

Estimating with the Gaussian, Clayton, and t Copulas  
An Exploration of Bond Defaults and Value of a Collateralized Debt  
Obligation

Charlotte Riggs

ACMS 60890

Advisor: Dr. James Delaney

## ABSTRACT

The use of copula analysis to estimate the lifetime of bonds is gaining popularity in the financial sector. Taking this concept a step further, the following research project explores the use of the copula method in estimating the joint occurrence of bond default as a method of predictive analysis of bond lifetime. This is accomplished by means of an in-depth discussion on copulas followed by a series of illustrative experiments and examples. In particular, this text covers not only the definition of three different copulas, but also an explanation of how these copulas can be used to estimate the lifetimes of bonds, predict the probability of default rate, and simulate a the value of collateralized debt obligation (CDO). The three copulas explored within this text the Gaussian copula, the Clayton copula, and the t-copula. Illustrative examples included 100,000 portfolios that were randomly generated each containing 10 bonds. After the lifetimes were calculated, a toy model was implemented to simulate a CDO with two tranches: a Senior tranche and a Equity tranche. The paper concludes with simulations that utilize probability values gleaned from historical data as provided by Joshua Coval, Jakub Jurek, and Erik Stafford in their work entitled “The Economics of Structured Finance.”

## PAPER OUTLINE

### I. INTRODUCTION (PG. 5)

- What is a Copula?
- The Gaussian Copula
- The Clayton Copula
- The t-Copula

### II. ESTIMATING WITH COPULAS (PG. 9)

- Estimating Bond Default
- Estimating the Value of CDO

### III. MODEL EXPLANATION (PG. 11)

- Model Description

### IV. EXPERIMENTS AND DISCUSSION OF RESULTS (PG. 12)

- Experimental Abstract
- Choosing Parameter Values for Copulas
- Investigation (1): Comparing Copulas
- Investigation (2): Using One Quantile Function
- Investigation (3): Using Two Different Quantile Functions

### V. SIMULATIONS UTILIZING HISTORICAL DATA INFORMATION (PG. 27)

- Overview of “The Economics of Structured Finance”
- Model Modifications

- Simulation Results

## VI. CONCLUSION (PG. 31)

- Model Critique
- Discussion of Different Copulas and CDO

## VII. REFERENCES (PG. 33)

## VIII. APPENDIX (PG. 35)

- Original Model
- Estimating the Parameter Values
- Modified Code for Estimating Parameter Values
- Modified Original Model

# I. INTRODUCTION

## WHAT IS A COPULA?

A significant topic of interest is the dependence between random variables, in particular, components of a random vector of financial risk factors. One method of observing this dependent behavior is by constructing a time series. However, as noted by Brendan Beare, copulas can also accomplish this, providing a model of this dependence over time in a stationary Markov chain. This is actually the preferred method because copulas better cordon off the dependence structure as portrayed by a joint distribution function for a random vector of risk factors (Beare 2007).

According to McNeil, Frey, and Embrechts (MFE), copulas achieve this by utilizing a bottom-up approach to multivariate model building. Which is particularly useful since the marginal behavior is easier to estimate than their dependence structure; copulas combine marginal models with probable dependence models (McNeil et al. 2005): “A d-dimensional copula is a distribution function on  $[0, 1]^d$  with standard uniform marginal distributions” (185). Let  $\mathcal{C}(\mathbf{u}) = \mathcal{C}(u_1, \dots, u_d)$  represent the multivariate distribution functions that are copulas. Meaning that C is “a mapping of the unity hypercube into the unit interval” (185) where the following properties must be true:

(1.)  $\mathcal{C}(\mathbf{u})$  is increasing with each component  $u_i$ .

(2.)  $\mathcal{C}(1, \dots, 1, u_i, 1, \dots, 1) = u_i$  for all  $i \in [1, \dots, d], u_i \in [0, 1]$ .

(3.) When all  $(a_1, \dots, a_d), (b_1, \dots, b_d) \in [0, 1]^d$  where

$$a_i \leq b_i, \sum_{i_1=1}^2 \dots \sum_{i_d=1}^2 (-1)^{i_1+\dots+i_d} \mathcal{C}(u_{1i_1}, \dots, u_{di_d}) \geq 0, \text{ where } u_{j1} = a_j$$

and  $u_{j2} = b_j$  for all  $j \in \{1, \dots, d\}$ .

In particular, this report focuses on and employs the “copula of  $F$ .” According to Sklar’s Theorem, all multivariate distribution functions have copulas and copulas used with univariate distribution functions can result in multivariate distribution functions (McNeil et al. 2005). More formally, Theorem 5.3 from MFE on page 186 gives Sklar’s theorem as: If  $F$  is a joint distribution function with margins  $F_1, \dots, F_d$ , then there is a copula  $C : [0, 1]^d \rightarrow [0, 1]$  such that for all  $x_1, \dots, x_d$  in  $\bar{\mathbb{R}} = [-\infty, \infty]$ ,

$$F(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d)).$$

From Sklar’s theorem, it can be concluded that when continuous margins are present, it is reasonable to describe the idea of the copula of a distribution, yielding the copula of  $F$ :

“If the random variable  $\mathbf{X}$  has joint distribution function  $F$  with continuous marginal distributions  $F_1, \dots, F_d$ , then the copula of  $F$  (or  $\mathbf{X}$ ) is the distribution function  $C$  of  $(F_1(X_1), \dots, F_d(X_d))$ ” (187).

In summary, if given a random pair of vectors  $(\mathbf{X}, \mathbf{Y})$  then the copula is equal to their joint cumulative distribution function. Or more simply,

$$H(x, y) = C(F(x), G(y)),$$

where  $C(u, v)$  is the copula and  $F$  and  $G$  are marginal distribution functions (Habiboallah 2007).

## THE GAUSSIAN COPULA

A given copula is considered a Gaussian copula, if the random vector  $\mathbf{Y} \sim \text{Nd}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  is Gaussian. As noted by MFE, the process of standardizing the margins results in applying a series of strictly increasing transformations. Which according to the following proposition,

strongly suggests that the copula of  $\mathbf{Y}$  is identical to the copula of  $\mathbf{X} \sim N_d(\mathbf{0}, P)$ , where  $P$  is the correlation matrix of  $\mathbf{Y}$  (McNeil et al. 2005).

As MFE's proposition 5.6 states, if  $(X_1, \dots, X_d)$  is a random vector with continuous margins and copula  $C$  and  $T_1, \dots, T_d$  are strictly increasing functions, then  $(T_1(X_1), \dots, T_d(X_d))$  also has copula  $C$ . Thus, the Gaussian copula can be defined as:

$$C_P^{Ga}(\mathbf{u}) = P(\Phi(X_1) \leq u_1, \dots, \Phi(X_d) \leq u_d) = \Phi_P(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_d)),$$

where  $\Phi$  is the standard univariate normal distribution and  $\Phi_P$  is the joint distribution function of  $\mathbf{X}$  and can be simulated using the following algorithm (McNeil et al. 2005):

- (1.) Generate  $\mathbf{Z} \sim N_d(\mathbf{0}, P)$ .
- (2.) Return  $\mathbf{U} = (\Phi(Z_1), \dots, \Phi(Z_d))'$ , where  $\Phi$  is the standard normal distribution function and the random vector  $\mathbf{U}$  has distribution function  $C_P^{Ga}$ .

## THE CLAYTON COPULA

The bivariate Clayton copula is one of several Archimedean copulas, where an Archimedean copula is described as follows by Beare's article entitled "Archimedean Copulas and Temporal Dependence:"

"A copula ... is said to be Archimedean if there exists a continuous, strictly decreasing, convex function  $\varphi : [0, 1] \rightarrow [0, \infty]$  with  $\varphi(1) = 0$  such that  $C(u, v) = \varphi^{[-1]}(\varphi(u) + \varphi(v))$  for all  $(u, v) \in [0, 1]^2$ . The function  $\varphi$  is referred to as the Archimedean generator of  $C$ . When  $\varphi(0) = \infty$ ,  $C$  is said to be strictly Archimedean, and  $\varphi$  is said to be a strict Archimedean generator" (3).

Or as defined by MFE, if  $C_\theta$  be any exchangeable bivariate copula, then a parametric family of asymmetric copulas  $C_{\theta,\alpha,\beta}$  is obtained by setting  $C_{\theta,\alpha,\beta}(u_1, u_2) = u_1^{1-\alpha}u_2^{1-\beta}C_\theta(u_1^\alpha, u_2^\beta)$ , where  $0 \leq u_1, u_2 \leq 1$ ,  $\alpha \geq 0$ , and  $\beta \leq 1$  (225).

Meaning that the Gaussian copula is taken from well-known multivariate distribution functions and does not have a simple closed form, and Archimedean copulas like the Clayton copula do have simple closed forms (Beare 2007). The algorithm as given by MFE for obtaining an asymmetric bivariate Archimedean copula is:

- (1.) Generate a random pair  $(V_1, V_2)$  with distribution function  $C_\theta$ .
- (2.) Independent of  $V_1, V_2$ , generate two independent standard uniform variates  $\widetilde{U}_1$  and  $\widetilde{U}_2$ .
- (3.) Return  $U_1 = \max\{V_1^{\frac{1}{\alpha}}, \widetilde{U}_1^{1-\alpha}\}$  and  $U_2 = \max\{V_2^{\frac{1}{\beta}}, \widetilde{U}_2^{1-\beta}\}$ .

The Clayton copula, as mentioned by Durante, Jaworski, and Mesiar, is particularly useful when  $\mathbf{X}$  and  $\mathbf{Y}$ , from the random pair  $(\mathbf{X}, \mathbf{Y})$ , are smaller than their  $\alpha$ -quantile and  $\beta$ -quantile. The formula for this copula is provided below (Durante et al. 2011).

$$C_\theta^{Cl}(u_1, u_2) = (u_1^{-\theta} + u_2^{-\theta} - 1)^{-1/\theta}, \text{ where } 0 < \theta < \infty.$$

## THE T-COPULA

As mentioned by Demarta and McNeil, between the Gaussian copula and the t-copula, the t-copula has been found to be the superior of the two. Since multivariate financial returns have frequently been observed of having tail dependence, it makes sense that the t-copula better models their dependence structure. This is because the t-copula produces both positive upper and lower tail dependence while the Gaussian copula has



neither (Demarta et al. 2005). However, in cases where the symmetry of the t-copula may not be preferred, the Clayton copula, which only has positive lower tail dependence, may be used.

Like the Gaussian copula, the t-copula is also implicit (McNeil et al. 2005). Consequently, the algorithm for its simulation, as given by Algorithm 5.10 in MFE, is very similar to that of the Gaussian copula:

- (1.) Generate  $\mathbf{X} \sim t_d(v, \mathbf{0}, P)$ .
- (2.) Return  $\mathbf{U} = (t_v(X_1), \dots, t_v(X_d))'$ , where  $t_v$  denotes the distribution function of a standard univariate t-distribution and the random vector  $\mathbf{U}$  has the distribution function  $C_{v,P}^t$ .

Note that the random vector  $\mathbf{X}$  has a multivariate t distribution whose density, as provided by Demarta and McNeil, is:

$$f(\mathbf{x}) = \frac{\Gamma\left(\frac{v+d}{2}\right)}{\Gamma\left(\frac{v}{2}\right)\sqrt{(\pi v)^d |\Sigma|}} \left(1 + \frac{(\mathbf{x} - \boldsymbol{\mu})' \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})}{v}\right)^{-\frac{v+d}{2}}$$

where the mean vector  $\boldsymbol{\mu}$  is equal to zero, which results in the unique copula:

$$C_{v,P}^t(\mathbf{u}) = \int_{-\infty}^{t_v^{-1}(u_1)} \dots \int_{-\infty}^{t_v^{-1}(u_d)} \frac{\Gamma\left(\frac{v+d}{2}\right)}{\Gamma\left(\frac{v}{2}\right)\sqrt{(\pi v)^d |P|}} \left(1 + \frac{\mathbf{x}' P^{-1} \mathbf{x}}{v}\right)^{-\frac{v+d}{2}} d\mathbf{x}.$$

## II. ESTIMATING WITH COPULAS

### ESTIMATING THE NUMBER OF DEFAULTS

As mentioned in the introduction, copulas are used to illustrate the dependence between two random variables. Thus, this method can easily be used to model the lifetimes of bonds in a given portfolio, and consequently the number of defaults within said portfolio.

This is accomplished by randomly generating a  $d$ -dimensional copula, where  $d$  is equal to the number of bonds in that portfolio. Then, the quantile for a distinct distribution is used on this copula to estimate the lifetimes of its bonds. In order to determine whether or not a default has occurred, these lifetimes are compared to a specified time horizon. If a bond's lifetime is a shorter length than that of the time horizon, then it can be concluded that the bond has defaulted. Therefore, the total number of bonds whose lifetimes are less than that of the time horizon is equal to the total number of defaults in this portfolio.

In conclusion, this method can be employed to determine the likelihood of having a particular number of defaults transpire within a certain time period, given that there are  $d$  bonds in a portfolio. Creating a significantly large group of portfolios each with  $d$  bonds, tabulating the total number of defaults within in each portfolio, and dividing each total default sum by the number of generated portfolios obtains these probabilities. However, it is important to note that the greater the amount of portfolios involved in this process, the more accurate these probabilities will be.

## ESTIMATING THE VALUE OF CDO

The definition for collateralized debt obligation, or CDO, as supplied by Duffie and Gârleanu, is an asset-backed security whose underlying collateral tends to be a portfolio of bonds or bank loans. They then go on to define tranches as a prioritized collection of CDO securities that are given interested income and principal repayments by a CDO cash-flow structure. These returns usually come from a collateral pool of different debt instruments. There are many alternate forms of these structures of tranches, but in general the Senior CDO notes are paid before mezzanine and other lower-subordinated notes, with the Equity tranche receiving any remaining returns after all others are paid (Duffie et al. 2001).

Copula models are popular for estimating the value of a CDO. As summarized by Habiboellah, in order to determine the amount paid to each of the tranches of a CDO the default probability, the default severity/recovery, and the default correlation must be known. This is where the use of a copula comes into play. One-factor copula models can be utilized to model default correlations, which are the probabilities of one bond, or whatever type of asset is of interest, defaulting as a result of another default (Habiboellah 2007).

### III. MODEL EXPLANATION

#### DESCRIPTION

As mentioned in the previous section, copulas can be used to model the lifetimes of bonds, whose probabilities of defaulting can then be applied to estimating the value of CDO. This particular model, defined by the R function “defaultSim(),” provides a simple way to simulate a specified number of portfolios each with a given number of bonds by using the Gaussian copula, the Clayton copula, and the t-Copula (see appendix for R code). Then, it either uses one quantile function to determine the lifetimes of all bonds or two quantile functions to determine the lifetimes of each half of the bonds in every portfolio. However, this function only has four different quantile functions to choose from: the quantile function for the exponential distribution, for the chi-squared distribution, for the lognormal distribution, and for the Weibull distribution.

If a bond’s lifetime is smaller than that of the provided time horizon, the bond has defaulted. The number of defaults in each portfolio is then tabulated, and the overall probability of a certain number of defaults occurring in a distinct portfolio is calculated. A barplot based off of this tabulation is output for each set of probabilities obtained from each data set derived from the three different copulas.

This tabulation of defaults can then be used in the toy model value of a CDO. This toy model assumes that the structure of tranches within each portfolio only includes two investors: Senior investors and Equity investors. It also assumes that the Senior investors own 80% of each portfolio and are paid first. Thus, the Equity investors own the other 20% and are paid second if there are enough returns remaining. Estimates of the expected values for the Senior tranche as well as the Equity tranche are printed for each set of simulations. Barplots with the probabilities of return values for each tranche are also produced for each set of data.

## IV. EXPERIMENTS AND DISCUSSION OF RESULTS

### ABSTRACT

Each of the subsequent experiments, aside from the first, perform a set of trials that manipulate the variables that account for the different types of quantile functions available to determine the lifetimes of all bonds. The other argument values of the function “defaultSim()” for these trials remain unchanged throughout all investigations, except the parameter values will vary depending on the selected quantile function(s). In particular, “toy.model” is habitually set to “TRUE” considering the fact that the CDO toy model is always of interest with the exception of Investigation (3).

### CHOOSING PARAMETER VALUES FOR COPULAS

As mentioned previously, three different copulas are being used to generate a desired number of portfolios in each investigation. Therefore, it is important that these results can be compared to one another on equal grounds. Meaning, that the parameter value for each

copula must be selected based on the fact that the correlation matrices for the lifetimes of all bonds from the three randomly generated sets of portfolios are very similar or ideally, equal. In order to find these values so that the succeeding investigations can be performed successfully, the code found in section (II) of the appendix was created. This code focuses on the quantile function for the exponential distribution, but was manipulated to examine the other three types of quantile functions included in the original model.

Once all of the copula parameters were found for each varying quantile function, the values were recorded below along with the induced correlation. Also, note that the lifetime parameter used in every case was 0.5.

	Exponential Quantile Function	Chi-squared Quantile Function	Lognormal Quantile Function	Weibull Quantile Function
Gaussian Copula	0.206	0.2	0.22	0.22
Clayton Copula	0.509	0.78	0.78	0.999
t-copula	0.17	0.04	0.07	0.005
Approximated Induced Correlation	0.17	0.12	0.14	0.11

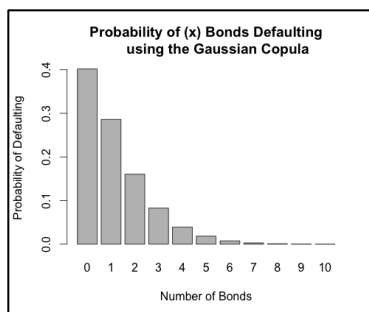
Lastly, consider the case where it is desired to use one type of quantile function to estimate the lifetimes of the first half of the bonds in a portfolio and a different quantile function to estimate the lifetimes of the second half. Here, the user needs to ensure that both sets of parameters result in approximately the same induced correlation. Meaning, that if the lifetimes of the first half are estimated using the exponential quantile function and the second by means of the chi-squared quantile function, then the correlation matrices for the first half should yield an induced correlation similar to that of the second. So for example, it would not be recommended to use two different sets of parameter values listed in the table above for this given method because each gives a slightly different induced correlation.

## INVESTIGATION (1): COMPARING COPULAS

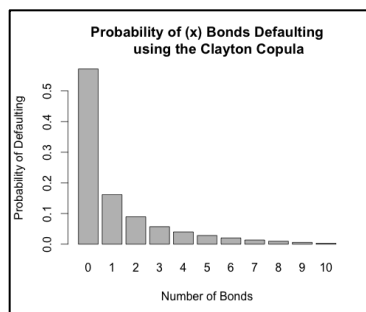
Hypothesis: Since the Gaussian copula has no upper or lower tail dependence, the likelihood that there will be a large number of defaults within a given portfolio, is less than that for the set of portfolios generated using either the Clayton copula or the t-Copula. Out of these two, it is anticipated that the portfolios generated using the t-copula will have a higher likelihood of multiple defaults occurring in a specific portfolio. This is because the t-copula has both positive upper and lower tail dependence, while the Clayton copula only has positive lower tail dependence. Thus as a result, the estimated expected value for the Senior tranche will be the highest when the data derived from the Gaussian copula generated portfolios are used, and the lowest when the data derived from the t-copula generated portfolios are used. Consequently, the estimated expected value for the Equity tranche will be the lowest when those simulated by the Gaussian copula are used and the highest when those simulated by the t-copula are used, with the expected value of those pertaining to the Clayton copula lying somewhere in between.

Implementation: `defaultSim(10, 100000, 10, "exp", 0, 0.206, 0.509, 0.17, 0, 0, 0.5, 0.25, T)`

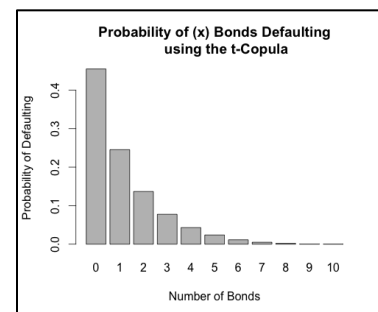
Results for Estimating the Number of Defaults:



(a.1)



(b.1)



(c.1)

FIGURES (a.1), (b.1), and (c.1) are barplots that give the probabilities of a number of defaults occurring in a portfolio generated by one of the three copulas of interest in this report. From these diagrams it can be noted that (b.1) has the highest probability value for the event of zero defaults occurring, while the (c.1) shows the lowest probability. However, the steepest “slope,” resulting in the lowest probabilities for the other totals, is given by figure (b.1), while the most gradual “slope” is observed in figure (a.1).

TABLE (1.1): Calculated probabilities of (x) number of bonds defaulting in a particular portfolio generated from the Gaussian copula:

Default #	0	1	2	3	4	5	6	7	8	9	10
Prob.	0.40172	0.28595	0.16043	0.08272	0.03925	0.01836	0.00748	0.00291	0.00096	0.00021	0.00001

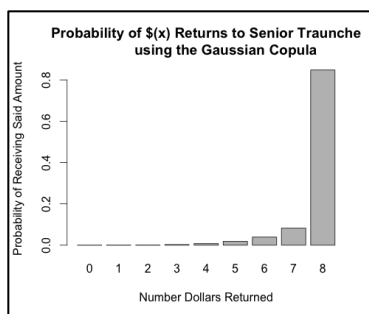
TABLE (1.2): Calculated probabilities of (x) number of bonds defaulting in a particular portfolio generated by the Clayton copula:

Default #	0	1	2	3	4	5	6	7	8	9	10
Prob.	0.57186	0.16170	0.08933	0.05714	0.03993	0.02831	0.02023	0.01358	0.00955	0.00561	0.00276

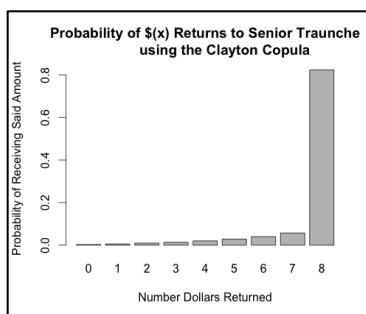
TABLE (1.3): Calculated probabilities of (x) number of bonds defaulting in a particular portfolio generated by the t-copula:

Default #	0	1	2	3	4	5	6	7	8	9	10
Prob.	0.45485	0.24528	0.13663	0.07745	0.04294	0.02378	0.01135	0.00492	0.00207	0.00060	0.00013

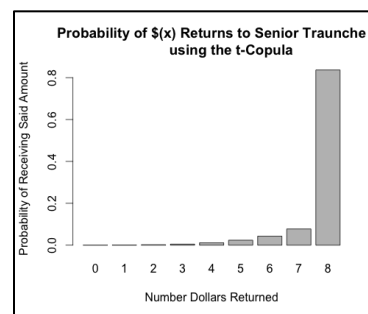
### Results for Estimating the Value of a CDO:



(d.1)



(e.1)



(f.1)

FIGURES (d.1), (e.1), and (f.1) are barplots that give the probabilities of a number of dollars returned to the Senior tranche from a given portfolio generated by a particular copula once the time horizon has expired. Just from these plots, it is extremely difficult to distinguish which data set resulted in the highest estimated expected value for the amount of returns received by the Senior tranche.

TABLE (1.4): Calculated probabilities of (x) number of dollars being returned to the Senior tranche from a particular portfolio generated by the Gaussian copula. The estimate of the expected value for the Senior tranche was \$7.73192:

Returns in (\$)	0	1	2	3	4	5	6	7	8
Prob.	0.00001	0.00021	0.00096	0.00291	0.00748	0.01836	0.03925	0.08272	0.84810

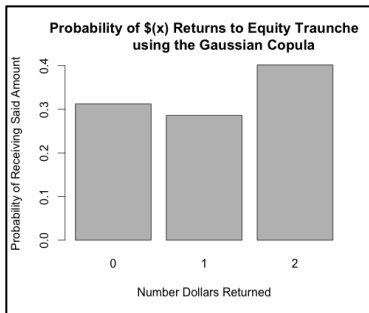
TABLE (1.5): Calculated probabilities of (x) number of dollars being returned to the Senior tranche from a particular portfolio generated by the Clayton copula. The estimate of the expected value for the Senior tranche was \$7.5106:

Returns in (\$)	0	1	2	3	4	5	6	7	8
Prob.	0.00276	0.00561	0.00955	0.01358	0.02023	0.02831	0.03993	0.05714	0.82289

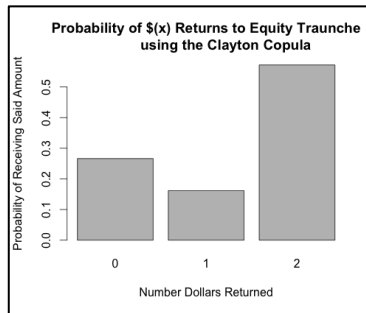


TABLE (1.6): Calculated probabilities of (x) number of dollars being returned to the Senior tranche from a particular portfolio generated by the t-copula. The estimate of the expected value for the Senior tranche was \$7.67767:

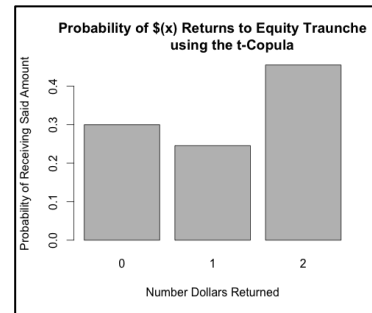
Returns in (\$)	0	1	2	3	4	5	6	7	8
Prob.	0.00013	0.00060	0.00207	0.00492	0.01135	0.02378	0.04294	0.07745	0.83676



(g.1)



(h.1)



(i.1)

FIGURES (g.1), (h.1), and (i.1) are barplots that give the probabilities of a number of dollars returned to the Equity tranche from a given portfolio generated by a particular copula once the time horizon has expired. From these plots, the probabilities for returns being equal to two dollars seem to be very similar to one another in all three cases. However, the probabilities for the returns being equal to zero dollars or one dollar do vary, (g.1) displaying the largest values and (h.1) displaying the smallest.

TABLE (1.7): Calculated probabilities of (x) number of dollars being returned to the Equity tranche from a particular portfolio generated by the Gaussian copula. The estimate of the expected value for the Equity tranche was \$1.08939:

Returns in (\$)	0	1	2
Prob.	0.31233	0.28595	0.40172

*TABLE (1.8):* Calculated probabilities of (x) number of dollars being returned to the Equity tranche from a particular portfolio generated by the Clayton copula. The estimate of the expected value for the Equity tranche was \$1.30542:

Returns in (\$)	0	1	2
Prob.	0.26644	0.16170	0.57186

*TABLE (1.9):* Calculated probabilities of (x) number of dollars being returned to the Equity tranche from a particular portfolio generated by the t-copula. The estimate of the expected value for the Equity tranche was \$1.15498:

Returns in (\$)	0	1	2
Prob.	0.29987	0.24528	0.45485

#### Discussion of Results:

From these results, it can be concluded that the original hypothesis was only partially correct. The Senior tranche was more likely to be paid the full eight dollars when the portfolios were simulated using the Gaussian copula than any other copula. Meaning that the expected value for the amount of returns the Senior tranche received in this case, was the highest of the three. Therefore, the original hypothesis was correct when it estimated that the Gaussian copula would produce the least number of total bond defaults in a given portfolio.

However, where the hypothesis failed was in comparing how the Clayton copula and the t-copula would influence the final output. It was originally thought that the t-Copula would produce a larger expected value for the Senior tranche than the Clayton copula. However, the observed results have suggested the opposite. Thus, it can be concluded that the bonds generated by means of the Clayton copula are more dependent than those generated by the t-copula.

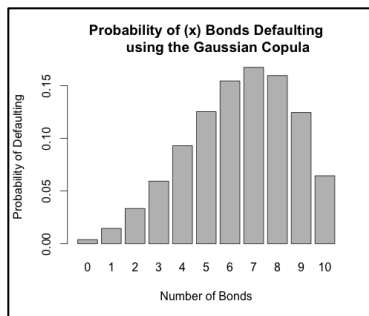
## INVESTIGATION (2): USING ONE QUANTILE FUNCTION

**TRIAL (1):** Using the quantile function for the *chi-squared* distribution to estimate the marginal lifetimes of all bonds: Argument “dist1” is equal to “chi.”

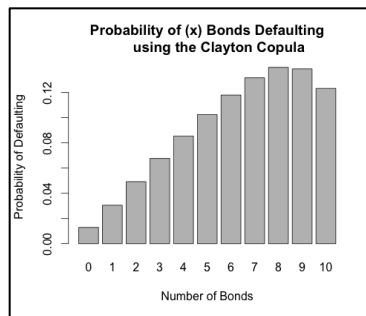
Hypothesis: The number of portfolios with little or no defaults will be significantly less than those with a large number of defaults considering that the chi-squared distribution that is negatively skewed.

Implementation: defaultSim(10, 100000, 10, "chi", 0, 0.2, 0.78, 0.04, 0, 0, 0.5, 0.25, T)

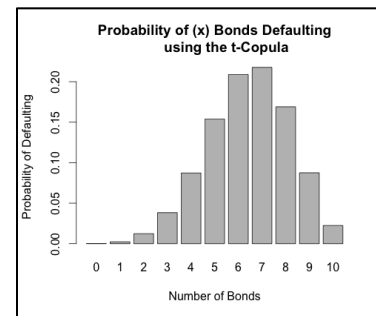
Results for Estimating the Number of Defaults:



(a.2.1)



(b.2.1)



(c.2.1)

FIGURES (a.2.1), (b.2.1), and (c.2.1) are barplots that give the probabilities of a number of defaults occurring in a portfolio when using the quantile function for the chi-squared distribution to estimate the lifetime of a bond. It is obvious from these pictures that the chi-squared quantile function produces lifetimes that are a lot shorter than those produced with the exponential quantile function.

Results for Estimating the Value of a CDO:

TABLE (2.1): Estimated expected values for the amount of returns received by each tranche once the given time horizon was reached:

Type of Copula	Gaussian	Clayton	t-Copula
Expected Value for Senior Traunche	\$3.57761	\$3.52666	\$3.59479
Expected Value for Equity Traunche	\$0.02209	\$0.05642	\$0.00277

Discussion of Results:

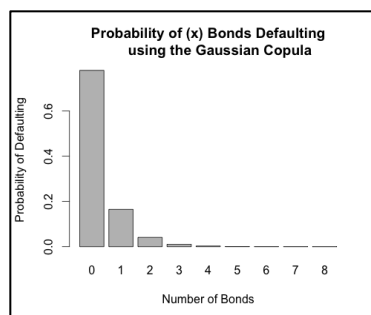
These results are almost exactly opposite of those obtained when using the quantile function for the exponential distribution to calculate the lifetimes of all bonds. In the previous experiment, it was shown that the total number of bonds in a particular portfolio was more likely to be a smaller number. However, in this investigation, a larger number of bonds defaulting in a given portfolio seems to be the norm. Therefore, the hypothesis was indeed correct.

**TRIAL (2):** Using the quantile function for the *lognormal* distribution to estimate the marginal lifetimes of all bonds: Argument “dist1” is equal to “lnorm.”

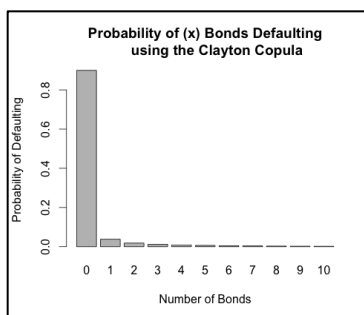
Hypothesis: The number of portfolios with little or no defaults will be slightly greater than those with a large number of defaults; having a distribution that is slightly less positively skewed than that of the exponential distribution.

Implementation: defaultSim(10, 100000, 10, "lnorm", 0, 0.22, 0.78, 0.07, 0, 0, 0, 0.5, 0.25, T)

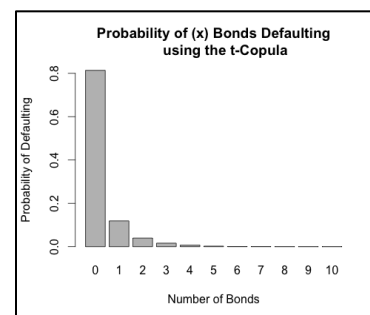
Results for Estimating the Number of Defaults:



(a.2.2)



(b.2.2)



(c.2.2)

FIGURES (a.2.2), (b.2.2), and (c.2.2) are barplots that give the probabilities of a number of defaults occurring in a portfolio when using the quantile function for the lognormal distribution to estimate the lifetime of a bond. These figures look similar to those from the first investigation, but with a steeper descent. However, the same basic behavior is observed in each case.

#### Results for Estimating the Value of a CDO:

TABLE (2.2): Estimated expected values for the amount of returns received by each tranche once the given time horizon was reached:

Type of Copula	Gaussian	Clayton	t-Copula
Expected Value for Senior Tranche	\$7.97938	\$7.86118	\$7.95577
Expected Value for Equity Tranche	\$1.72337	\$1.8378	\$1.74675

#### Discussion of Results:

The hypothesis was slightly incorrect in its assumption that the total number of defaults in a given portfolio would be on average slightly less than that of the lifetimes estimated using the quantile function for the exponential distribution. In fact, the results

illustrate that the tranches will be paid more when the lognormal quantile function is used to estimate the lifetimes of all bonds.

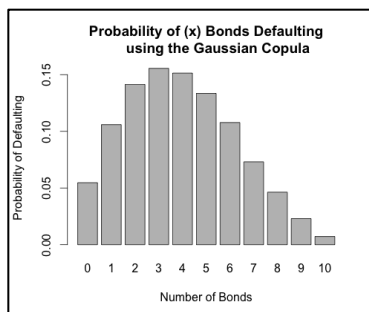
**TRIAL (3):** Using the quantile function for the *Weibull* distribution to estimate the marginal lifetimes of all bonds: Argument “dist1” is equal to “weibull.”

Hypothesis: The probabilities that a given number of bonds defaulting will be approximately normally distributed. Therefore, the tranches will be paid less than that of the tranches in the cases where the exponential quantile function estimates the bond lifetimes, and where that of the lognormal quantile function estimates the bond lifetimes. However, these tranches will be paid slightly more than those whose bonds were estimated using the quantile function for the chi-squared distribution.

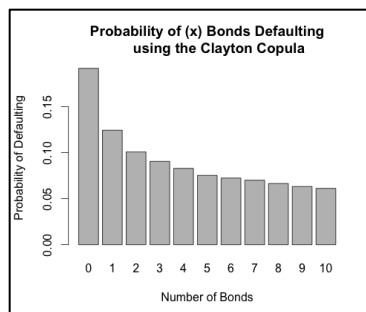
Implementation: defaultSim(10, 100000, 10, "weibull", 0, 0.22, 0.999, 0.005, 0, 0, 0, 0.5, 0.25,

T)

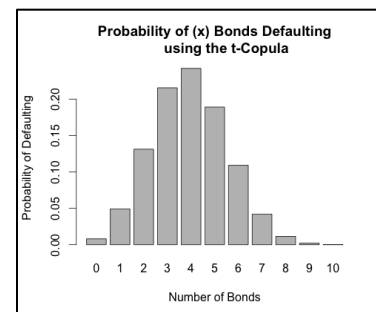
Results for Estimating the Number of Defaults:



(a.2.3)



(b.2.3)



(c.2.3)

FIGURES (a.2.3), (b.2.3), and (c.2.3) are barplots that give the probabilities of a number of defaults occurring in a portfolio when using the quantile function for the weibull distribution to estimate the lifetime of a bond. For the most part, each plot shows an approximately normal distribution for these probabilities, the exception of course being figure (b.2.3).

Results for Estimating the Value of a CDO:

TABLE (2.3): Estimated expected values for the amount of returns received by each tranche once the given time horizon was reached:

Type of Copula	Gaussian	Clayton	t-Copula
Expected Value for Senior Traunche	\$5.84746	\$5.54461	\$6.00199
Expected Value for Equity Traunche	\$0.21531	\$0.50823	\$0.06526

Discussion of Results:

The hypothesis for this trial was exactly right. The probabilities of a given number of bonds defaulting within a specific portfolio whose lifetimes were estimated using the exponential or the lognormal quantile function have, as noted early, a positively skewed distribution. On the other hand, also previously shown, the probabilities of a given number of bonds defaulting within a specific portfolio whose lifetimes were estimated using the chi-squared quantile function have a negatively skewed distribution. Therefore, if the weibull quantile function causes the default probabilities to have an approximately normal distribution, it would make sense that the estimated value of returns would be somewhere in between those estimated using the other quantile functions.

### INVESTIGATION (3): USING TWO QUANTILE FUNCTIONS

**TRIAL (1):** Using the quantile function for the *exponential* distribution to estimate the marginal lifetimes of the first half of bonds in all of the portfolios, and the quantile function for the *chi-squared* distribution to estimate the second half: Argument “dist1” is equal to “exp” and argument “dist2” is equal to “chi.” The common induced correlation is approximately 0.16.

Implementation: defaultSim(10, 100000, 10, "exp", "chi", 0.206, 0.509, 0.17, 0.25, 0.99, 0.1, 0.5, 0.25, F)

Results: As observed, the exponential quantile function creates a distribution of bond defaults that is very positively skewed, and the chi-squared quantile function creates a distribution of bond defaults that is negatively skewed. Therefore, combining the two in one portfolio will result in a slightly less positively skewed distribution for these probabilities in comparison to those lifetimes estimated using only the quantile function of the exponential distribution.

**TRIAL (2):** Using the quantile function for the *exponential* distribution to estimate the marginal lifetimes of the first half of bonds in all of the portfolios, and the quantile function for the *lognormal* distribution to estimate the second half: Argument “dist1” is equal to “exp” and argument “dist2” is equal to “lnorm.” The common induced correlation is approximately 0.16.

Implementation: defaultSim(10, 100000, 10, "exp", "lnorm", 0.206, 0.509, 0.17, 0.25, 0.97, 0.1, 0.5, 0.25, F)

Results: As observed, the exponential quantile function creates a distribution of bond defaults that is positively skewed, and the lognormal quantile function creates a distribution



of bond defaults that is also positively skewed. Therefore, combining the two in one portfolio will result in an even more positively skewed distribution of these probabilities than that of the bond lifetimes created with just the exponential quantile function.

**TRIAL (3):** Using the quantile function for the *exponential* distribution to estimate the marginal lifetimes of the first half of bonds in all of the portfolios, and the quantile function for the *Weibull* distribution to estimate the second half: Argument “dist1” is equal to “exp” and argument “dist2” is equal to “weibull.” The common induced correlation is approximately 0.11.

Implementation: defaultSim(10, 100000, 10, "exp", "weibull", 0.14, 0.3, 0.09, 0.22, 0.999, 0.005, 0.5, 0.25, F)

Results: As observed, the exponential quantile function creates a distribution of bond defaults that is positively skewed, and the Weibull quantile function creates a distribution of bond defaults that is usually approximately normal. Therefore, combining the two in one portfolio will result in a positively skewed distribution of these probabilities.

**TRIAL (4):** Using the quantile function for the *chi-squared* distribution to estimate the marginal lifetimes of the first half of bonds in all of the portfolios, and the quantile function for the *lognormal* distribution to estimate the second half: Argument “dist1” is equal to “chi” and argument “dist2” is equal to “lnorm.” The common induced correlation is approximately 0.12.

Implementation: defaultSim(10, 100000, 10, "chi", "lnorm", 0.2, 0.78, 0.04, 0.19, 0.7, 0.05, 0.5, 0.25, F)

Results: As observed, the chi-squared quantile function creates a distribution of bond defaults that is slightly negatively skewed, and the lognormal quantile function creates a distribution of bond defaults that is rather positively skewed. Therefore, combining the two in one portfolio will result in a positively skewed distribution of these probabilities that is slightly less skewed than that of the probabilities resulting solely from the lognormal distribution.

**TRIAL (5):** Using the quantile function for the *chi-squared* distribution to estimate the marginal lifetimes of the first half of bonds in all of the portfolios, and the quantile function for the *Weibull* distribution to estimate the second half: Argument “dist1” is equal to “chi” and argument “dist2” is equal to “weibull.” The common induced correlation is approximately 0.11.

Implementation: defaultSim(10, 100000, 10, "chi", "weibull", 0.18, 0.76, 0.03, 0.22, 0.999, 0.005, 0.5, 0.25, F)

Results: As observed, the chi-squared quantile function creates a distribution of bond defaults that is negatively skewed, and the Weibull quantile function creates a distribution of bond defaults that is usually approximately normal. Therefore, combining the two in one portfolio will result in a slightly positively skewed distribution of these probabilities.

**TRIAL (6):** Using the quantile function for the *lognormal* distribution to estimate the marginal lifetimes of the first half of bonds in all of the portfolios, and the quantile function for the *Weibull* distribution to estimate the second half: Argument “dist1” is equal to “lnorm” and argument “dist2” is equal to “weibull.” The common induced correlation is approximately 0.11.

Implementation: defaultSim(10, 100000, 10, "lnorm", "weibull", 0.18, 0.6, 0.04, 0.22, 0.999, 0.005, 0.5, 0.25, F)

Results: As observed, the lognormal quantile function creates a distribution of bond default probabilities that is slightly positively skewed, and the Weibull quantile function creates a distribution of bond default probabilities that is approximately normal. Therefore, combining the two in one portfolio will result in a distribution that is ever so slightly positively skewed for these probabilities.

## V. SIMULATIONS UTILIZING HISTORICAL DATA INFORMATION

### OVERVIEW OF “THE ECONOMICS OF STRUCTURED FINANCE”

The purpose of Joshua Coval, Jakub Jurek, and Erik Stafford’s article entitled “The Economics of Structured Finance” is to describe credit ratings and to examine how “modest imprecision in evaluating risks and their exposure to systematic risks” (2) contribute to the sporadic behavior of structured finance. What is of particular interest in this piece is the simulation spelled out on page nine of Coval et al. The authors demonstrated the sensitivity of the CDOs to errors in parameter estimates by simulating payoffs to 40 CDO pools. Each pool contained “100 bonds with a five-year default probability of five percent and a recovery rate of 50 percent of face value conditional on default” (9). With the annualized default rates obtained from historical data for BB+ credit rated bonds and a fixed pairwise bond default correlation of 0.20, Coval et al. estimated the lifetimes of these bonds and then applied these results to a capital structure. This structure included three prioritized tranches: the “junior tranche” being the first to experience loss if the total loss of the portfolio is larger than six

percent, the “mezzanine tranche” which experiences loss when the portfolio loss ranges from six to twelve percent, and the “senior tranche” which only experiences loss if the portfolio loss is greater than twelve percent.

## MODEL MODIFICATIONS

To provide a more realistic and informative example, an investigation using the information and design of the Coval et al. simulation recounted above. In order to accomplish this, the code utilized previously was modified to accommodate the needs of this example. First, the parameter values for the marginal lifetimes had to be calculated. This was done by setting letting the continuous random variable  $X_i$  be equal to the lifetime of bond  $i$  in years. Next, assuming that the marginal distribution is exponential, the copula parameters for each of the three copulas were determined in the following manner.

- (1.) Setting  $P(X_i \leq 1)$  equal to 0.1018 (the annualized default rate provided by Coval et al.) and solving for  $\beta$ :

$$0.0107 = P(X_i \leq 1) = 1 - e^{-\beta}$$

$$0.0107 - 1 = -e^{-\beta}$$

$$0.9893 = e^{-\beta}$$

$$\ln(0.9893) = -\beta$$

- (2.) This value was then used in the code previously created to generate correlation matrices for each set of copula estimates. Parameter values were finalized once all of the values within each correlation matrix were approximately equal to 0.20. The finalized results for these parameter estimates are listed in the table below. See appendix for modified code of the function “corCopula().”

Copula Type	Gaussian	Clayton	t-Copula
Marginal Lifetime Parameter Estimate	0.239	0.640	0.186

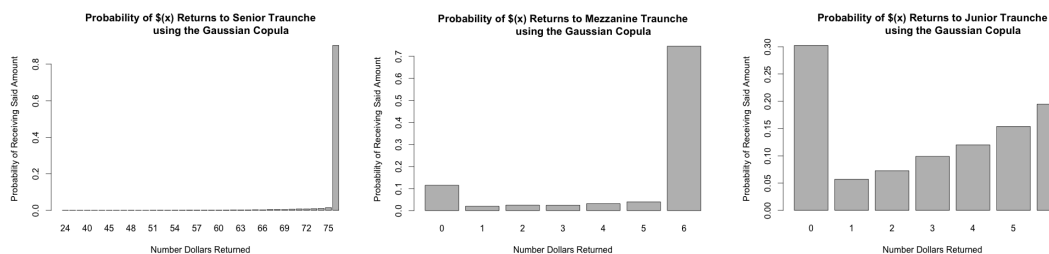
(3.) Finally, the original function “defaultSim()” was manipulated to incorporate three tranches in the toy model for CDO with attachment points identical to those of the Coval et al. simulation. The modified model was then implemented using the following R command:

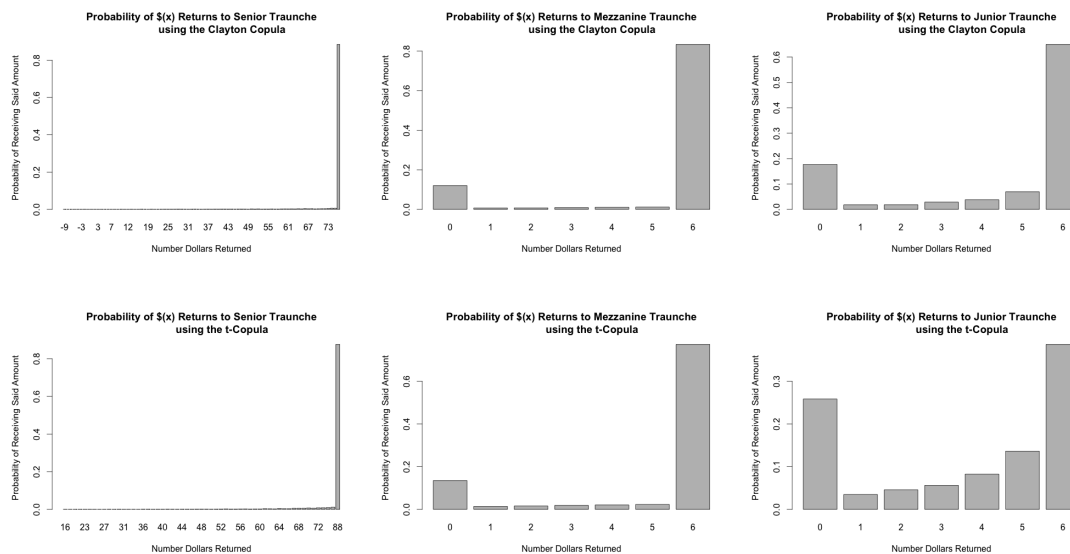
```
> defaultSim(10, 10000, 100, "exp", 0, 0.239, 0.640, 0.186, 0, 0, 0, 0.2, 1, T)
```

where the argument for the number of portfolios generated was changed to 10000, the number of bonds in a portfolio to 100, the lifetime parameter to 0.05, the time horizon to one year, and the copula parameters to the values found in the previous table.

## SIMULATION RESULTS

The bar plots below show the probabilities of each tranche recovering a certain amount of their investment over the course of a year as estimated by the three separate copulas:





More precisely, the following estimations of the expected values for each of the three tranches were found to be the following:

Estimation Method	Senior Tranche	Mezzanine Tranche	Junior Tranche
Gaussian Copula	\$86.1063	\$4.9394	\$2.9179
Clayton Copula	\$84.0937	\$5.1535	\$4.5407
t-Copula	\$85.0234	\$4.9341	\$3.623

These results support Coval's et al. argument that structured finance allows for the credit enhancement of the Senior claims. Finally, this example supports the overarching hypothesis of the Coval et al. paper because the slightest change in the parameter values created significantly different results. For instance, if the parameter values for each of the copulas were chosen or estimated incorrectly a proper correlation matrix would not result and thus the estimates of the lifetimes of each bond would be incorrect.

## VI. CONCLUSION

### MODEL CRITIQUE

This is a very simple model for estimating bond default as well as estimating the value of a CDO. However, the model can be revised to include more bonds as well as more tranches. It could also be altered to account a larger selection of various quantile functions for calculating the marginal lifetimes of each bond. Perhaps it could be changed so that it can assign different quantile functions to estimate lifetimes for each bond rather than limiting it to only half of the portfolio being estimated by one distribution and the other half by another. Another critique of this model is that it includes a large number of arguments. While this was beneficial for the purposes of this report, if a user is not as familiar with R or with the theory behind the creation of the model, it could easily be misused. The last perhaps undesirable feature of this function is that the toy model for estimating value of CDO can become very computationally expensive if the number entered for “N” is extremely large. The reason why this is a weakness is because ideally “N” should be very large in order to ensure that the probabilities that are gleaned from experimentation are as accurate as possible.

### DISCUSSION OF DIFFERENT COPULAS AND CDO

Overall the investigations provided in the previous section offered significant illustrations and comments on how returns are influenced by bond default, and how the total number of defaults in a specified portfolio can be estimated in various ways. Thus, it is very important for a user to know what factors will come into play for their specific portfolio in order for them to make accurate predictions. For example, they would need to know

whether or not an implicit copula, like the Gaussian copula or the t-Copula, would be more desirable or whether or not an explicit copula, like the Clayton copula, would be better. If the bonds depend more heavily on one another, than using the Clayton would be more beneficial than using one of the other two copulas discussed. Another important component of creating an ideal model would be the number of tranches involved and how to distribute the returns among them when the time horizon has expired. Take for instance, a case where a third tranche is added to the structure of a portfolio. Assume that this tranche has second priority; this will then negatively impact the Equity tranche because it has least priority. In conclusion, copulas are a very effective method for modeling bond defaults and as a result the value of CDO.



## VII. REFERENCES

- Beare, Brenden K. "Archimedean Copulas and Temporal Dependence." *Economic Theory* 28.06 (2012): 1165-185. *Cambridge Journals*. Cambridge University Press, Apr. 2012. Web. 14 Dec. 2013.  
<<http://journals.cambridge.org.proxy.library.nd.edu/action/displayAbstract?fromPage=online&aid=8732518>>.
- Coval, Joshua, Jakub Jurek, and Erik Stafford. "The Economics of Structured Finance." *Journal of Economic Perspectives* 23.1 (2009): 3-25. Print.
- Demarta, Stefano, and Alexander J. McNeil. "The T Copula and Related Copulas." *International Statistical Review* 73.1 (2005): 111-29.  
<<http://www.macs.hw.ac.uk/~mcneil/ftp/tCopula.pdf>> Web. 16 Dec. 2013.
- Duffie, Darrell, and Nicolae Gârleanu. "Risk and Valuation of Collateralized Debt Obligations." *Financial Analysts Journal* 57.1 (2001): 41-59. Web. 15 Dec. 2013.  
<<http://faculty.haas.berkeley.edu/garleanu/ddng.pdf>>.
- Fabrizio Durante, Piotr Jaworski, Radko Mesiar. "Invariant Dependence Structures and Archimedean Copulas." *Statistics & Probability Letters* 81.12 (2011): 1995-2003. *Science Direct*. Web. 14 Dec. 2013. <<http://dx.doi.org/10.1016/j.spl.2011.08.018>>.
- Habiboellah, Farzia. "Copulas: Modeling Dependencies in Financial Risk Management." Thesis. Vrije Universiteit in Amsterdam, 2007. Web. 15 Dec. 2013.  
<<http://www.math.vu.nl/~sbhulai/theses/werkstuk-habiboelah.pdf>>.
- McNeil, Alexander J., Rüdiger Frey, and Paul Embrechts. *Quantitative Risk Management: Concepts, Techniques and Tools*. Princeton, NJ: Princeton UP, 2005. Print.
- Pareek, Mukul. "Distributions in Finance." *Distributions in Finance*. Risk Prep, 21 Oct. 2009.

Web. 14 Dec. 2013. <<https://www.riskprep.com/all-tutorials/36-exam-22/63-distributions-in-finance>>.

## VIII. APPENDIX

### I. ORIGINAL MODEL

```
# This function creates "N" number of portfolios that contain a certain number of
# bonds as defined by the second argument "bonds." These are randomly generated
# by means of the Gaussian copula and the Clayton copula and the t-copula with the
# copula parameters "gauss.par.1," "clay.par.1," "t.par.1," "gauss.par.2," "clay.par.2,"
# and "t.par.2" using a certain "seed" value. The lifetimes of these bonds are then
# obtained by utilizing the arguments "dist1," "dist2," and "life.par." If "dist1" is
# given and "dist2" equals zero, then all bonds have the same marginal lifetime
# distribution. If "bonds" is even and both "dist1" and "dist2" are given, then the
# first half of the portfolio has a marginal lifetime distribution as defined by "dist1"
# and the second half by "dist2." Lastly, "life.par" is the parameter for marginal
# lifetimes. Once the portfolios have been created and the number of defaults
# determined, if the "toy.model" argument is equal to TRUE, a toy model value
# of a CDO will be implemented. In this simple model, 80% of a given
# portfolio is owned by Senior investors and 20% of the portfolio is owned by
# Equity investors. If 0.8*bonds is not equal to a whole number, an error
# message will be printed.
```

```
defaultSim = function(seed, N, bonds, dist1, dist2, gauss.par.1, clay.par.1, t.par.1,
                      gauss.par.2, clay.par.2, t.par.2, life.par, horizon, toy.model=F)
{
  # Set seed
  if (is.numeric(seed) == TRUE)
  {
    set.seed(seed)
  }
  else
  {
    stop("Error: Please enter a numeric value for seed.")
  }

  # Create a random Gaussian copula, a random Clayton copula and
  # a random t-copula to determine lifetimes of all bonds based off
  # of dist1 when only given the argument "dist1"
  if (dist2 == 0)
  {

    # Create a random copulas
    library(copula)
    myGaussian = rCopula(N, normalCopula(gauss.par.1, bonds))
    myClayton = rCopula(N, claytonCopula(clay.par.1, bonds))
    myT = rCopula(N, tCopula(t.par.1, bonds))

    if (dist1 == "exp")
```

```

{
  myGaussianlifetimes = qexp(myGaussian, life.par)
  myClaylifetimes = qexp(myClayton, life.par)
  myTlifetimes = qexp(myT, life.par)
}
else if (dist1 == "chi")
{
  myGaussianlifetimes = qchisq(myGaussian, life.par)
  myClaylifetimes = qchisq(myClayton, life.par)
  myTlifetimes = qchisq(myT, life.par)
}
else if (dist1 == "lnorm")
{
  myGaussianlifetimes = qlnorm(myGaussian, life.par)
  myClaylifetimes = qlnorm(myClayton, life.par)
  myTlifetimes = qlnorm(myT, life.par)
}
else if (dist1 == "weibull")
{
  myGaussianlifetimes = qweibull(myGaussian, life.par)
  myClaylifetimes = qweibull(myClayton, life.par)
  myTlifetimes = qweibull(myT, life.par)
}
else if (dist1 != "exp" || dist1 != "chi" || dist1 != "lnorm" || dist1 != "weibull")
{
  stop("Error: Please enter either exp, chi, lnorm, or weibull for argument dist1.")
}
}

else if (bonds%%2 == 0)
{
  # Calculate half the number of bonds entered
  half = bonds/2

  # Set seed
  set.seed(seed)

  # Generate both halves of the portfolios using copulas
  library(copula)

  myGaussianFirstHalf = rCopula(N, normalCopula(gauss.par.1, half))
  myGaussianSecHalf = rCopula(N, normalCopula(gauss.par.2, half))
  myGaussian = cbind(myGaussianFirstHalf, myGaussianSecHalf)

  myClaytonFirstHalf = rCopula(N, claytonCopula(clay.par.1, half))
  myClaytonSecHalf = rCopula(N, claytonCopula(clay.par.2, half))
  myClayton = cbind(myClaytonFirstHalf, myClaytonSecHalf)

```

```

myTFirstHalf = rCopula(N, tCopula(t.par.1, half))
myTSecHalf = rCopula(N, tCopula(t.par.2, half))
myT = cbind(myTFirstHalf, myTSecHalf)

# Determine lifetimes of the first half of the portfolio from dist1
if (dist1 == "exp")
{
  myGaussianlifetimes = qexp(myGaussian[, 1:half], life.par)
  myClaylifetimes = qexp(myClayton[, 1:half], life.par)
  myTlifetimes = qexp(myT[, 1:half], life.par)
}
else if (dist1 == "chi")
{
  myGaussianlifetimes = qchisq(myGaussian[, 1:half], life.par)
  myClaylifetimes = qchisq(myClayton[, 1:half], life.par)
  myTlifetimes = qchisq(myT[, 1:half], life.par)
}
else if (dist1 == "lnorm")
{
  myGaussianlifetimes = qlnorm(myGaussian[, 1:half], life.par)
  myClaylifetimes = qlnorm(myClayton[, 1:half], life.par)
  myTlifetimes = qlnorm(myT[, 1:half], life.par)
}
else if (dist1 == "weibull")
{
  myGaussianlifetimes = qweibull(myGaussian[, 1:half], life.par)
  myClaylifetimes = qweibull(myClayton[, 1:half], life.par)
  myTlifetimes = qweibull(myT[, 1:half], life.par)
}
else if (dist1 != "exp" || dist1 != "chi" || dist1 != "lnorm" || dist1 != "weibull")
{
  stop("Error: Please enter either exp, chi, lnorm, or weibull for argument dist1.")
}

# Determine lifetimes of the second half of the portfolio from dist2
if (dist2 == "exp")
{
  myGaussianlifetimes = qexp(myGaussian[, half:bonds], life.par)
  myClaylifetimes = qexp(myClayton[, half:bonds], life.par)
  myTlifetimes = qexp(myT[, half:bonds], life.par)
}
else if (dist2 == "chi")
{
  myGaussianlifetimes = qchisq(myGaussian[, half:bonds], life.par)
  myClaylifetimes = qchisq(myClayton[, half:bonds], life.par)
  myTlifetimes = qchisq(myT[, half:bonds], life.par)
}
else if (dist2 == "lnorm")

```

```

{
  myGaussianlifetimes = qlnorm(myGaussian[, half:bonds], life.par)
  myClaylifetimes = qlnorm(myClayton[, half:bonds], life.par)
  myTlifetimes = qlnorm(myT[, half:bonds], life.par)
}
else if (dist2 == "weibull")
{
  myGaussianlifetimes = qweibull(myGaussian[, half:bonds], life.par)
  myClaylifetimes = qweibull(myClayton[, half:bonds], life.par)
  myTlifetimes = qweibull(myT[, half:bonds], life.par)
}
else if (dist2 != "exp" || dist1 != "chi" || dist1 != "lnorm" || dist1 != "weibull")
{
  stop("Error: Please enter either exp, chi, lnorm, weibull, or zero for argument dist2.")
}
}
else
{
  stop("Error: Bond number must be even in order to create a split portfolio.")
}

```

```

# Find the number of defaults as estimated by both copulas

```

```

defaultGauss = myGaussianlifetimes < horizon
numDefaultsGauss = apply(defaultGauss, 1, sum)
defaultClay = myClaylifetimes < horizon
numDefaultsClay = apply(defaultClay, 1, sum)
defaultT = myTlifetimes < horizon
numDefaultsT = apply(defaultT, 1, sum)

```

```

# Calculate the probability of a given number of bonds defaulting

```

```

probsGauss = table(numDefaultsGauss)/N
probsClay = table(numDefaultsClay)/N
probsT = table(numDefaultsT)/N

```

```

# Output results

```

```

barplot(probsGauss, main="Probability of (x) Bonds Defaulting
  using the Gaussian Copula", xlab="Number of Bonds", ylab=
  "Probability of Defaulting")
print(probsGauss)
barplot(probsClay, main="Probability of (x) Bonds Defaulting
  using the Clayton Copula", xlab="Number of Bonds",
  ylab="Probability of Defaulting")
print(probsClay)
barplot(probsT, main="Probability of (x) Bonds Defaulting
  using the t-Copula", xlab="Number of Bonds", ylab=
  "Probability of Defaulting")
print(probsT)

```

```

# Toy Model Value of a C.D.O.
# Traunches = "senior" and "equity"
if (toy.model == TRUE)
{
  # Calculate percentage of Senior bonds and percentage of Equity bonds
  senior = 0.8*bonds
  equity = 0.2*bonds
  is.wholenumber = function(x, tol = .Machine$double.eps^0.5) abs(x - round(x)) < tol

  # Ensure that the portfolio can be split 80:20
  if (is.wholenumber(senior) == FALSE)
  {
    stop("Error 0.8*bonds must be equal to a whole number; cannot run toy model.")
  }
  else
  {
    # Define variables for returns as determined by the Gaussian
    # copula
    seniorGauss = c()
    equityGauss = c()

    # Paying each Investor based off of the number of defaults as
    # determined by the Gaussian copula
    for (i in 1:N)
    {
      # If less than 20% of the bonds default, both traunches are paid
      if (numDefaultsGauss[i] < equity)
      {
        equity2 = bonds - numDefaultsGauss[i] - senior
        seniorGauss = rbind(seniorGauss, senior)
        equityGauss = rbind(equityGauss, equity2)
      }

      # Else if more than 20% of the bonds default, the Senior traunche
      # is paid the number of bonds minus the number of defaults and the Equity
      # traunche is not paid at all
      else if (numDefaultsGauss[i] >= equity)
      {
        seniorGauss = rbind(seniorGauss, bonds-numDefaultsGauss[i])
        equityGauss = rbind(equityGauss, 0)
      }

      # Print what percentage has been completed
      if (i%%(0.1*N) == 0)
      {
        print(paste(100*(i/N), "% complete when using Gaussian Copula."))
      }
    }
  }
}

```

```

# Index counter
i = i+1
}

# Calculate the returns probability for each tranche
probsSeniorGauss = table(seniorGauss)/N
probsEquityGauss = table(equityGauss)/N

# Calculate the expected value for each tranche
expectedSeniorGauss = sum(probsSeniorGauss*as.integer(names(probsSeniorGauss)))
expectedEquityGauss = sum(probsEquityGauss*as.integer(names(probsEquityGauss)))

# Print results when utilizing the Gaussian copula
barplot(probsSeniorGauss, main="Probability of $(x) Returns to Senior Traunche
using the Gaussian Copula", xlab="Number Dollars Returned", ylab=
"Probability of Receiving Said Amount")
print(probsSeniorGauss)
print(paste(expectedSeniorGauss, "= Estimate of Expected value for Senior Traunche
(Gaussian)"))
barplot(probsEquityGauss, main="Probability of $(x) Returns to Equity Traunche
using the Gaussian Copula", xlab="Number Dollars Returned", ylab=
"Probability of Receiving Said Amount")
print(probsEquityGauss)
print(paste(expectedEquityGauss, "= Estimate of Expected value for Equity Traunche
(Gaussian)"))

# Define variables for returns as determined by the Clayton
# copula
seniorClay = c()
equityClay = c()

# Paying each Investor based off of the number of defaults as
# determined by the Clayton copula
for (i in 1:N)
{
# If less than 20% of the bonds default, both tranches are paid
if (numDefaultsClay[i] < equity)
{
equity2 = bonds - numDefaultsClay[i] - senior
seniorClay = rbind(seniorClay, senior)
equityClay = rbind(equityClay, equity2)
}
}

# Else if more than 20% of the bonds default, the Senior tranche
# is paid the number of bonds minus the number of defaults and the Equity
# tranche is not paid at all
else if (numDefaultsClay[i] >= equity)
{

```



```

seniorClay = rbind(seniorClay, bonds-numDefaultsClay[i])
equityClay = rbind(equityClay, 0)
}

# Print what percentage has been completed
if (i%%(0.1*N) == 0)
{
  print(paste(100*(i/N), "% complete when using Clayton Copula."))
}

# Index counter
i = i+1
}

# Calculate the returns probability for each tranche
probsSeniorClay = table(seniorClay)/N
probsEquityClay = table(equityClay)/N

# Calculate the expected value for each tranche
expectedSeniorClay = sum(probsSeniorClay*as.integer(names(probsSeniorClay)))
expectedEquityClay = sum(probsEquityClay*as.integer(names(probsEquityClay)))

# Print results when utilizing the Clayton copula
barplot(probsSeniorClay, main="Probability of $(x) Returns to Senior Traunche
using the Clayton Copula", xlab="Number Dollars Returned", ylab=
"Probability of Receiving Said Amount")
print(probsSeniorClay)
print(paste(expectedSeniorClay, "= Estimate of Expected value for Senior Traunche
(Clayton)"))
barplot(probsEquityClay, main="Probability of $(x) Returns to Equity Traunche
using the Clayton Copula", xlab="Number Dollars Returned", ylab=
"Probability of Receiving Said Amount")
print(probsEquityClay)
print(paste(expectedEquityClay, "= Estimate of Expected value for Equity Traunche
(Clayton)"))

# Define variables for returns as determined by the t-copula
seniorT = c()
equityT = c()

# Paying each Investor based off of the number of defaults as
# determined by the t-copula
for (i in 1:N)
{
  # If less than 20% of the bonds default, both traunches are paid
  if (numDefaultsT[i] < equity)
  {
    equity2 = bonds - numDefaultsT[i] - senior

```

```

    seniorT = rbind(seniorT, senior)
    equityT = rbind(equityT, equity2)
  }

  # Else if more than 20% of the bonds default, the Senior tranche
  # is paid the number of bonds minus the number of defaults and the Equity
  # tranche is not paid at all
  else if (numDefaultsT[i] >= equity)
  {
    seniorT = rbind(seniorT, bonds-numDefaultsT[i])
    equityT = rbind(equityT, 0)
  }

  # Print what percentage has been completed
  if (i%%(0.1*N) == 0)
  {
    print(paste(100*(i/N), "% complete when using Clayton Copula."))
  }

  # Index counter
  i = i+1
}

# Calculate the returns probability for each tranche
probsSeniorT = table(seniorT)/N
probsEquityT = table(equityT)/N

# Calculate the expected value of each tranche
expectedSeniorT = sum(probsSeniorT*as.integer(names(probsSeniorT)))
expectedEquityT = sum(probsEquityT*as.integer(names(probsEquityT)))

# Print results when utilizing the t-copula
barplot(probsSeniorT, main="Probability of $(x) Returns to Senior Traunche
  using the t-Copula", xlab="Number Dollars Returned", ylab=
  "Probability of Receiving Said Amount")
print(probsSeniorT)
print(paste(expectedSeniorT, "= Estimate of Expected value for Senior Traunche
  (t-Copula)"))
barplot(probsEquityT, main="Probability of $(x) Returns to Equity Traunche
  using the t-Copula", xlab="Number Dollars Returned", ylab=
  "Probability of Receiving Said Amount")
print(probsEquityT)
print(paste(expectedEquityT, "= Estimate of Expected value for Equity Traunche
  (t-Copula)"))
}
}
}

```

## II. ESTIMATING PARAMETER VALUES

```
# This function creates sets of portfolios generated by three different copulas,
# identical to those generated in the original model implemented by "defaultSim()."
# Then, again like the previous function, this function estimates the lifetimes
# of all bonds for each grouping of portfolios by means of a specific quantile
# function. Since this is a much simpler function than the last, and since it is
# not part of the main body of code, the quantile function is edited manually as
# need. In this particular showing, the quantile function for the exponential
# distribution is shown.
```

```
corCopula = function(N, d, gauss.par, clay.par, t.par)
{
  # Set seed
  set.seed(10)

  # Generate portfolios using the copulas
  library(copula)
  myGaussian=rCopula(N,normalCopula(gauss.par, d))
  myClayton = rCopula(N,claytonCopula(clay.par, d))
  myT = rCopula(N, tCopula(t.par, d))

  # Estimate lifetimes of all bonds
  myGaussianlifetimes=qexp(myGaussian, 0.5)
  myClaylifetimes = qexp(myClayton, 0.5)
  myTlifetimes = qexp(myT, 0.5)

  # Compute correlation matrices for each data set
  cor1 = cor(myGaussianlifetimes)
  cor2 = cor(myClaylifetimes)
  cor3 = cor(myTlifetimes)

  # Print correlations matrices
  print(cor1)
  print(cor2)
  print(cor3)
}
```

## III. MODIFIED CODE FOR ESTIMATING PARAMETER VALUES

```
corCopula = function(N, d, gauss.par, clay.par, t.par)
{
  # Set seed
  set.seed(10)

  # Generate portfolios using the copulas
```

```

library(copula)
myGaussian=rCopula(N,normalCopula(gauss.par, d))
myClayton = rCopula(N,claytonCopula(clay.par, d))
myT = rCopula(N, tCopula(t.par, d))

# Estimate lifetimes of all bonds
marginal = -log(0.9893)
myGaussianlifetimes=qexp(myGaussian, marginal)
myClaylifetimes = qexp(myClayton, marginal)
myTlifetimes = qexp(myT, marginal)

# Compute correlation matrices for each data set
cor1 = cor(myGaussianlifetimes)
cor2 = cor(myClaylifetimes)
cor3 = cor(myTlifetimes)

# Print correlations matrices
print(cor1)
print(cor2)
print(cor3)
}

```

#### IV. MODIFIED ORIGINAL MODEL

# The R code listed in this section omits the majority of the text that defines the function # defaultSim() since it is same as that of the original model. The only modifications that # have been made are to the toy model of CDO.

```

defaultSim = function(seed, N, bonds, dist1, dist2, gauss.par.1, clay.par.1, t.par.1,
                      gauss.par.2, clay.par.2, t.par.2, life.par, horizon, toy.model=F)
{
  ...
  # Toy Model Value of a C.D.O.
  # Traunches = "senior," "mezzanine," and "junior"
  if (toy.model == TRUE)
  {
    # Calculate percentage of Senior bonds and percentage of Equity bonds
    senior = 0.88*bonds
    mezzanine = 0.06*bonds
    junior = 0.06*bonds
    is.wholenumber = function(x, tol = .Machine$double.eps^0.5) abs(x - round(x)) < tol

    # Ensure that the portfolio can be split 80:20
    if (is.wholenumber(senior) == FALSE)
    {
      stop("Error 0.88*bonds must be equal to a whole number; cannot run toy model.")
    }
  }
}

```

```

else
{
  # Define variables for returns as determined by the Gaussian
  # copula
  seniorGauss = c()
  mezzanineGauss = c()
  juniorGauss = c()

  # Paying each Investor based off of the number of defaults as
  # determined by the Gaussian copula
  for (i in 1:N)
  {
    # If less than 6% of the bonds default, all tranches are paid
    if (numDefaultsGauss[i] <= junior)
    {
      junior2 = bonds - numDefaultsGauss[i] - senior - mezzanine
      seniorGauss = rbind(seniorGauss, senior)
      mezzanineGauss = rbind(mezzanineGauss, mezzanine)
      juniorGauss = rbind(juniorGauss, junior2)
    }

    # Else if more than 6%, but less than 12% of the bonds default only the senior
    # tranche and the mezzanine tranche are paid
    else if (numDefaultsGauss[i] > junior & numDefaultsGauss[i] <= 2*mezzanine)
    {
      mezzanine2 = bonds - numDefaultsGauss[i] - senior
      seniorGauss = rbind(seniorGauss, senior)
      mezzanineGauss = rbind(mezzanineGauss, mezzanine2)
      juniorGauss = rbind(juniorGauss, 0)
    }

    # Else if more than 12% of the bonds default, only the senior tranche is paid
    else if (numDefaultsGauss[i] > 2*mezzanine)
    {
      senior2 = senior - numDefaultsGauss[i]
      seniorGauss = rbind(seniorGauss, senior2)
      mezzanineGauss = rbind(mezzanineGauss, 0)
      juniorGauss = rbind(juniorGauss, 0)
    }

    # Print what percentage has been completed
    if (i%%(0.1*N) == 0)
    {
      print(paste(100*(i/N), "% complete when using Gaussian Copula."))
    }

    # Index counter
    i = i+1
  }
}

```

```

}

# Calculate the returns probability for each tranche
probsSeniorGauss = table(seniorGauss)/N
probsMezzanineGauss = table(mezzanineGauss)/N
probsJuniorGauss = table(juniorGauss)/N

# Calculate the expected value for each tranche
expectedSeniorGauss = sum(probsSeniorGauss*as.integer(names(probsSeniorGauss)))
expectedMezzanineGauss =
  sum(probsMezzanineGauss*as.integer(names(probsMezzanineGauss)))
expectedJuniorGauss = sum(probsJuniorGauss*as.integer(names(probsJuniorGauss)))

# Print results when utilizing the Gaussian copula
barplot(probsSeniorGauss, main="Probability of $(x) Returns to Senior Traunche
      using the Gaussian Copula", xlab="Number Dollars Returned", ylab=
      "Probability of Receiving Said Amount")
print(probsSeniorGauss)
print(paste(expectedSeniorGauss, "= Estimate of Expected value for Senior Traunche
      (Gaussian)"))
barplot(probsMezzanineGauss, main="Probability of $(x) Returns to Mezzanine
      Traunche using the Gaussian Copula", xlab="Number Dollars Returned", ylab=
      "Probability of Receiving Said Amount")
print(probsMezzanineGauss)
print(paste(expectedMezzanineGauss, "= Estimate of Expected value for Mezzanine
      Traunche (Gaussian)"))
barplot(probsJuniorGauss, main="Probability of $(x) Returns to Junior Traunche
      using the Gaussian Copula", xlab="Number Dollars Returned", ylab=
      "Probability of Receiving Said Amount")
print(probsJuniorGauss)
print(paste(expectedJuniorGauss, "= Estimate of Expected value for Junior Traunche
      (Gaussian)"))

# Define variables for returns as determined by the Clayton
# copula
seniorClay = c()
mezzanineClay = c()
juniorClay = c()

# Paying each Investor based off of the number of defaults as
# determined by the Clayton copula
for (i in 1:N)
{
  # If less than 6% of the bonds default, all tranches are paid
  if (numDefaultsClay[i] <= junior)
  {
    junior2 = bonds - numDefaultsClay[i] - senior - mezzanine
    seniorClay = rbind(seniorClay, senior)
  }
}

```

```

mezzanineClay = rbind(mezzanineClay, mezzanine)
juniorClay = rbind(juniorClay, junior2)
}

# Else if more than 6%, but less than 12% of the bonds default, only the senior
# traunche and the mezzanine traunche are paid
else if (numDefaultsClay[i] > junior & numDefaultsClay[i] <= 2*mezzanine)
{
  mezzanine2 = bonds - numDefaultsClay[i] - senior
  seniorClay = rbind(seniorClay, senior)
  mezzanineClay = rbind(mezzanineClay, mezzanine2)
  juniorClay = rbind(juniorClay, 0)
}

# Else if more than 12% of the bonds default, only the senior traunche is paid
else if (numDefaultsClay[i] > 2*mezzanine)
{
  senior2 = senior - numDefaultsClay[i]
  seniorClay = rbind(seniorClay, senior2)
  mezzanineClay = rbind(mezzanineClay, 0)
  juniorClay = rbind(juniorClay, 0)
}

# Print what percentage has been completed
if (i%%(0.1*N) == 0)
{
  print(paste(100*(i/N), "% complete when using Clayton Copula."))
}

# Index counter
i = i+1
}

# Calculate the returns probability for each traunche
probsSeniorClay = table(seniorClay)/N
probsMezzanineClay = table(mezzanineClay)/N
probsJuniorClay = table(juniorClay)/N

# Calculate the expected value for each traunche
expectedSeniorClay = sum(probsSeniorClay*as.integer(names(probsSeniorClay)))
expectedMezzanineClay =
  sum(probsMezzanineClay*as.integer(names(probsMezzanineClay)))
expectedJuniorClay = sum(probsJuniorClay*as.integer(names(probsJuniorClay)))

# Print results when utilizing the Clayton copula
barplot(probsSeniorClay, main="Probability of $(x) Returns to Senior Traunche
  using the Clayton Copula", xlab="Number Dollars Returned", ylab=
  "Probability of Receiving Said Amount")

```

```

print(probsSeniorClay)
print(paste(expectedSeniorClay, "= Estimate of Expected value for Senior Traunche
(Clayton)"))
barplot(probsMezzanineClay, main="Probability of $(x) Returns to Mezzanine
Traunche using the Clayton Copula", xlab="Number Dollars Returned", ylab=
"Probability of Receiving Said Amount")
print(probsMezzanineClay)
print(paste(expectedMezzanineClay, "= Estimate of Expected value for Mezzanine
Traunche (Clayton)"))
barplot(probsJuniorClay, main="Probability of $(x) Returns to Junior Traunche
using the Clayton Copula", xlab="Number Dollars Returned", ylab=
"Probability of Receiving Said Amount")
print(probsJuniorClay)
print(paste(expectedJuniorClay, "= Estimate of Expected value for Junior Traunche
(Clayton)"))

# Define variables for returns as determined by the t-copula
seniorT = c()
mezzanineT = c()
juniorT = c()

# Paying each Investor based off of the number of defaults as
# determined by the t-copula
for (i in 1:N)
{
  # If less than 6% of the bonds default, all tranches are paid
  if (numDefaultsT[i] <= junior)
  {
    junior2 = bonds - numDefaultsT[i] - senior - mezzanine
    seniorT = rbind(seniorT, senior)
    mezzanineT = rbind(mezzanineT, mezzanine)
    juniorT = rbind(juniorT, junior2)
  }

  # Else if more than 6%, but less than 12% of the bonds default, only the senior
  # traunche and the mezzanine traunche are paid
  else if (numDefaultsT[i] > junior & numDefaultsT[i] <= 2*mezzanine)
  {
    mezzanine2 = bonds - numDefaultsT[i] - senior
    seniorT = rbind(seniorT, senior)
    mezzanineT = rbind(mezzanineT, mezzanine2)
    juniorT = rbind(juniorT, 0)
  }

  # Else if more than 12% of the bonds default, only the senior traunche is paid
  else if (numDefaultsT[i] > 2*mezzanine)
  {
    senior2 = senior - numDefaultsT[i]
  }
}

```



```

    seniorT = rbind(seniorT, senior2)
    mezzanineT = rbind(mezzanineT, 0)
    juniorT = rbind(juniorT, 0)
  }

  # Print what percentage has been completed
  if (i%%(0.1*N) == 0)
  {
    print(paste(100*(i/N), "% complete when using Clayton Copula."))
  }

  # Index counter
  i = i+1
}

# Calculate the returns probability for each tranche
probsSeniorT = table(seniorT)/N
probsMezzanineT = table(mezzanineT)/N
probsJuniorT = table(juniorT)/N

# Calculate the expected value for each tranche
expectedSeniorT = sum(probsSeniorT*as.integer(names(probsSeniorT)))
expectedMezzanineT = sum(probsMezzanineT*as.integer(names(probsMezzanineT)))
expectedJuniorT = sum(probsJuniorT*as.integer(names(probsJuniorT)))

# Print results when utilizing the t-copula
barplot(probsSeniorT, main="Probability of $(x) Returns to Senior Traunche
        using the t-Copula", xlab="Number Dollars Returned", ylab=
        "Probability of Receiving Said Amount")
print(probsSeniorT)
print(paste(expectedSeniorT, "= Estimate of Expected value for Senior Traunche (t-
        Copula)"))
barplot(probsMezzanineT, main="Probability of $(x) Returns to Mezzanine Traunche
        using the t-Copula", xlab="Number Dollars Returned", ylab=
        "Probability of Receiving Said Amount")
print(probsMezzanineT)
print(paste(expectedMezzanineT, "= Estimate of Expected value for Mezzanine
        Traunche (t-Copula)"))
barplot(probsJuniorT, main="Probability of $(x) Returns to Junior Traunche
        using the t-Copula", xlab="Number Dollars Returned", ylab=
        "Probability of Receiving Said Amount")
print(probsJuniorT)
print(paste(expectedJuniorT, "= Estimate of Expected value for Junior Traunche (t-
        Copula)"))
}
}
}

```