)^2)-1) * ( (1-cdf_fl)^(-(c(2)^2)) + (1-cdf_ca)^(-(c(2)^2))-1 )^(-1/(c(2)^2)
     -2) )*f1*f2
logl_Clay_surv_Clay_ca_fl.append logl54 = log(term1)
smpl 1976q1 2009q1
logl_Clay_surv_Clay_ca_fl.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
     in the code.'
scalar Clay_surv_clay_ca_fl_CLAY = logl_Clay_surv_Clay_ca_fl.@coef(2)
scalar Clay_surv_clay_ca_fl_surv_clay = logl_Clay_surv_Clay_ca_fl.@coef(3)
scalar Clay_surv_clay_ca_fl_PI = logl_Clay_surv_Clay_ca_fl.@coef(1)


'Clayton-survival Clayton mixture CA-NV'
c(1) = clayton_ca_nv
c(2) = survival_clayton_ca_nv
c(3) = -1

logl logl_Clay_surv_Clay_ca_nv
logl_Clay_surv_Clay_ca_nv.append @logl logl55
logl_Clay_surv_Clay_ca_nv.append f1=pdf_ca
logl_Clay_surv_Clay_ca_nv.append f2=pdf_nv
logl_Clay_surv_Clay_ca_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_nv*
     cdf_ca)^(-(c(1)^2)-1) * ( cdf_ca^(-(c(1)^2)) + cdf_nv^(-(c(1)^2)) -1)^(-1/(c(1)^2)
     -2) + (1-exp(c(3))/(1+exp(c(3)))) * (c(2)^2)*(1/(c(2)^2)+1) * ((1-cdf_nv)*(1-cdf_ca))
     ^(-(c(2)^2)-1) * ( (1-cdf_nv)^(-(c(2)^2)) + (1-cdf_ca)^(-(c(2)^2))-1 )^(-1/(c(2)^2)
     -2) )*f1*f2
logl_Clay_surv_Clay_ca_nv.append logl55 = log(term1)
smpl 1976q1 2009q1
logl_Clay_surv_Clay_ca_nv.ml(showstart)
```

```
'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_surv_clay_ca_nv_CLAY = logl_Clay_surv_Clay_ca_nv.@coef(2)
scalar Clay_surv_clay_ca_nv_surv_clay = logl_Clay_surv_Clay_ca_nv.@coef(3)
scalar Clay_surv_clay_ca_nv_PI = logl_Clay_surv_Clay_ca_nv.@coef(1)

'Clayton-survival Clayton mixture FL-NV'
c(1) = clayton_fl_nv
c(2) = survival_clayton_fl_nv
c(3) = -1

logl logl_Clay_surv_Clay_fl_nv
logl_Clay_surv_Clay_fl_nv.append @logl logl56
logl_Clay_surv_Clay_fl_nv.append f1=pdf_fl
logl_Clay_surv_Clay_fl_nv.append f2=pdf_nv
logl_Clay_surv_Clay_fl_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*((c(1)^2)+1)*(cdf_nv*
    cdf_fl)^(-(c(1)^2)-1) * ( cdf_fl^(-(c(1)^2)) + cdf_nv^(-(c(1)^2)) -1)^(-1/(c(1)^2)
    -2) + (1-exp(c(3))/(1+exp(c(3)))) * (c(2)^2)*(1/(c(2)^2)+1) * ((1-cdf_nv)*(1-cdf_fl))
    ^(-(c(2)^2)-1) * ( (1-cdf_nv)^(-(c(2)^2)) + (1-cdf_fl)^(-(c(2)^2))-1 )^(-1/(c(2)^2)
    -2) )*f1*f2
logl_Clay_surv_Clay_fl_nv.append logl56 = log(term1)
smpl 1976q1 2009q1
logl_Clay_surv_Clay_fl_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Clay_surv_clay_fl_nv_CLAY = logl_Clay_surv_Clay_fl_nv.@coef(2)
scalar Clay_surv_clay_fl_nv_surv_clay = logl_Clay_surv_Clay_fl_nv.@coef(3)
scalar Clay_surv_clay_fl_nv_PI = logl_Clay_surv_Clay_fl_nv.@coef(1)

'-------------------------------------------------------------------------------------'
'Since the obtained coefficient values are made by incorporating the constraint on theta,
     we still have to obtain the actual theta'

'Clayton-survival Clayton mixture'
theta_clay_CSC_AZCA(!i) = Clay_surv_clay_az_ca_CLAY^2
```

```
theta_survClay_CSC_AZCA(!i) = Clay_surv_clay_az_ca_surv_clay^2
pi_CSC_AZCA(!i) = exp(Clay_surv_clay_az_ca_PI)/(1+exp(Clay_surv_clay_az_ca_PI))

theta_clay_CSC_AZFL(!i) = Clay_surv_clay_az_fl_CLAY^2
theta_survClay_CSC_AZFL(!i) = Clay_surv_clay_az_fl_surv_clay^2
pi_CSC_AZFL(!i) = exp(Clay_surv_clay_az_fl_PI)/(1+exp(Clay_surv_clay_az_fl_PI))

theta_clay_CSC_AZNV(!i) = Clay_surv_clay_az_nv_CLAY^2
theta_survClay_CSC_AZNV(!i) = Clay_surv_clay_az_nv_surv_clay^2
pi_CSC_AZNV(!i) = exp(Clay_surv_clay_az_nv_PI)/(1+exp(Clay_surv_clay_az_nv_PI))

theta_clay_CSC_CAFL(!i) = Clay_surv_clay_ca_fl_CLAY^2
theta_survClay_CSC_CAFL(!i) = Clay_surv_clay_ca_fl_surv_clay^2
pi_CSC_CAFL(!i) = exp(Clay_surv_clay_ca_fl_PI)/(1+exp(Clay_surv_clay_ca_fl_PI))

theta_clay_CSC_CANV(!i) = Clay_surv_clay_ca_nv_CLAY^2
theta_survClay_CSC_CANV(!i) = Clay_surv_clay_ca_nv_surv_clay^2
pi_CSC_CANV(!i) = exp(Clay_surv_clay_ca_nv_PI)/(1+exp(Clay_surv_clay_ca_nv_PI))

theta_clay_CSC_FLNV(!i) = Clay_surv_clay_fl_nv_CLAY^2
theta_survClay_CSC_FLNV(!i) = Clay_surv_clay_fl_nv_surv_clay^2
pi_CSC_FLNV(!i) = exp(Clay_surv_clay_fl_nv_PI)/(1+exp(Clay_surv_clay_fl_nv_PI))

'-------------------------------------------------------------------------------------'


'Calculating Kendall`s tau'
TAU_CLAY_CSC_AZCA(!i) = THETA_CLAY_CSC_AZCA(!i) / (THETA_CLAY_CSC_AZCA(!i) +2)
TAU_CLAY_CSC_AZFL(!i) = THETA_CLAY_CSC_AZFL(!i) / (THETA_CLAY_CSC_AZFL(!i) +2)
TAU_CLAY_CSC_AZNV(!i) = THETA_CLAY_CSC_AZNV(!i) / (THETA_CLAY_CSC_AZNV(!i)+2)
TAU_CLAY_CSC_CAFL(!i) = THETA_CLAY_CSC_CAFL(!i) / (THETA_CLAY_CSC_CAFL(!i) +2)
TAU_CLAY_CSC_CANV(!i) = THETA_CLAY_CSC_CANV(!i) / (THETA_CLAY_CSC_CANV(!i) +2)
TAU_CLAY_CSC_FLNV(!i) =THETA_CLAY_CSC_FLNV(!i) / (THETA_CLAY_CSC_FLNV(!i) +2)
TAU_SURVCLAY_CSC_AZCA(!i) = THETA_SURVCLAY_CSC_AZCA(!i)/(2+THETA_SURVCLAY_CSC_AZCA(!i))
TAU_SURVCLAY_CSC_AZFL(!i) = THETA_SURVCLAY_CSC_AZFL(!i) /(2+THETA_SURVCLAY_CSC_AZFL(!i))
TAU_SURVCLAY_CSC_AZNV (!i) = THETA_SURVCLAY_CSC_AZNV(!i) /(2+THETA_SURVCLAY_CSC_AZNV(!i))
TAU_SURVCLAY_CSC_CAFL(!i) = THETA_SURVCLAY_CSC_CAFL(!i) /(2+THETA_SURVCLAY_CSC_CAFL(!i))
TAU_SURVCLAY_CSC_CANV(!i) = THETA_SURVCLAY_CSC_CANV(!i) /(2+THETA_SURVCLAY_CSC_CANV(!i))
```

```
TAU_SURVCLAY_CSC_FLNV(!i) = THETA_SURVCLAY_CSC_FLNV(!i) /(2+THETA_SURVCLAY_CSC_FLNV(!i))

'Tail dependence'
dep_clay_csc_azca(!i) = 2^(-1/THETA_CLAY_CSC_AZCA(!i))
dep_clay_csc_azfl(!i) = 2^(-1/THETA_CLAY_CSC_AZFL(!i))
dep_clay_csc_aznv(!i) = 2^(-1/THETA_CLAY_CSC_AZNV(!i))
dep_clay_csc_cafl(!i) = 2^(-1/THETA_CLAY_CSC_CAFL(!i))
dep_clay_csc_canv(!i) = 2^(-1/THETA_CLAY_CSC_CANV(!i))
dep_clay_csc_flnv(!i) = 2^(-1/THETA_CLAY_CSC_FLNV(!i))
dep_survclay_csc_azca(!i) = 2^(-1/THETA_SURVCLAY_CSC_AZCA(!i))
dep_survclay_csc_azfl(!i) = 2^(-1/THETA_SURVCLAY_CSC_AZFL(!i))
dep_survclay_csc_aznv(!i) = 2^(-1/THETA_SURVCLAY_CSC_AZNV(!i))
dep_survclay_csc_cafl(!i) = 2^(-1/THETA_SURVCLAY_CSC_CAFL(!i))
dep_survclay_csc_canv(!i) = 2^(-1/THETA_SURVCLAY_CSC_CANV(!i))
dep_survclay_csc_flnv(!i) = 2^(-1/THETA_SURVCLAY_CSC_FLNV(!i))

Next


'Calculating the standard deviations of the variables'
scalar STD_tau_clay_csc_azca = @stdev(tau_clay_csc_azca)
scalar STD_tau_clay_csc_azfl = @stdev(tau_clay_csc_azfl)
scalar STD_tau_clay_csc_aznv = @stdev(tau_clay_csc_aznv)
scalar STD_tau_clay_csc_cafl = @stdev(tau_clay_csc_cafl)
scalar STD_tau_clay_csc_canv = @stdev(tau_clay_csc_canv)
scalar STD_tau_clay_csc_flnv = @stdev(tau_clay_csc_flnv)

scalar STD_tau_survclay_csc_azca = @stdev(tau_survclay_csc_azca)
scalar STD_tau_survclay_csc_azfl = @stdev(tau_survclay_csc_azfl)
scalar STD_tau_survclay_csc_aznv = @stdev(tau_survclay_csc_aznv)
scalar STD_tau_survclay_csc_cafl = @stdev(tau_survclay_csc_cafl)
scalar STD_tau_survclay_csc_canv = @stdev(tau_survclay_csc_canv)
scalar STD_tau_survclay_csc_flnv = @stdev(tau_survclay_csc_flnv)

scalar STD_pi_CSC_azca = @stdev(pi_CSC_azca)
scalar STD_pi_CSC_azfl = @stdev(pi_CSC_azfl)
scalar STD_pi_CSC_aznv = @stdev(pi_CSC_aznv)
```

```
scalar STD_pi_CSC_cafl = @stdev(pi_CSC_cafl)
scalar STD_pi_CSC_canv = @stdev(pi_CSC_canv)
scalar STD_pi_CSC_flnv = @stdev(pi_CSC_flnv)

scalar STD_dep_clay_csc_azca = @stdev(dep_clay_csc_azca)
scalar STD_dep_clay_csc_azfl = @stdev(dep_clay_csc_azfl)
scalar STD_dep_clay_csc_aznv = @stdev(dep_clay_csc_aznv)
scalar STD_dep_clay_csc_cafl = @stdev(dep_clay_csc_cafl)
scalar STD_dep_clay_csc_canv = @stdev(dep_clay_csc_canv)
scalar STD_dep_clay_csc_flnv = @stdev(dep_clay_csc_flnv)
scalar STD_dep_survclay_csc_azca = @stdev(dep_survclay_csc_azca)
scalar STD_dep_survclay_csc_azfl = @stdev(dep_survclay_csc_azfl)
scalar STD_dep_survclay_csc_aznv = @stdev(dep_survclay_csc_aznv)
scalar STD_dep_survclay_csc_cafl = @stdev(dep_survclay_csc_cafl)
scalar STD_dep_survclay_csc_canv = @stdev(dep_survclay_csc_canv)
scalar STD_dep_survclay_csc_flnv = @stdev(dep_survclay_csc_flnv)
'-----------------------------------------------------------------------------------'

'Block bootstrap approach gumbel-survival gumbel'

'Set the sample the same as in Zimmer (2012)'
smpl 1975Q2 2009Q1

'create a group such that we can resample the input all at the same time'
group input az_percent ca_percent fl_percent nv_percent

'set seed for random number generator'
rndseed 10006

scalar rep=500 'number of replications'

'Initialise the vectors'
'theta'
vector(rep) theta_gum_GSG_AZCA
vector(rep) theta_survGum_GSG_AZCA
vector(rep) pi_GSG_AZCA
```

```
vector(rep) theta_gum_GSG_AZFL
vector(rep) theta_survGum_GSG_AZFL
vector(rep) pi_GSG_AZFL

vector(rep) theta_gum_GSG_AZNV
vector(rep) theta_survGum_GSG_AZNV
vector(rep) pi_GSG_AZNV

vector(rep) theta_gum_GSG_CAFL
vector(rep) theta_survGum_GSG_CAFL
vector(rep) pi_GSG_CAFL

vector(rep) theta_gum_GSG_CANV
vector(rep) theta_survGum_GSG_CANV
vector(rep) pi_GSG_CANV

vector(rep) theta_gum_GSG_FLNV
vector(rep) theta_survGum_GSG_FLNV
vector(rep) pi_GSG_FLNV
'-----------------------------------------------------------------------------------------'
'Kendall`s tau'
vector(rep) TAU_GUM_GSG_AZCA
vector(rep) TAU_GUM_GSG_AZFL
vector(rep) TAU_GUM_GSG_AZNV
vector(rep) TAU_GUM_GSG_CAFL
vector(rep) TAU_GUM_GSG_CANV
vector(rep) TAU_GUM_GSG_FLNV
vector(rep) TAU_SURVGUM_GSG_AZCA
vector(rep) TAU_SURVGUM_GSG_AZFL
vector(rep) TAU_SURVGUM_GSG_AZNV
vector(rep) TAU_SURVGUM_GSG_CAFL
vector(rep) TAU_SURVGUM_GSG_CANV
vector(rep) TAU_SURVGUM_GSG_FLNV

'Tail dependence'
vector(rep) dep_gum_gsg_azca
vector(rep) dep_gum_gsg_azfl
```

```
vector(rep) dep_gum_gsg_aznv
vector(rep) dep_gum_gsg_cafl
vector(rep) dep_gum_gsg_canv
vector(rep) dep_gum_gsg_flnv
vector(rep) dep_survgum_gsg_azca
vector(rep) dep_survgum_gsg_azfl
vector(rep) dep_survgum_gsg_aznv
vector(rep) dep_survgum_gsg_cafl
vector(rep) dep_survgum_gsg_canv
vector(rep) dep_survgum_gsg_flnv

'block bootstrap approach'
For !i = 1 to rep
'make sure the sample is back to the original sample size'
smpl 1975Q2 2009Q1

input.resample(block=20) az_percent_B ca_percent_B fl_percent_B nv_percent_B

'AR(1)-GARCH(1,1) model and the corresponding conditional variance of the error term are
     calculated. Then the eventual corrected filtered price changes are determined'

'Arizona'
equation eqAZ.arch az_percent_B az_percent_B(-1) AR(1)
series az_resid = resid
eqAZ.makegarch AZcvar
series az_filter = (az_resid) / @sqrt(AZcvar)

'California'
equation eqCA.arch ca_percent_B ca_percent_B(-1) AR(1)
series ca_resid = resid
eqAZ.makegarch CAcvar
series ca_filter = (ca_resid) / @sqrt(CAcvar)

'Florida'
equation eqFL.arch fl_percent_B fl_percent_B(-1) AR(1)
series fl_resid = resid
eqFL.makegarch FLcvar
```

```
series fl_filter = (fl_resid) / @sqrt(FLcvar)

'Nevada'
equation eqNV.arch nv_percent_B nv_percent_B(-1) AR(1)
series nv_resid = resid
eqNV.makegarch NVcvar
series nv_filter = (nv_resid) / @sqrt(NVcvar)

'CDFs of the marginals'
series cdf_az = @cnorm(az_filter)
series cdf_ca = @cnorm(ca_filter)
series cdf_fl = @cnorm(fl_filter)
series cdf_nv = @cnorm(nv_filter)

'PDFs of the marginals'
series pdf_az = @dnorm(az_filter)
series pdf_ca = @dnorm(ca_filter)
series pdf_fl = @dnorm(fl_filter)
series pdf_nv = @dnorm(nv_filter)

'-------------------------------------------------------------------------------------'
'to avoid errors in the parameter statement, eviews starts the estimation with the values
      specified in coefficient vector c'

'Gumbel-survival Gumbel mixture AZ-CA'
c(1) = Gumbel_az_ca
c(2) = survival_Gumbel_az_ca
c(3) = -1

logl logl_Gum_surv_Gum_az_ca
logl_Gum_surv_Gum_az_ca.append @logl logl71
logl_Gum_surv_Gum_az_ca.append f1=pdf_az
logl_Gum_surv_Gum_az_ca.append f2=pdf_ca
logl_Gum_surv_Gum_az_ca.append term1 = ( exp(c(3))/(1+exp(c(3)))* exp(-((-log(cdf_az))^(
     exp(c(1))+1)+(-log(cdf_ca))^(exp(c(1))+1))^(1/(exp(c(1))+1))) * (cdf_az*cdf_ca)^-1 *
     (-log(cdf_az)*-log(cdf_ca))^((exp(c(1))+1)-1) * ((-log(cdf_az))^(exp(c(1))+1) + (-log
     (cdf_ca))^(exp(c(1))+1))^(-2+1/(exp(c(1))+1)) * (((-log(cdf_az))^(exp(c(1))+1)+(-log(
```

```
    cdf_ca))^(exp(c(1))+1))^(1/(exp(c(1))+1))+(exp(c(1))+1)-1) + (1-exp(c(3))/(1+exp(c(3)
    ))) * ((exp(-((-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))
    ^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_ca))^(1/(
    exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp
    (c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log
    (1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((cdf_ca - 1)*(cdf_az - 1)) - (exp
    (-((-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2)
    )/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log
    (1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c
    (2))+1))))^((exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2)
    )+1)) - 1)*(-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c
    (2))+1))*(cdf_ca - 1)*(cdf_az - 1))) )*f1 *f2
logl_Gum_surv_Gum_az_ca.append logl71 = log(term1)
smpl 1976q1 2009q1
logl_Gum_surv_Gum_az_ca.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Gum_surv_Gum_az_ca_GUM = logl_Gum_surv_Gum_az_ca.@coef(2)
scalar Gum_surv_Gum_az_ca_surv_GUM = logl_Gum_surv_Gum_az_ca.@coef(3)
scalar Gum_surv_Gum_az_ca_PI = logl_Gum_surv_Gum_az_ca.@coef(1)



'Gumbel-survival Gumbel mixture AZ-FL'
c(1) = Gumbel_az_fl
c(2) = survival_Gumbel_az_fl
c(3) = -1

logl logl_Gum_surv_Gum_az_fl
logl_Gum_surv_Gum_az_fl.append @logl logl72
logl_Gum_surv_Gum_az_fl.append f1=pdf_az
logl_Gum_surv_Gum_az_fl.append f2=pdf_fl
logl_Gum_surv_Gum_az_fl.append term1 = ( exp(c(3))/(1+exp(c(3)))* exp(-((-log(cdf_az))^(
    exp(c(1))+1)+(-log(cdf_fl))^(exp(c(1))+1))^(1/(exp(c(1))+1))) * (cdf_az*cdf_fl)^-1 *
    (-log(cdf_az)*-log(cdf_fl))^((exp(c(1))+1)-1) * ((-log(cdf_az))^(exp(c(1))+1) + (-log
    (cdf_fl))^(exp(c(1))+1))^(-2+1/(exp(c(1))+1)) * (((-log(cdf_az))^(exp(c(1))+1)+(-log(
    cdf_fl))^(exp(c(1))+1))^(1/(exp(c(1))+1))+(exp(c(1))+1)-1) + (1-exp(c(3))/(1+exp(c(3)
```

```
    ))) * ((exp(-((-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))
    ^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_fl))^(1/(
    exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp
    (c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log
    (1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((cdf_fl - 1)*(cdf_az - 1)) - (exp
    (-((-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2)
    )/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log
    (1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c
    (2))+1))))^((exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2)
    )+1)) - 1)*(-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c
    (2))+1))*(cdf_fl - 1)*(cdf_az - 1))) )*f1 *f2
logl_Gum_surv_Gum_az_fl.append logl72 = log(term1)
smpl 1976q1 2009q1
logl_Gum_surv_Gum_az_fl.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Gum_surv_Gum_az_fl_GUM = logl_Gum_surv_Gum_az_fl.@coef(2)
scalar Gum_surv_Gum_az_fl_surv_GUM = logl_Gum_surv_Gum_az_fl.@coef(3)
scalar Gum_surv_Gum_az_fl_PI = logl_Gum_surv_Gum_az_fl.@coef(1)



'Gumbel-survival Gumbel mixture AZ-NV'
c(1) = Gumbel_az_nv
c(2) = survival_Gumbel_az_nv
c(3) = -1

logl logl_Gum_surv_Gum_az_nv
logl_Gum_surv_Gum_az_nv.append @logl logl73
logl_Gum_surv_Gum_az_nv.append f1=pdf_az
logl_Gum_surv_Gum_az_nv.append f2=pdf_nv
logl_Gum_surv_Gum_az_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))* exp(-((-log(cdf_az))^(
    exp(c(1))+1)+(-log(cdf_nv))^(exp(c(1))+1))^(1/(exp(c(1))+1))) * (cdf_az*cdf_nv)^-1 *
    (-log(cdf_az)*-log(cdf_nv))^((exp(c(1))+1)-1) * ((-log(cdf_az))^(exp(c(1))+1) + (-log
    (cdf_nv))^(exp(c(1))+1))^(-2+1/(exp(c(1))+1)) * (((-log(cdf_az))^(exp(c(1))+1)+(-log(
    cdf_nv))^(exp(c(1))+1))^(1/(exp(c(1))+1))+(exp(c(1))+1)-1) + (1-exp(c(3))/(1+exp(c(3)
    ))) * ((exp(-((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))
```

```
    ^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_nv))^(1/(
    exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp
    (c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log
    (1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((cdf_nv - 1)*(cdf_az - 1)) - (exp
    (-((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2)
    )/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log
    (1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c
    (2))+1))))^((exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2)
    )+1)) - 1)*(-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c
    (2))+1))*(cdf_nv - 1)*(cdf_az - 1))) )*f1 *f2
logl_Gum_surv_Gum_az_nv.append logl73 = log(term1)
smpl 1976q1 2009q1
logl_Gum_surv_Gum_az_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Gum_surv_Gum_az_nv_GUM = logl_Gum_surv_Gum_az_nv.@coef(2)
scalar Gum_surv_Gum_az_nv_surv_GUM = logl_Gum_surv_Gum_az_nv.@coef(3)
scalar Gum_surv_Gum_az_nv_PI = logl_Gum_surv_Gum_az_nv.@coef(1)


'Gumbel-survival Gumbel mixture CA-FL'
c(1) = Gumbel_ca_fl
c(2) = survival_Gumbel_ca_fl
c(3) = -1

logl logl_Gum_surv_Gum_ca_fl
logl_Gum_surv_Gum_ca_fl.append @logl logl74
logl_Gum_surv_Gum_ca_fl.append f1=pdf_ca
logl_Gum_surv_Gum_ca_fl.append f2=pdf_fl
logl_Gum_surv_Gum_ca_fl.append term1 = ( exp(c(3))/(1+exp(c(3)))* exp(-((-log(cdf_ca))^(
    exp(c(1))+1)+(-log(cdf_fl))^(exp(c(1))+1))^(1/(exp(c(1))+1))) * (cdf_ca*cdf_fl)^-1 *
    (-log(cdf_ca)*-log(cdf_fl))^((exp(c(1))+1)-1) * ((-log(cdf_ca))^(exp(c(1))+1) + (-log
    (cdf_fl))^(exp(c(1))+1))^(-2+1/(exp(c(1))+1)) * (((-log(cdf_ca))^(exp(c(1))+1)+(-log(
    cdf_fl))^(exp(c(1))+1))^(1/(exp(c(1))+1))+(exp(c(1))+1)-1) + (1-exp(c(3))/(1+exp(c(3)
    ))) * ((exp(-((-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))
    ^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_fl))^(1/(
```

```
    exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp
    (c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log
    (1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((cdf_fl - 1)*(cdf_ca - 1)) - (exp
    (-((-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2)
    )/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log
    (1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c
    (2))+1))))^((exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2)
    )+1)) - 1)*(-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c
    (2))+1))*(cdf_fl - 1)*(cdf_ca - 1))) )*f1 *f2
logl_Gum_surv_Gum_ca_fl.append logl74 = log(term1)
smpl 1976q1 2009q1
logl_Gum_surv_Gum_ca_fl.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Gum_surv_Gum_ca_fl_GUM = logl_Gum_surv_Gum_ca_fl.@coef(2)
scalar Gum_surv_Gum_ca_fl_surv_GUM = logl_Gum_surv_Gum_ca_fl.@coef(3)
scalar Gum_surv_Gum_ca_fl_PI = logl_Gum_surv_Gum_ca_fl.@coef(1)



'Gumbel-survival Gumbel mixture CA-NV'
c(1) = Gumbel_ca_nv
c(2) = survival_Gumbel_ca_nv
c(3) = -1

logl logl_Gum_surv_Gum_ca_nv
logl_Gum_surv_Gum_ca_nv.append @logl logl75
logl_Gum_surv_Gum_ca_nv.append f1=pdf_ca
logl_Gum_surv_Gum_ca_nv.append f2=pdf_nv
logl_Gum_surv_Gum_ca_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))* exp(-((-log(cdf_ca))^(
    exp(c(1))+1)+(-log(cdf_nv))^(exp(c(1))+1))^(1/(exp(c(1))+1))) * (cdf_ca*cdf_nv)^-1 *
    (-log(cdf_ca)*-log(cdf_nv))^((exp(c(1))+1)-1) * ((-log(cdf_ca))^(exp(c(1))+1) + (-log
    (cdf_nv))^(exp(c(1))+1))^(-2+1/(exp(c(1))+1)) * (((-log(cdf_ca))^(exp(c(1))+1)+(-log(
    cdf_nv))^(exp(c(1))+1))^(1/(exp(c(1))+1))+(exp(c(1))+1)-1) + (1-exp(c(3))/(1+exp(c(3)
    ))) * ((exp(-((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))
    ^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_nv))^(1/(
    exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp
```

```
    (c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log
    (1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((cdf_nv - 1)*(cdf_ca - 1)) - (exp
    (-((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2)
    )/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log
    (1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c
    (2))+1))))^((exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2)
    )+1)) - 1)*(-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c
    (2))+1))*(cdf_nv - 1)*(cdf_ca - 1))) )*f1 *f2
logl_Gum_surv_Gum_ca_nv.append logl75 = log(term1)
smpl 1976q1 2009q1
logl_Gum_surv_Gum_ca_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Gum_surv_Gum_ca_nv_GUM = logl_Gum_surv_Gum_ca_nv.@coef(2)
scalar Gum_surv_Gum_ca_nv_surv_GUM = logl_Gum_surv_Gum_ca_nv.@coef(3)
scalar Gum_surv_Gum_ca_nv_PI = logl_Gum_surv_Gum_ca_nv.@coef(1)


'Gumbel-survival Gumbel mixture FL-NV'
c(1) = Gumbel_fl_nv
c(2) = survival_Gumbel_fl_nv
c(3) = -1

logl logl_Gum_surv_Gum_fl_nv
logl_Gum_surv_Gum_fl_nv.append @logl logl76
logl_Gum_surv_Gum_fl_nv.append f1=pdf_fl
logl_Gum_surv_Gum_fl_nv.append f2=pdf_nv
logl_Gum_surv_Gum_fl_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))* exp(-((-log(cdf_fl))^(
    exp(c(1))+1)+(-log(cdf_nv))^(exp(c(1))+1))^(1/(exp(c(1))+1))) * (cdf_fl*cdf_nv)^-1 *
    (-log(cdf_fl)*-log(cdf_nv))^((exp(c(1))+1)-1) * ((-log(cdf_fl))^(exp(c(1))+1) + (-log
    (cdf_nv))^(exp(c(1))+1))^(-2+1/(exp(c(1))+1)) * (((-log(cdf_fl))^(exp(c(1))+1)+(-log(
    cdf_nv))^(exp(c(1))+1))^(1/(exp(c(1))+1))+(exp(c(1))+1)-1) + (1-exp(c(3))/(1+exp(c(3)
    ))) * ((exp(-((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_fl))
    ^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_nv))^(1/(
    exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp
    (c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log
```

```
    (1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((cdf_nv - 1)*(cdf_fl - 1)) - (exp
    (-((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_fl))^(1/(exp(c(2)
    )/(exp(c(2))+1))))^(exp(c(2))/(exp(c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log
    (1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c
    (2))+1))))^((exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2)
    )+1)) - 1)*(-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c
    (2))+1))*(cdf_nv - 1)*(cdf_fl - 1))) )*f1 *f2
logl_Gum_surv_Gum_fl_nv.append logl76 = log(term1)
smpl 1976q1 2009q1
logl_Gum_surv_Gum_fl_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar Gum_surv_Gum_fl_nv_GUM = logl_Gum_surv_Gum_fl_nv.@coef(2)
scalar Gum_surv_Gum_fl_nv_surv_GUM = logl_Gum_surv_Gum_fl_nv.@coef(3)
scalar Gum_surv_Gum_fl_nv_PI = logl_Gum_surv_Gum_fl_nv.@coef(1)

'-------------------------------------------------------------------------------------'
'Since the obtained coefficient values are made by incorporating the constraint on theta,
     we still have to obtain the actual theta'

'gumbel-survival gumbel mixture'
theta_gum_GSG_AZCA(!i) = exp(Gum_surv_Gum_az_ca_GUM)+1
theta_survGum_GSG_AZCA(!i) = exp(Gum_surv_Gum_az_ca_surv_GUM)/(1+exp(
    Gum_surv_Gum_az_ca_surv_GUM ))
pi_GSG_AZCA(!i) = exp(Gum_surv_Gum_az_ca_PI)/(1+exp(Gum_surv_Gum_az_ca_PI ))

theta_gum_GSG_AZFL(!i) = exp(Gum_surv_Gum_az_fl_GUM)+1
theta_survGum_GSG_AZFL(!i) = exp(Gum_surv_Gum_az_fl_surv_GUM)/(1+exp(
    Gum_surv_Gum_az_fl_surv_GUM ))
pi_GSG_AZFL(!i) = exp(Gum_surv_Gum_az_fl_PI)/(1+exp(Gum_surv_Gum_az_fl_PI ))

theta_gum_GSG_AZNV(!i) = exp(Gum_surv_Gum_az_nv_GUM)+1
theta_survGum_GSG_AZNV(!i) = exp(Gum_surv_Gum_az_nv_surv_GUM)/(1+exp(
    Gum_surv_Gum_az_nv_surv_GUM ))
pi_GSG_AZNV(!i) = exp(Gum_surv_Gum_az_nv_PI)/(1+exp(Gum_surv_Gum_az_nv_PI ))
```

```
theta_gum_GSG_CAFL(!i) = exp(Gum_surv_Gum_ca_fl_GUM)+1
theta_survGum_GSG_CAFL(!i) = exp(Gum_surv_Gum_ca_fl_surv_GUM)/(1+exp(
     Gum_surv_Gum_ca_fl_surv_GUM ))
pi_GSG_CAFL(!i) = exp(Gum_surv_Gum_ca_fl_PI)/(1+exp(Gum_surv_Gum_ca_fl_PI ))

theta_gum_GSG_CANV(!i) = exp(Gum_surv_Gum_ca_nv_GUM)+1
theta_survGum_GSG_CANV(!i) = exp(Gum_surv_Gum_ca_nv_surv_GUM)/(1+exp(
     Gum_surv_Gum_ca_nv_surv_GUM ))
pi_GSG_CANV(!i) = exp(Gum_surv_Gum_ca_nv_PI)/(1+exp(Gum_surv_Gum_ca_nv_PI ))

theta_gum_GSG_FLNV(!i) = exp(Gum_surv_Gum_fl_nv_GUM)+1
theta_survGum_GSG_FLNV(!i) = exp(Gum_surv_Gum_fl_nv_surv_GUM)/(1+exp(
     Gum_surv_Gum_fl_nv_surv_GUM ))
pi_GSG_FLNV(!i) = exp(Gum_surv_Gum_fl_nv_PI)/(1+exp(Gum_surv_Gum_fl_nv_PI ))

'-------------------------------------------------------------------------------------'

'Calculating Kendall`s tau'
TAU_GUM_GSG_AZCA(!i) = 1-1/THETA_GUM_GSG_AZCA(!i)
TAU_GUM_GSG_AZFL(!i) = 1-1/THETA_GUM_GSG_AZFL(!i)
TAU_GUM_GSG_AZNV(!i) = 1-1/THETA_GUM_GSG_AZNV(!i)
TAU_GUM_GSG_CAFL(!i) = 1-1/THETA_GUM_GSG_CAFL(!i)
TAU_GUM_GSG_CANV(!i) = 1-1/THETA_GUM_GSG_CANV(!i)
TAU_GUM_GSG_FLNV(!i) = 1-1/THETA_GUM_GSG_FLNV(!i)

TAU_SURVGUM_GSG_AZCA (!i) = 1-THETA_SURVGUM_GSG_AZCA(!i)
TAU_SURVGUM_GSG_AZFL (!i) = 1-THETA_SURVGUM_GSG_AZFL(!i)
TAU_SURVGUM_GSG_AZNV(!i) = 1-THETA_SURVGUM_GSG_AZNV(!i)
TAU_SURVGUM_GSG_CAFL(!i) = 1-THETA_SURVGUM_GSG_CAFL(!i)
TAU_SURVGUM_GSG_CANV(!i) = 1-THETA_SURVGUM_GSG_CANV(!i)
TAU_SURVGUM_GSG_FLNV(!i) = 1-THETA_SURVGUM_GSG_FLNV(!i)

'Tail dependence'
dep_gum_gsg_azca(!i) = 2-2^(1/THETA_GUM_GSG_AZCA(!i))
dep_gum_gsg_azfl(!i) = 2-2^(1/THETA_GUM_GSG_AZFL(!i))
dep_gum_gsg_aznv(!i) = 2-2^(1/THETA_GUM_GSG_AZNV(!i))
dep_gum_gsg_cafl(!i) = 2-2^(1/THETA_GUM_GSG_CAFL(!i))
```

```
dep_gum_gsg_canv(!i) = 2-2^(1/THETA_GUM_GSG_CANV(!i))
dep_gum_gsg_flnv(!i) = 2-2^(1/THETA_GUM_GSG_FLNV(!i))
dep_survgum_gsg_azca(!i) = 2-2^(THETA_SURVGUM_GSG_AZCA(!i))
dep_survgum_gsg_azfl(!i) = 2-2^(THETA_SURVGUM_GSG_AZFL(!i))
dep_survgum_gsg_aznv(!i) = 2-2^(THETA_SURVGUM_GSG_AZNV(!i))
dep_survgum_gsg_cafl(!i) = 2-2^(THETA_SURVGUM_GSG_CAFL(!i))
dep_survgum_gsg_canv(!i) = 2-2^(THETA_SURVGUM_GSG_CANV(!i))
dep_survgum_gsg_flnv(!i) = 2-2^(THETA_SURVGUM_GSG_FLNV(!i))

Next

'Calculating the standard deviations of the variables'
scalar STD_tau_gum_gsg_azca = @stdev(tau_gum_gsg_azca)
scalar STD_tau_gum_gsg_azfl = @stdev(tau_gum_gsg_azfl)
scalar STD_tau_gum_gsg_aznv = @stdev(tau_gum_gsg_aznv)
scalar STD_tau_gum_gsg_cafl = @stdev(tau_gum_gsg_cafl)
scalar STD_tau_gum_gsg_canv = @stdev(tau_gum_gsg_canv)
scalar STD_tau_gum_gsg_flnv = @stdev(tau_gum_gsg_flnv)

scalar STD_tau_survgum_gsg_azca = @stdev(tau_survgum_gsg_azca)
scalar STD_tau_survgum_gsg_azfl = @stdev(tau_survgum_gsg_azfl)
scalar STD_tau_survgum_gsg_aznv = @stdev(tau_survgum_gsg_aznv)
scalar STD_tau_survgum_gsg_cafl = @stdev(tau_survgum_gsg_cafl)
scalar STD_tau_survgum_gsg_canv = @stdev(tau_survgum_gsg_canv)
scalar STD_tau_survgum_gsg_flnv = @stdev(tau_survgum_gsg_flnv)

scalar STD_pi_GSG_azca = @stdev(pi_GSG_azca)
scalar STD_pi_GSG_azfl = @stdev(pi_GSG_azfl)
scalar STD_pi_GSG_aznv = @stdev(pi_GSG_aznv)
scalar STD_pi_GSG_cafl = @stdev(pi_GSG_cafl)
scalar STD_pi_GSG_canv = @stdev(pi_GSG_canv)
scalar STD_pi_GSG_flnv = @stdev(pi_GSG_flnv)

scalar STD_dep_gum_gsg_azca = @stdev(dep_gum_gsg_azca)
scalar STD_dep_gum_gsg_azfl = @stdev(dep_gum_gsg_azfl)
scalar STD_dep_gum_gsg_aznv = @stdev(dep_gum_gsg_aznv)
scalar STD_dep_gum_gsg_cafl = @stdev(dep_gum_gsg_cafl)
```

```
scalar STD_dep_gum_gsg_canv = @stdev(dep_gum_gsg_canv)
scalar STD_dep_gum_gsg_flnv = @stdev(dep_gum_gsg_flnv)
scalar STD_dep_survgum_gsg_azca = @stdev(dep_survgum_gsg_azca)
scalar STD_dep_survgum_gsg_azfl = @stdev(dep_survgum_gsg_azfl)
scalar STD_dep_survgum_gsg_aznv = @stdev(dep_survgum_gsg_aznv)
scalar STD_dep_survgum_gsg_cafl = @stdev(dep_survgum_gsg_cafl)
scalar STD_dep_survgum_gsg_canv = @stdev(dep_survgum_gsg_canv)
scalar STD_dep_survgum_gsg_flnv = @stdev(dep_survgum_gsg_flnv)
```

### Extension code

```
'Extension: survival Clayton-survival Gumbel mixture'

'to avoid errors in the parameter statement. Now eviews starts the estimation with the
     values specified in coefficient vector c'

'Clayton-survival Clayton mixture AZ-CA'

c(1) = survival_Clayton_az_ca
c(2) = survival_Gumbel_az_ca
c(3) = -1

logl logl_SClay_SGUM_az_ca
logl_SClay_SGUM_az_ca.append @logl logl81
logl_SClay_SGUM_az_ca.append f1=pdf_az
logl_SClay_SGUM_az_ca.append f2=pdf_ca
logl_SClay_SGUM_az_ca.append term1 = ( exp(c(3))/(1+exp(c(3)))*(c(1)^2)*(1/(c(1)^2)+1) *
     ((1-cdf_ca)*(1-cdf_az))^(-(c(1)^2)-1) * ( (1-cdf_ca)^(-(c(1)^2)) + (1-cdf_az)^(-(c(1)
     ^2))-1 )^(-1/(c(1)^2)-2) + (1-exp(c(3))/(1+exp(c(3)))) * ((exp(-((-log(1 - cdf_ca))
     ^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(
     exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1
      - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1
     - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c
     (2))+1)) - 1))/((cdf_ca - 1)*(cdf_az - 1)) - (exp(-((-log(1 - cdf_ca))^(1/(exp(c(2))
     /(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(
     c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c
     (2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^((exp(c(2))/(exp(c(2))
```

```
    +1)) - 2)*(-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_az))
    ^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c(2))+1))*(cdf_ca - 1)*(cdf_az -
     1))) )*f1 *f2
logl_SClay_SGUM_az_ca.append logl81 = log(term1)
smpl 1976q1 2009q1
logl_SClay_SGUM_az_ca.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar SClay_SGUM_az_ca_SCLAY = logl_SClay_SGUM_az_ca.@coef(2)
scalar SClay_SGUM_az_ca_SGUM = logl_SClay_SGUM_az_ca.@coef(3)
scalar SClay_SGUM_az_ca_PI = logl_SClay_SGUM_az_ca.@coef(1)

'Creating theta'
scalar theta_sClay_SCSG_AZCA = SClay_SGUM_az_ca_SCLAY^2
scalar theta_sGum_SCSG_AZCA = exp( SClay_SGUM_az_ca_SGUM)/(1+exp( SClay_SGUM_az_ca_SGUM )
    )
scalar pi_SCSG_AZCA = exp(SClay_SGUM_az_ca_PI)/(1+exp(SClay_SGUM_az_ca_PI))

'Calculating Kendall`s tau'
scalar TAU_SURVCLAY_SCSG_AZCA = theta_sClay_SCSG_AZCA/(2+ theta_sClay_SCSG_AZCA)
scalar TAU_SURVGUM_SCSG_AZCA = 1-theta_sGum_SCSG_AZCA


'Clayton-survival Clayton mixture AZ-FL'
c(1) = survival_Clayton_az_fl
c(2) = survival_Gumbel_az_fl
c(3) = -1

logl logl_SClay_SGUM_az_fl
logl_SClay_SGUM_az_fl.append @logl logl82
logl_SClay_SGUM_az_fl.append f1=pdf_az
logl_SClay_SGUM_az_fl.append f2=pdf_fl
logl_SClay_SGUM_az_fl.append term1 = ( exp(c(3))/(1+exp(c(3)))*(c(1)^2)*(1/(c(1)^2)+1) *
    ((1-cdf_fl)*(1-cdf_az))^(-(c(1)^2)-1) * ( (1-cdf_fl)^(-(c(1)^2)) + (1-cdf_az)^(-(c(1)
    ^2))-1 )^(-1/(c(1)^2)-2) + (1-exp(c(3))/(1+exp(c(3)))) * ((exp(-((-log(1 - cdf_fl))
    ^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(
```

```
    exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1
     - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1
    - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c
    (2))+1)) - 1))/((cdf_fl - 1)*(cdf_az - 1)) - (exp(-((-log(1 - cdf_fl))^(1/(exp(c(2))
    /(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(
    c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c
    (2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^((exp(c(2))/(exp(c(2))
    +1)) - 2)*(-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_az))
    ^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c(2))+1))*(cdf_fl - 1)*(cdf_az -
     1))) )*f1 *f2
logl_SClay_SGUM_az_fl.append logl82 = log(term1)
smpl 1976q1 2009q1
logl_SClay_SGUM_az_fl.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar SClay_SGUM_az_fl_SCLAY = logl_SClay_SGUM_az_fl.@coef(2)
scalar SClay_SGUM_az_fl_SGUM = logl_SClay_SGUM_az_fl.@coef(3)
scalar SClay_SGUM_az_fl_PI = logl_SClay_SGUM_az_fl.@coef(1)

'Creating theta'
scalar theta_sClay_SCSG_AZFL = SClay_SGUM_az_fl_SCLAY^2
scalar theta_sGum_SCSG_AZFL = exp( SClay_SGUM_az_fl_SGUM)/(1+exp( SClay_SGUM_az_fl_SGUM )
    )
scalar pi_SCSG_AZFL = exp(SClay_SGUM_az_fl_PI)/(1+exp(SClay_SGUM_az_fl_PI))

'Calculating Kendall`s tau'
scalar TAU_SURVCLAY_SCSG_AZFL = theta_sClay_SCSG_AZFL/(2+ theta_sClay_SCSG_AZFL)
scalar TAU_SURVGUM_SCSG_AZFL = 1-theta_sGum_SCSG_AZFL



'Clayton-survival Clayton mixture AZ-NV'
c(1) = survival_Clayton_az_nv
c(2) = survival_Gumbel_az_nv
c(3) = -1

logl logl_SClay_SGUM_az_nv
```

```
logl_SClay_SGUM_az_nv.append @logl logl83
logl_SClay_SGUM_az_nv.append f1=pdf_az
logl_SClay_SGUM_az_nv.append f2=pdf_nv
logl_SClay_SGUM_az_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*(c(1)^2)*(1/(c(1)^2)+1) *
    ((1-cdf_nv)*(1-cdf_az))^(-(c(1)^2)-1) * ( (1-cdf_nv)^(-(c(1)^2)) + (1-cdf_az)^(-(c(1)
    ^2))-1 )^(-1/(c(1)^2)-2) + (1-exp(c(3))/(1+exp(c(3)))) * ((exp(-((-log(1 - cdf_nv))
    ^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(
    exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1
     - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1
    - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c
    (2))+1)) - 1))/((cdf_nv - 1)*(cdf_az - 1)) - (exp(-((-log(1 - cdf_nv))^(1/(exp(c(2))
    /(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(
    c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c
    (2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^((exp(c(2))/(exp(c(2))
    +1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_az))
    ^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c(2))+1))*(cdf_nv - 1)*(cdf_az -
     1))) )*f1 *f2
logl_SClay_SGUM_az_nv.append logl83 = log(term1)
smpl 1976q1 2009q1
logl_SClay_SGUM_az_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar SClay_SGUM_az_nv_SCLAY = logl_SClay_SGUM_az_nv.@coef(2)
scalar SClay_SGUM_az_nv_SGUM = logl_SClay_SGUM_az_nv.@coef(3)
scalar SClay_SGUM_az_nv_PI = logl_SClay_SGUM_az_nv.@coef(1)

'Creating theta'
scalar theta_sClay_SCSG_AZNV = SClay_SGUM_az_nv_SCLAY^2
scalar theta_sGum_SCSG_AZNV = exp( SClay_SGUM_az_nv_SGUM)/(1+exp( SClay_SGUM_az_nv_SGUM )
    )
scalar pi_SCSG_AZNV = exp(SClay_SGUM_az_nv_PI)/(1+exp(SClay_SGUM_az_nv_PI))

'Calculating Kendall`s tau'
scalar TAU_SURVCLAY_SCSG_AZNV = theta_sClay_SCSG_AZNV/(2+ theta_sClay_SCSG_AZNV)
scalar TAU_SURVGUM_SCSG_AZNV = 1-theta_sGum_SCSG_AZNV
```

```
'Clayton-survival Clayton mixture CA-FL'
c(1) = survival_Clayton_ca_fl
c(2) = survival_Gumbel_ca_fl
c(3) = -1

logl logl_SClay_SGUM_ca_fl
logl_SClay_SGUM_ca_fl.append @logl logl84
logl_SClay_SGUM_ca_fl.append f1=pdf_ca
logl_SClay_SGUM_ca_fl.append f2=pdf_fl
logl_SClay_SGUM_ca_fl.append term1 = ( exp(c(3))/(1+exp(c(3)))*(c(1)^2)*(1/(c(1)^2)+1) *
     ((1-cdf_fl)*(1-cdf_ca))^(-(c(1)^2)-1) * ( (1-cdf_fl)^(-(c(1)^2)) + (1-cdf_ca)^(-(c(1)
     ^2))-1 )^(-1/(c(1)^2)-2) + (1-exp(c(3))/(1+exp(c(3)))) * ((exp(-((-log(1 - cdf_fl))
     ^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^(
     exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1
      - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1
     - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c
     (2))+1)) - 1))/((cdf_fl - 1)*(cdf_ca - 1)) - (exp(-((-log(1 - cdf_fl))^(1/(exp(c(2))
     /(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(
     c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c
     (2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^((exp(c(2))/(exp(c(2))
     +1)) - 2)*(-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_ca))
     ^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c(2))+1))*(cdf_fl - 1)*(cdf_ca -
      1))) )*f1 *f2
logl_SClay_SGUM_ca_fl.append logl84 = log(term1)
smpl 1976q1 2009q1
logl_SClay_SGUM_ca_fl.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
     in the code.'
scalar SClay_SGUM_ca_fl_SCLAY = logl_SClay_SGUM_ca_fl.@coef(2)
scalar SClay_SGUM_ca_fl_SGUM = logl_SClay_SGUM_ca_fl.@coef(3)
scalar SClay_SGUM_ca_fl_PI = logl_SClay_SGUM_ca_fl.@coef(1)

'Creating theta'
scalar theta_sClay_SCSG_CAFL = SClay_SGUM_ca_fl_SCLAY^2
```

```
scalar theta_sGum_SCSG_CAFL = exp( SClay_SGUM_ca_fl_SGUM)/(1+exp( SClay_SGUM_ca_fl_SGUM )
     )
scalar pi_SCSG_CAFL = exp(SClay_SGUM_ca_fl_PI)/(1+exp(SClay_SGUM_ca_fl_PI))

'Calculating Kendall‘s tau'
scalar TAU_SURVCLAY_SCSG_CAFL = theta_sClay_SCSG_CAFL/(2+ theta_sClay_SCSG_CAFL)
scalar TAU_SURVGUM_SCSG_CAFL = 1-theta_sGum_SCSG_CAFL


'Clayton-survival Clayton mixture CA-NV'
c(1) = survival_Clayton_ca_nv
c(2) = survival_Gumbel_ca_nv
c(3) = -1

logl logl_SClay_SGUM_ca_nv
logl_SClay_SGUM_ca_nv.append @logl logl85
logl_SClay_SGUM_ca_nv.append f1=pdf_ca
logl_SClay_SGUM_ca_nv.append f2=pdf_nv
logl_SClay_SGUM_ca_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*(c(1)^2)*(1/(c(1)^2)+1) *
     ((1-cdf_nv)*(1-cdf_ca))^(-(c(1)^2)-1) * ( (1-cdf_nv)^(-(c(1)^2)) + (1-cdf_ca)^(-(c(1)
     ^2))-1 )^(-1/(c(1)^2)-2) + (1-exp(c(3))/(1+exp(c(3)))) * ((exp(-((-log(1 - cdf_nv))
     ^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^(
     exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1
      - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1
     - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c
     (2))+1)) - 1))/((cdf_nv - 1)*(cdf_ca - 1)) - (exp(-((-log(1 - cdf_nv))^(1/(exp(c(2))
     /(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(
     c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c
     (2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^((exp(c(2))/(exp(c(2))
     +1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_ca))
     ^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c(2))+1))*(cdf_nv - 1)*(cdf_ca -
      1))) )*f1 *f2
logl_SClay_SGUM_ca_nv.append logl85 = log(term1)
smpl 1976q1 2009q1
logl_SClay_SGUM_ca_nv.ml(showstart)
```

```
'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar SClay_SGUM_ca_nv_SCLAY = logl_SClay_SGUM_ca_nv.@coef(2)
scalar SClay_SGUM_ca_nv_SGUM = logl_SClay_SGUM_ca_nv.@coef(3)
scalar SClay_SGUM_ca_nv_PI = logl_SClay_SGUM_ca_nv.@coef(1)

'Creating theta'
scalar theta_sClay_SCSG_CANV = SClay_SGUM_ca_nv_SCLAY^2
scalar theta_sGum_SCSG_CANV = exp( SClay_SGUM_ca_nv_SGUM)/(1+exp( SClay_SGUM_ca_nv_SGUM )
    )
scalar pi_SCSG_CANV = exp(SClay_SGUM_ca_nv_PI)/(1+exp(SClay_SGUM_ca_nv_PI))

'Calculating Kendall`s tau'
scalar TAU_SURVCLAY_SCSG_CANV = theta_sClay_SCSG_CANV/(2+ theta_sClay_SCSG_CANV)
scalar TAU_SURVGUM_SCSG_CANV = 1-theta_sGum_SCSG_CANV



'Clayton-survival Clayton mixture FL-NV'
c(1) = survival_Clayton_fl_nv
c(2) = survival_Gumbel_fl_nv
c(3) = -1

logl logl_SClay_SGUM_fl_nv
logl_SClay_SGUM_fl_nv.append @logl logl86
logl_SClay_SGUM_fl_nv.append f1=pdf_fl
logl_SClay_SGUM_fl_nv.append f2=pdf_nv
logl_SClay_SGUM_fl_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*(c(1)^2)*(1/(c(1)^2)+1) *
    ((1-cdf_nv)*(1-cdf_fl))^(-(c(1)^2)-1) * ( (1-cdf_nv)^(-(c(1)^2)) + (1-cdf_fl)^(-(c(1)
    ^2))-1 )^(-1/(c(1)^2)-2) + (1-exp(c(3))/(1+exp(c(3)))) * ((exp(-((-log(1 - cdf_nv))
    ^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))))^(
    exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1
     - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1
    - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c
    (2))+1)) - 1))/((cdf_nv - 1)*(cdf_fl - 1)) - (exp(-((-log(1 - cdf_nv))^(1/(exp(c(2))
    /(exp(c(2))+1))) + (-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(
    c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c
    (2))+1))) + (-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))))^((exp(c(2))/(exp(c(2))
```

```
    +1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_fl))
    ^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c(2))+1))*(cdf_nv - 1)*(cdf_fl -
     1))) )*f1 *f2
logl_SClay_SGUM_fl_nv.append logl86 = log(term1)
smpl 1976q1 2009q1
logl_SClay_SGUM_fl_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar SClay_SGUM_fl_nv_SCLAY = logl_SClay_SGUM_fl_nv.@coef(2)
scalar SClay_SGUM_fl_nv_SGUM = logl_SClay_SGUM_fl_nv.@coef(3)
scalar SClay_SGUM_fl_nv_PI = logl_SClay_SGUM_fl_nv.@coef(1)

'Creating theta'
scalar theta_sClay_SCSG_FLNV = SClay_SGUM_fl_nv_SCLAY^2
scalar theta_sGum_SCSG_FLNV = exp( SClay_SGUM_fl_nv_SGUM)/(1+exp( SClay_SGUM_fl_nv_SGUM )
    )
scalar pi_SCSG_FLNV = exp(SClay_SGUM_fl_nv_PI)/(1+exp(SClay_SGUM_fl_nv_PI))

'Calculating Kendall`s tau'
scalar TAU_SURVCLAY_SCSG_FLNV = theta_sClay_SCSG_FLNV/(2+ theta_sClay_SCSG_FLNV)
scalar TAU_SURVGUM_SCSG_FLNV = 1-theta_sGum_SCSG_FLNV

'-------------------------------------------------------------------------------------'
'Tail dependence'
scalar dep_survclay_scsg_azca = 2^(-1/THETA_SCLAY_scsg_AZCA)
scalar dep_survclay_scsg_azfl = 2^(-1/THETA_SCLAY_scsg_AZFL)
scalar dep_survclay_scsg_aznv = 2^(-1/THETA_SCLAY_scsg_AZNV)
scalar dep_survclay_scsg_cafl = 2^(-1/THETA_SCLAY_scsg_CAFL)
scalar dep_survclay_scsg_canv = 2^(-1/THETA_SCLAY_scsg_CANV)
scalar dep_survclay_scsg_flnv = 2^(-1/THETA_SCLAY_scsg_FLNV)

scalar dep_survgum_scsg_azca = 2-2^(THETA_SGUM_scsg_AZCA)
scalar dep_survgum_scsg_azfl = 2-2^(THETA_SGUM_scsg_AZFL)
scalar dep_survgum_scsg_aznv = 2-2^(THETA_SGUM_scsg_AZNV)
scalar dep_survgum_scsg_cafl = 2-2^(THETA_SGUM_scsg_CAFL)
scalar dep_survgum_scsg_canv = 2-2^(THETA_SGUM_scsg_CANV)
```

```
scalar dep_survgum_scsg_flnv = 2-2^(THETA_SGUM_scsg_FLNV)
'-------------------------------------------------------------------------------------------'
'BIC and Vuong'

'number of observations used in the maximum likelihood estimation'
scalar numofobs = @obssmpl

scalar BIC_scsg_AZCA = logl_SClay_SGUM_az_ca.@sc * numofobs
scalar BIC_scsg_AZFL = logl_SClay_SGUM_az_fl.@sc * numofobs
scalar BIC_scsg_AZNV = logl_SClay_SGUM_az_nv.@sc * numofobs
scalar BIC_scsg_CAFL = logl_SClay_SGUM_ca_fl.@sc * numofobs
scalar BIC_scsg_CANV = logl_SClay_SGUM_ca_nv.@sc * numofobs
scalar BIC_scsg_FLNV = logl_SClay_SGUM_fl_nv.@sc * numofobs

'Clayton-gumbel vs survial Clayton-survival Gumbel az-ca'
series m = logl81 - logl31
scalar z_cg_scsg_azca = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
scalar Vuong_cg_scsg_azca = 2*@cnorm(-@abs(z_cg_scsg_azca))

'Clayton-gumbel vs survial Clayton-survival Gumbel az-fl'
series m = logl82 - logl32
scalar z_cg_scsg_azfl = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
scalar Vuong_cg_scsg_azfl = 2*@cnorm(-@abs(z_cg_scsg_azfl))

'Clayton-gumbel vs survial Clayton-survival Gumbel az-nv'
series m = logl83 - logl33
scalar z_cg_scsg_aznv = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
scalar Vuong_cg_scsg_aznv = 2*@cnorm(-@abs(z_cg_scsg_aznv))

'Clayton-gumbel vs survial Clayton-survival Gumbel ca-fl'
series m = logl84 - logl34
scalar z_cg_scsg_cafl = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
scalar Vuong_cg_scsg_cafl = 2*@cnorm(-@abs(z_cg_scsg_cafl))

'Clayton-gumbel vs survial Clayton-survival Gumbel ca-nv'
series m = logl85 - logl35
scalar z_cg_scsg_canv = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
```

```
scalar Vuong_cg_scsg_canv = 2*@cnorm(-@abs(z_cg_scsg_canv))

'Clayton-gumbel vs survial Clayton-survival Gumbel fl-nv'
series m = logl86 - logl36
scalar z_cg_scsg_flnv = @sum(m)/@sqrt( @sumsq(m-@mean(m)) )
scalar Vuong_cg_scsg_flnv = 2*@cnorm(-@abs(z_cg_scsg_flnv))

'---------------------------------------------------------------------------------------'

'Block bootstrap survival Clayton-survival Gumbel'

'Set the sample the same as in Zimmer (2012)'
smpl 1975Q2 2009Q1

'create a group such that we can resample the input all at the same time'
group input az_percent ca_percent fl_percent nv_percent

'set seed for random number generator'
rndseed 10006

scalar rep=500 'number of replications'

'Initialise the vectors'
'theta'
vector(rep) theta_sclay_scsg_AZCA
vector(rep) theta_sgum_scsg_AZCA
vector(rep) pi_scsg_AZCA

vector(rep) theta_sclay_scsg_AZFL
vector(rep) theta_sgum_scsg_AZFL
vector(rep) pi_scsg_AZFL

vector(rep) theta_sclay_scsg_AZNV
vector(rep) theta_sgum_scsg_AZNV
vector(rep) pi_scsg_AZNV

vector(rep) theta_sclay_scsg_CAFL
```

```
vector(rep) theta_sgum_scsg_CAFL
vector(rep) pi_scsg_CAFL

vector(rep) theta_sclay_scsg_CANV
vector(rep) theta_sgum_scsg_CANV
vector(rep) pi_scsg_CANV

vector(rep) theta_sclay_scsg_FLNV
vector(rep) theta_sgum_scsg_FLNV
vector(rep) pi_scsg_FLNV
'-----------------------------------------------------------------------------------'
'Kendall`s tau'
vector(rep) TAU_SCLAY_SCSG_AZCA
vector(rep) TAU_SCLAY_SCSG_AZFL
vector(rep) TAU_SCLAY_SCSG_AZNV
vector(rep) TAU_SCLAY_SCSG_CAFL
vector(rep) TAU_SCLAY_SCSG_CANV
vector(rep) TAU_SCLAY_SCSG_FLNV
vector(rep) TAU_SGUM_SCSG_AZCA
vector(rep) TAU_SGUM_SCSG_AZFL
vector(rep) TAU_SGUM_SCSG_AZNV
vector(rep) TAU_SGUM_SCSG_CAFL
vector(rep) TAU_SGUM_SCSG_CANV
vector(rep) TAU_SGUM_SCSG_FLNV

'Tail dependence'
vector(rep) dep_SCLAY_SCSG_azca
vector(rep) dep_SCLAY_SCSG_azfl
vector(rep) dep_SCLAY_SCSG_aznv
vector(rep) dep_SCLAY_SCSG_cafl
vector(rep) dep_SCLAY_SCSG_canv
vector(rep) dep_SCLAY_SCSG_flnv
vector(rep) dep_SGUM_SCSG_azca
vector(rep) dep_SGUM_SCSG_azfl
vector(rep) dep_SGUM_SCSG_aznv
vector(rep) dep_SGUM_SCSG_cafl
vector(rep) dep_SGUM_SCSG_canv
```

```
vector(rep) dep_SGUM_SCSG_flnv

'block bootstrap approach'
For !i = 1 to rep
'make sure the sample is back to the original sample size'
smpl 1975Q2 2009Q1

input.resample(block=20) az_percent_B ca_percent_B fl_percent_B nv_percent_B

'AR(1)-GARCH(1,1) model and the corresponding conditional variance of the error term are
    calculated. Then the eventual corrected filtered price changes are determined'

'Arizona'
equation eqAZ.arch az_percent_B az_percent_B(-1) AR(1)
series az_resid = resid
eqAZ.makegarch AZcvar
series az_filter = (az_resid) / @sqrt(AZcvar)

'California'
equation eqCA.arch ca_percent_B ca_percent_B(-1) AR(1)
series ca_resid = resid
eqAZ.makegarch CAcvar
series ca_filter = (ca_resid) / @sqrt(CAcvar)

'Florida'
equation eqFL.arch fl_percent_B fl_percent_B(-1) AR(1)
series fl_resid = resid
eqFL.makegarch FLcvar
series fl_filter = (fl_resid) / @sqrt(FLcvar)

'Nevada'
equation eqNV.arch nv_percent_B nv_percent_B(-1) AR(1)
series nv_resid = resid
eqNV.makegarch NVcvar
series nv_filter = (nv_resid) / @sqrt(NVcvar)
```

```
'CDFs of the marginals'
series cdf_az = @cnorm(az_filter)
series cdf_ca = @cnorm(ca_filter)
series cdf_fl = @cnorm(fl_filter)
series cdf_nv = @cnorm(nv_filter)

'PDFs of the marginals'
series pdf_az = @dnorm(az_filter)
series pdf_ca = @dnorm(ca_filter)
series pdf_fl = @dnorm(fl_filter)
series pdf_nv = @dnorm(nv_filter)

'-----------------------------------------------------------------------------------'

'Clayton-survival Clayton mixture AZ-CA'

'to avoid errors in the parameter statement. Now eviews starts the estimation with the
    values originally present in coefficient vector c'
c(1) = survival_Clayton_az_ca
c(2) = survival_Gumbel_az_ca
c(3) = -1

logl logl_SClay_SGUM_az_ca
logl_SClay_SGUM_az_ca.append @logl logl81
logl_SClay_SGUM_az_ca.append f1=pdf_az
logl_SClay_SGUM_az_ca.append f2=pdf_ca
logl_SClay_SGUM_az_ca.append term1 = ( exp(c(3))/(1+exp(c(3)))*(c(1)^2)*(1/(c(1)^2)+1) *
    ((1-cdf_ca)*(1-cdf_az))^(-(c(1)^2)-1) * ( (1-cdf_ca)^(-(c(1)^2)) + (1-cdf_az)^(-(c(1)
    ^2))-1 )^(-1/(c(1)^2)-2) + (1-exp(c(3))/(1+exp(c(3)))) * ((exp(-((-log(1 - cdf_ca))
    ^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(
    exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1
     - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1
    - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c
    (2))+1)) - 1))/((cdf_ca - 1)*(cdf_az - 1)) - (exp(-((-log(1 - cdf_ca))^(1/(exp(c(2))
    /(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(
    c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c
    (2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^((exp(c(2))/(exp(c(2))
```

```
    +1)) - 2)*(-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_az))
    ^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c(2))+1))*(cdf_ca - 1)*(cdf_az -
     1))) )*f1 *f2
logl_SClay_SGUM_az_ca.append logl81 = log(term1)
smpl 1976q1 2009q1
logl_SClay_SGUM_az_ca.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
    in the code.'
scalar SClay_SGUM_az_ca_SCLAY = logl_SClay_SGUM_az_ca.@coef(2)
scalar SClay_SGUM_az_ca_SGUM = logl_SClay_SGUM_az_ca.@coef(3)
scalar SClay_SGUM_az_ca_PI = logl_SClay_SGUM_az_ca.@coef(1)

'Creating theta'
theta_sClay_SCSG_AZCA(!i) = SClay_SGUM_az_ca_SCLAY^2
theta_sGum_SCSG_AZCA(!i) = exp( SClay_SGUM_az_ca_SGUM)/(1+exp( SClay_SGUM_az_ca_SGUM ))
pi_SCSG_AZCA(!i) = exp(SClay_SGUM_az_ca_PI)/(1+exp(SClay_SGUM_az_ca_PI))

'Calculating Kendall`s tau'
TAU_SCLAY_SCSG_AZCA(!i) = theta_sClay_SCSG_AZCA(!i)/(2+ theta_sClay_SCSG_AZCA(!i))
TAU_SGUM_SCSG_AZCA(!i) = 1-theta_sGum_SCSG_AZCA(!i)



'Clayton-survival Clayton mixture AZ-FL'
c(1) = survival_Clayton_az_fl
c(2) = survival_Gumbel_az_fl
c(3) = -1

logl logl_SClay_SGUM_az_fl
logl_SClay_SGUM_az_fl.append @logl logl82
logl_SClay_SGUM_az_fl.append f1=pdf_az
logl_SClay_SGUM_az_fl.append f2=pdf_fl
logl_SClay_SGUM_az_fl.append term1 = ( exp(c(3))/(1+exp(c(3)))*(c(1)^2)*(1/(c(1)^2)+1) *
    ((1-cdf_fl)*(1-cdf_az))^(-(c(1)^2)-1) * ( (1-cdf_fl)^(-(c(1)^2)) + (1-cdf_az)^(-(c(1)
    ^2))-1 )^(-1/(c(1)^2)-2) + (1-exp(c(3))/(1+exp(c(3)))) * ((exp(-((-log(1 - cdf_fl))
    ^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(
    exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1
```

```
 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1
     - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c
     (2))+1)) - 1))/((cdf_fl - 1)*(cdf_az - 1)) - (exp(-((-log(1 - cdf_fl))^(1/(exp(c(2))
     /(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(
     c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c
     (2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^((exp(c(2))/(exp(c(2))
     +1)) - 2)*(-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_az))
     ^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c(2))+1))*(cdf_fl - 1)*(cdf_az -
      1))) )*f1 *f2
logl_SClay_SGUM_az_fl.append logl82 = log(term1)
smpl 1976q1 2009q1
logl_SClay_SGUM_az_fl.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
     in the code.'
scalar SClay_SGUM_az_fl_SCLAY = logl_SClay_SGUM_az_fl.@coef(2)
scalar SClay_SGUM_az_fl_SGUM = logl_SClay_SGUM_az_fl.@coef(3)
scalar SClay_SGUM_az_fl_PI = logl_SClay_SGUM_az_fl.@coef(1)

'Creating theta'
theta_sClay_SCSG_AZFL(!i) = SClay_SGUM_az_fl_SCLAY^2
theta_sGum_SCSG_AZFL(!i) = exp( SClay_SGUM_az_fl_SGUM)/(1+exp( SClay_SGUM_az_fl_SGUM ))
pi_SCSG_AZFL(!i) = exp(SClay_SGUM_az_fl_PI)/(1+exp(SClay_SGUM_az_fl_PI))

'Calculating Kendall`s tau'
TAU_SCLAY_SCSG_AZFL(!i) = theta_sClay_SCSG_AZFL(!i)/(2+ theta_sClay_SCSG_AZFL(!i))
TAU_SGUM_SCSG_AZFL(!i) = 1-theta_sGum_SCSG_AZFL(!i)

'Clayton-survival Clayton mixture AZ-NV'
c(1) = survival_Clayton_az_nv
c(2) = survival_Gumbel_az_nv
c(3) = -1

logl logl_SClay_SGUM_az_nv
logl_SClay_SGUM_az_nv.append @logl logl83
logl_SClay_SGUM_az_nv.append f1=pdf_az
logl_SClay_SGUM_az_nv.append f2=pdf_nv
```

```
logl_SClay_SGUM_az_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*(c(1)^2)*(1/(c(1)^2)+1) *
     ((1-cdf_nv)*(1-cdf_az))^(-(c(1)^2)-1) * ( (1-cdf_nv)^(-(c(1)^2)) + (1-cdf_az)^(-(c(1)
     ^2))-1 )^(-1/(c(1)^2)-2) + (1-exp(c(3))/(1+exp(c(3)))) * ((exp(-((-log(1 - cdf_nv))
     ^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(
     exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1
      - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1
     - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c
     (2))+1)) - 1))/((cdf_nv - 1)*(cdf_az - 1)) - (exp(-((-log(1 - cdf_nv))^(1/(exp(c(2))
     /(exp(c(2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(
     c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c
     (2))+1))) + (-log(1 - cdf_az))^(1/(exp(c(2))/(exp(c(2))+1))))^((exp(c(2))/(exp(c(2))
     +1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_az))
     ^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c(2))+1))*(cdf_nv - 1)*(cdf_az -
      1))) )*f1 *f2
logl_SClay_SGUM_az_nv.append logl83 = log(term1)
smpl 1976q1 2009q1
logl_SClay_SGUM_az_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
     in the code.'
scalar SClay_SGUM_az_nv_SCLAY = logl_SClay_SGUM_az_nv.@coef(2)
scalar SClay_SGUM_az_nv_SGUM = logl_SClay_SGUM_az_nv.@coef(3)
scalar SClay_SGUM_az_nv_PI = logl_SClay_SGUM_az_nv.@coef(1)

'Creating theta'
theta_sClay_SCSG_AZNV(!i) = SClay_SGUM_az_nv_SCLAY^2
theta_sGum_SCSG_AZNV(!i) = exp( SClay_SGUM_az_nv_SGUM)/(1+exp( SClay_SGUM_az_nv_SGUM ))
pi_SCSG_AZNV(!i) = exp(SClay_SGUM_az_nv_PI)/(1+exp(SClay_SGUM_az_nv_PI))

'Calculating Kendall`s tau'
TAU_SCLAY_SCSG_AZNV(!i) = theta_sClay_SCSG_AZNV(!i)/(2+ theta_sClay_SCSG_AZNV(!i))
TAU_SGUM_SCSG_AZNV(!i) = 1-theta_sGum_SCSG_AZNV(!i)

'Clayton-survival Clayton mixture CA-FL'
c(1) = survival_Clayton_ca_fl
c(2) = survival_Gumbel_ca_fl
c(3) = -1
```

```
logl logl_SClay_SGUM_ca_fl
logl_SClay_SGUM_ca_fl.append @logl logl84
logl_SClay_SGUM_ca_fl.append f1=pdf_ca
logl_SClay_SGUM_ca_fl.append f2=pdf_fl
logl_SClay_SGUM_ca_fl.append term1 = ( exp(c(3))/(1+exp(c(3)))*(c(1)^2)*(1/(c(1)^2)+1) *
     ((1-cdf_fl)*(1-cdf_ca))^(-(c(1)^2)-1) * ( (1-cdf_fl)^(-(c(1)^2)) + (1-cdf_ca)^(-(c(1)
     ^2))-1 )^(-1/(c(1)^2)-2) + (1-exp(c(3))/(1+exp(c(3)))) * ((exp(-((-log(1 - cdf_fl))
     ^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^(
     exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1
      - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1
     - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c
     (2))+1)) - 1))/((cdf_fl - 1)*(cdf_ca - 1)) - (exp(-((-log(1 - cdf_fl))^(1/(exp(c(2))
     /(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(
     c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c
     (2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^((exp(c(2))/(exp(c(2))
     +1)) - 2)*(-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_ca))
     ^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c(2))+1))*(cdf_fl - 1)*(cdf_ca -
      1))) )*f1 *f2
logl_SClay_SGUM_ca_fl.append logl84 = log(term1)
smpl 1976q1 2009q1
logl_SClay_SGUM_ca_fl.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
     in the code.'
scalar SClay_SGUM_ca_fl_SCLAY = logl_SClay_SGUM_ca_fl.@coef(2)
scalar SClay_SGUM_ca_fl_SGUM = logl_SClay_SGUM_ca_fl.@coef(3)
scalar SClay_SGUM_ca_fl_PI = logl_SClay_SGUM_ca_fl.@coef(1)

'Creating theta'
theta_sClay_SCSG_CAFL(!i) = SClay_SGUM_ca_fl_SCLAY^2
theta_sGum_SCSG_CAFL(!i) = exp( SClay_SGUM_ca_fl_SGUM)/(1+exp( SClay_SGUM_ca_fl_SGUM ))
pi_SCSG_CAFL(!i) = exp(SClay_SGUM_ca_fl_PI)/(1+exp(SClay_SGUM_ca_fl_PI))

'Calculating Kendall`s tau'
TAU_SCLAY_SCSG_CAFL(!i) = theta_sClay_SCSG_CAFL(!i)/(2+ theta_sClay_SCSG_CAFL(!i))
TAU_SGUM_SCSG_CAFL(!i) = 1-theta_sGum_SCSG_CAFL(!i)
```

```
'Clayton-survival Clayton mixture CA-NV'
c(1) = survival_Clayton_ca_nv
c(2) = survival_Gumbel_ca_nv
c(3) = -1

logl logl_SClay_SGUM_ca_nv
logl_SClay_SGUM_ca_nv.append @logl logl85
logl_SClay_SGUM_ca_nv.append f1=pdf_ca
logl_SClay_SGUM_ca_nv.append f2=pdf_nv
logl_SClay_SGUM_ca_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*(c(1)^2)*(1/(c(1)^2)+1) *
     ((1-cdf_nv)*(1-cdf_ca))^(-(c(1)^2)-1) * ( (1-cdf_nv)^(-(c(1)^2)) + (1-cdf_ca)^(-(c(1)
     ^2))-1 )^(-1/(c(1)^2)-2) + (1-exp(c(3))/(1+exp(c(3)))) * ((exp(-((-log(1 - cdf_nv))
     ^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^(
     exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1
      - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1
     - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c
     (2))+1)) - 1))/((cdf_nv - 1)*(cdf_ca - 1)) - (exp(-((-log(1 - cdf_nv))^(1/(exp(c(2))
     /(exp(c(2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(
     c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c
     (2))+1))) + (-log(1 - cdf_ca))^(1/(exp(c(2))/(exp(c(2))+1))))^((exp(c(2))/(exp(c(2))
     +1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_ca))
     ^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c(2))+1))*(cdf_nv - 1)*(cdf_ca -
      1))) )*f1 *f2
logl_SClay_SGUM_ca_nv.append logl85 = log(term1)
smpl 1976q1 2009q1
logl_SClay_SGUM_ca_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
     in the code.'
scalar SClay_SGUM_ca_nv_SCLAY = logl_SClay_SGUM_ca_nv.@coef(2)
scalar SClay_SGUM_ca_nv_SGUM = logl_SClay_SGUM_ca_nv.@coef(3)
scalar SClay_SGUM_ca_nv_PI = logl_SClay_SGUM_ca_nv.@coef(1)

'Creating theta'
theta_sClay_SCSG_CANV(!i) = SClay_SGUM_ca_nv_SCLAY^2
theta_sGum_SCSG_CANV(!i) = exp( SClay_SGUM_ca_nv_SGUM)/(1+exp( SClay_SGUM_ca_nv_SGUM ))
```

```
pi_SCSG_CANV(!i) = exp(SClay_SGUM_ca_nv_PI)/(1+exp(SClay_SGUM_ca_nv_PI))

'Calculating Kendall‘s tau'
TAU_SCLAY_SCSG_CANV(!i) = theta_sClay_SCSG_CANV(!i)/(2+ theta_sClay_SCSG_CANV(!i))
TAU_SGUM_SCSG_CANV(!i) = 1-theta_sGum_SCSG_CANV(!i)

'Clayton-survival Clayton mixture FL-NV'
c(1) = survival_Clayton_fl_nv
c(2) = survival_Gumbel_fl_nv
c(3) = -1

logl logl_SClay_SGUM_fl_nv
logl_SClay_SGUM_fl_nv.append @logl logl86
logl_SClay_SGUM_fl_nv.append f1=pdf_fl
logl_SClay_SGUM_fl_nv.append f2=pdf_nv
logl_SClay_SGUM_fl_nv.append term1 = ( exp(c(3))/(1+exp(c(3)))*(c(1)^2)*(1/(c(1)^2)+1) *
     ((1-cdf_nv)*(1-cdf_fl))^(-(c(1)^2)-1) * ( (1-cdf_nv)^(-(c(1)^2)) + (1-cdf_fl)^(-(c(1)
     ^2))-1 )^(-1/(c(1)^2)-2) + (1-exp(c(3))/(1+exp(c(3)))) * ((exp(-((-log(1 - cdf_nv))
     ^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))))^(
     exp(c(2))/(exp(c(2))+1)))*((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1))) + (-log(1
      - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))))^(2*(exp(c(2))/(exp(c(2))+1)) - 2)*(-log(1
     - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c
     (2))+1)) - 1))/((cdf_nv - 1)*(cdf_fl - 1)) - (exp(-((-log(1 - cdf_nv))^(1/(exp(c(2))
     /(exp(c(2))+1))) + (-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))))^(exp(c(2))/(exp(
     c(2))+1)))*((exp(c(2))/(exp(c(2))+1)) - 1)*((-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c
     (2))+1))) + (-log(1 - cdf_fl))^(1/(exp(c(2))/(exp(c(2))+1))))^((exp(c(2))/(exp(c(2))
     +1)) - 2)*(-log(1 - cdf_nv))^(1/(exp(c(2))/(exp(c(2))+1)) - 1)*(-log(1 - cdf_fl))
     ^(1/(exp(c(2))/(exp(c(2))+1)) - 1))/((exp(c(2))/(exp(c(2))+1))*(cdf_nv - 1)*(cdf_fl -
      1))) )*f1 *f2
logl_SClay_SGUM_fl_nv.append logl86 = log(term1)
smpl 1976q1 2009q1
logl_SClay_SGUM_fl_nv.ml(showstart)

'Note that the output first gives the coefficient for pi, even though it is called c(3)
     in the code.'
scalar SClay_SGUM_fl_nv_SCLAY = logl_SClay_SGUM_fl_nv.@coef(2)
scalar SClay_SGUM_fl_nv_SGUM = logl_SClay_SGUM_fl_nv.@coef(3)
```

```
scalar SClay_SGUM_fl_nv_PI = logl_SClay_SGUM_fl_nv.@coef(1)

'Creating theta'
theta_sClay_SCSG_FLNV(!i) = SClay_SGUM_fl_nv_SCLAY^2
theta_sGum_SCSG_FLNV(!i) = exp( SClay_SGUM_fl_nv_SGUM)/(1+exp( SClay_SGUM_fl_nv_SGUM ))
pi_SCSG_FLNV(!i) = exp(SClay_SGUM_fl_nv_PI)/(1+exp(SClay_SGUM_fl_nv_PI))

'Calculating Kendall`s tau'
TAU_SCLAY_SCSG_FLNV(!i) = theta_sClay_SCSG_FLNV(!i)/(2+ theta_sClay_SCSG_FLNV(!i))
TAU_SGUM_SCSG_FLNV(!i) = 1-theta_sGum_SCSG_FLNV(!i)

'--------------------------------------------------------------------------------------'
'Tail dependence'
dep_sclay_scsg_azca(!i) = 2^(-1/THETA_SCLAY_scsg_AZCA(!i))
dep_sclay_scsg_azfl(!i) = 2^(-1/THETA_SCLAY_scsg_AZFL(!i))
dep_sclay_scsg_aznv(!i) = 2^(-1/THETA_SCLAY_scsg_AZNV(!i))
dep_sclay_scsg_cafl(!i) = 2^(-1/THETA_SCLAY_scsg_CAFL(!i))
dep_sclay_scsg_canv(!i) = 2^(-1/THETA_SCLAY_scsg_CANV(!i))
dep_sclay_scsg_flnv(!i) = 2^(-1/THETA_SCLAY_scsg_FLNV(!i))

dep_sgum_scsg_azca(!i) = 2-2^(THETA_SGUM_scsg_AZCA(!i))
dep_sgum_scsg_azfl(!i) = 2-2^(THETA_SGUM_scsg_AZFL(!i))
dep_sgum_scsg_aznv(!i) = 2-2^(THETA_SGUM_scsg_AZNV(!i))
dep_sgum_scsg_cafl(!i) = 2-2^(THETA_SGUM_scsg_CAFL(!i))
dep_sgum_scsg_canv(!i) = 2-2^(THETA_SGUM_scsg_CANV(!i))
dep_sgum_scsg_flnv(!i) = 2-2^(THETA_SGUM_scsg_FLNV(!i))

Next

'Calculating the standard deviations of the variables'
scalar STD_tau_sclay_scsg_azca = @stdev(tau_sclay_scsg_azca)
scalar STD_tau_sclay_scsg_azfl = @stdev(tau_sclay_scsg_azfl)
scalar STD_tau_sclay_scsg_aznv = @stdev(tau_sclay_scsg_aznv)
scalar STD_tau_sclay_scsg_cafl = @stdev(tau_sclay_scsg_cafl)
scalar STD_tau_sclay_scsg_canv = @stdev(tau_sclay_scsg_canv)
scalar STD_tau_sclay_scsg_flnv = @stdev(tau_sclay_scsg_flnv)
```

```
scalar STD_tau_sgum_scsg_azca = @stdev(tau_sgum_scsg_azca)
scalar STD_tau_sgum_scsg_azfl = @stdev(tau_sgum_scsg_azfl)
scalar STD_tau_sgum_scsg_aznv = @stdev(tau_sgum_scsg_aznv)
scalar STD_tau_sgum_scsg_cafl = @stdev(tau_sgum_scsg_cafl)
scalar STD_tau_sgum_scsg_canv = @stdev(tau_sgum_scsg_canv)
scalar STD_tau_sgum_scsg_flnv = @stdev(tau_sgum_scsg_flnv)

scalar STD_pi_scsg_azca = @stdev(pi_scsg_azca)
scalar STD_pi_scsg_azfl = @stdev(pi_scsg_azfl)
scalar STD_pi_scsg_aznv = @stdev(pi_scsg_aznv)
scalar STD_pi_scsg_cafl = @stdev(pi_scsg_cafl)
scalar STD_pi_scsg_canv = @stdev(pi_scsg_canv)
scalar STD_pi_scsg_flnv = @stdev(pi_scsg_flnv)

scalar STD_dep_sclay_scsg_azca = @stdev(dep_sclay_scsg_azca)
scalar STD_dep_sclay_scsg_azfl = @stdev(dep_sclay_scsg_azfl)
scalar STD_dep_sclay_scsg_aznv = @stdev(dep_sclay_scsg_aznv)
scalar STD_dep_sclay_scsg_cafl = @stdev(dep_sclay_scsg_cafl)
scalar STD_dep_sclay_scsg_canv = @stdev(dep_sclay_scsg_canv)
scalar STD_dep_sclay_scsg_flnv = @stdev(dep_sclay_scsg_flnv)
scalar STD_dep_sgum_scsg_azca = @stdev(dep_sgum_scsg_azca)
scalar STD_dep_sgum_scsg_azfl = @stdev(dep_sgum_scsg_azfl)
scalar STD_dep_sgum_scsg_aznv = @stdev(dep_sgum_scsg_aznv)
scalar STD_dep_sgum_scsg_cafl = @stdev(dep_sgum_scsg_cafl)
scalar STD_dep_sgum_scsg_canv = @stdev(dep_sgum_scsg_canv)
scalar STD_dep_sgum_scsg_flnv = @stdev(dep_sgum_scsg_flnv)
```