Credit Spread for a Basket Product (CR)

Avik Samanta

June 2020 cohort

Contents

Credit Spread for a Basket Product

Introduction

Basket credit derivatives are financial instruments whose pay-out depends on the credit behaviour of a set of reference credits. Unlike a plain vanilla Credit Default Swap (CDS) that provides protection against the default of a single name, a basket default swap provides protection against the k^{th} default on a basket of n reference names where n>=k. The most common of basket default swaps are the ones with k=1 and these are called first-to-default swaps where the protection seller compensates the buyer for the loss on the reference credit that defaults first. However, products with higher k's are possible. The protection buyer pays a premium (called the fair spread), which is akin to an insurance fee that is paid until the maturity or until the k^{th} default occurs.

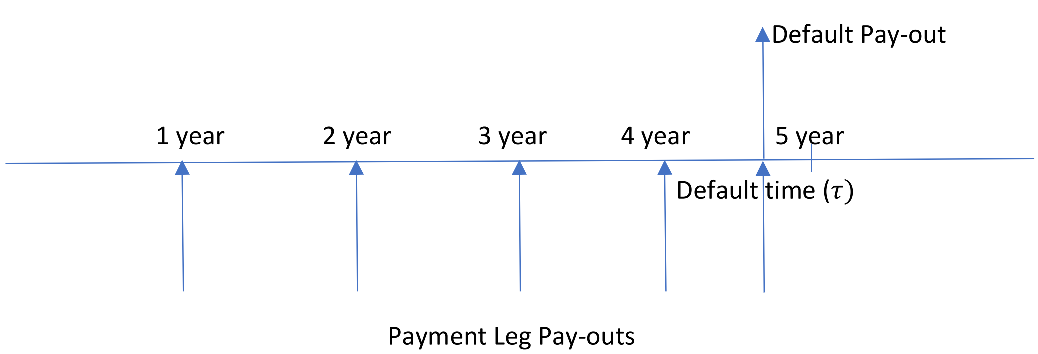
We will study the pricing of the fair spread of a basket default swap consisting of five reference names that are leading US technology/e-commerce companies.

Reference Names	**Stock Ticker**
Google	GOOG
Amazon	AMZN
Microsoft	MSFT
Apple	AAPL
Netflix	NFLX

The fair spreads for different K's are calculated separately as an expectation over the joint loss distribution. The relationships between the reference names are represented in the form of a correlation matrix derived based on the asset (stock) returns. As there are no closed form solutions for the joint distribution of the default times of the individual reference names, a non-parametric solution is obtained through Monte Carlo simulation by sampling from elliptical Copulas (both Gaussian and T Copula).

Pricing of a basket credit default swap

In a Credit Default Swap, the Default leg pays at the time of the default whereas the payment leg pays at a pre-defined frequency (typically yearly). The fair spread of a CDS is obtained by equating the pay-out from the Default Leg and the Payment Leg.



In the above example, for demonstration purpose, we have assumed that the default occurs between 4 to 5 years. The default pay-out would occur at the default time τ and the premium leg would accrue until the same time.

The default is modelled as a Poisson process, so the default time (τ) would be a function of the Hazard rate, which is the intensity of the Poisson process. However, in a basket credit default swap, we are not dealing with one hazard rate. We would have hazard processes for each of the names included in the basket. So, the pricing of the basket credit default swap would involve the modelling of the joint default times. We can model the default risk of the basket CDS if the following joint distribution function is known:

$$F(t_1, \ldots, t_n) = \Pr(\tau_1 \leq t_1, \ldots, \tau_n \leq t_n)$$

Where τ_i <= t_i signifies the default event on the i^{th} reference name.

The joint distribution for k-th to default time across all reference names $\tau_k \sim F_k(t_1, t_2, \ldots, t_n)$ has **no closed form.** However, we can isolate the dependence structure among multiple variables using Copula. Copula approach separates the joint distribution into two parts:

- The marginal distribution for each variable
- The dependence structure among the variables

Sklar Theorem proves that for every multivariate cumulative distribution function (CDF) F with marginals F_i there exists some Copula function C such that :

$$F(x_1, x_2, \ldots, x_n) \equiv C(u_1, u_2, \ldots, u_n)$$

And if the joint distribution is continuous, then the Copula function would be unique. Copula method is a way to structure Monte Carlo simulation. The approach is non-parametric which gives it the flexibility. The calculation of the basket spread would involve sampling from the Copula based on the below procedure (discussed in detailed in a subsequent section of the report)-

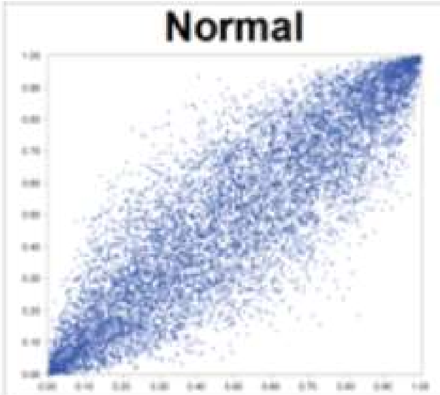
- we start with generating a correlated set (u_1, u_2, …, u_n), where the correlation is enforced through a linear correlation matrix. In our case, n or the number of names in the basket CDS is equal to 5.
- Then convert the vector of correlated uniform variables (u_1, u_2, …, u_n) into a vector of default times. The conversion involves comparison with the hazard rates and arriving at the exact default time. Hazard rate bootstrapping and calculation of the exact default time are discussed in the next sections.

Elliptical Copulas are the copulas of elliptically contoured distributions. The most common elliptical distributions are Normal (Gaussian) and Student-t distributions. The advantage of the elliptical copulas is that different level of correlations can be specified for the marginals i.e., it does not assume the same correlation for the whole of the distribution. However, the disadvantage is that it does not have a closed-form solution and so non-parametric solutions such as Monte Carlo simulation would have to be used.

The multivariate **Gaussian copula** can be expressed as:

$$C(u_1, u_2, \ldots, u_n) = \Phi_n(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_n); \Sigma)$$

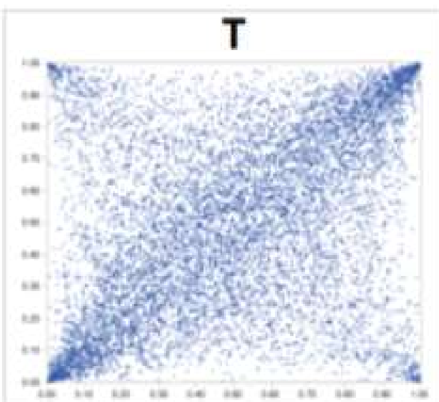
Where ϕ_n is the CDF for the multivariate normal distribution and Σ is the linear correlation matrix.



The multivariate **Student's t copula** can be expressed as:

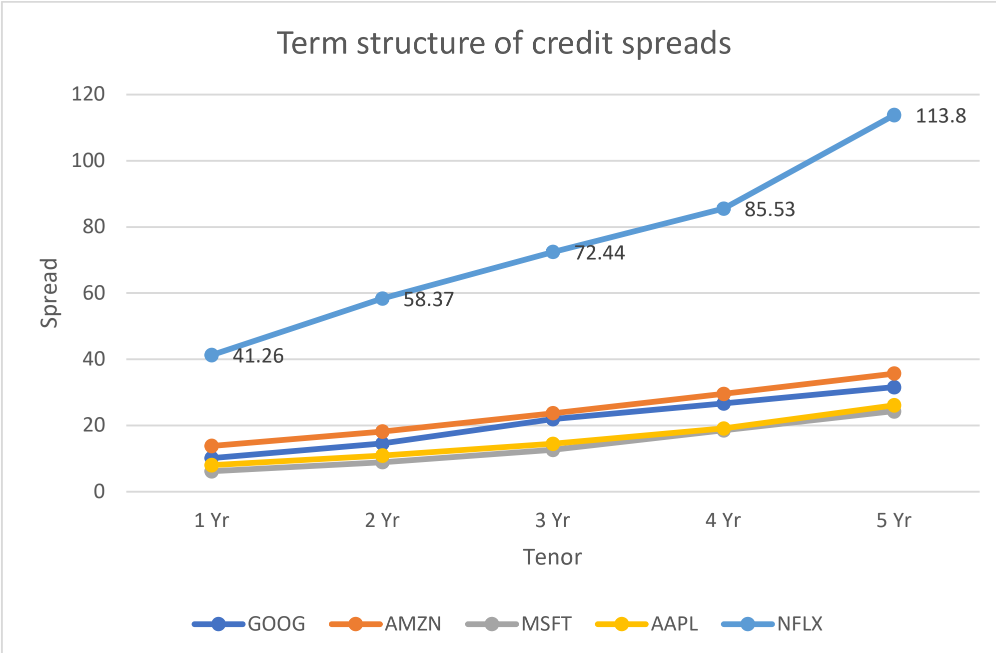
$$C(u_1, u_2, \ldots, u_n) = T_\nu(T_\nu^{-1}(u_1), \ldots, T_\nu^{-1}(u_n); \Sigma)$$

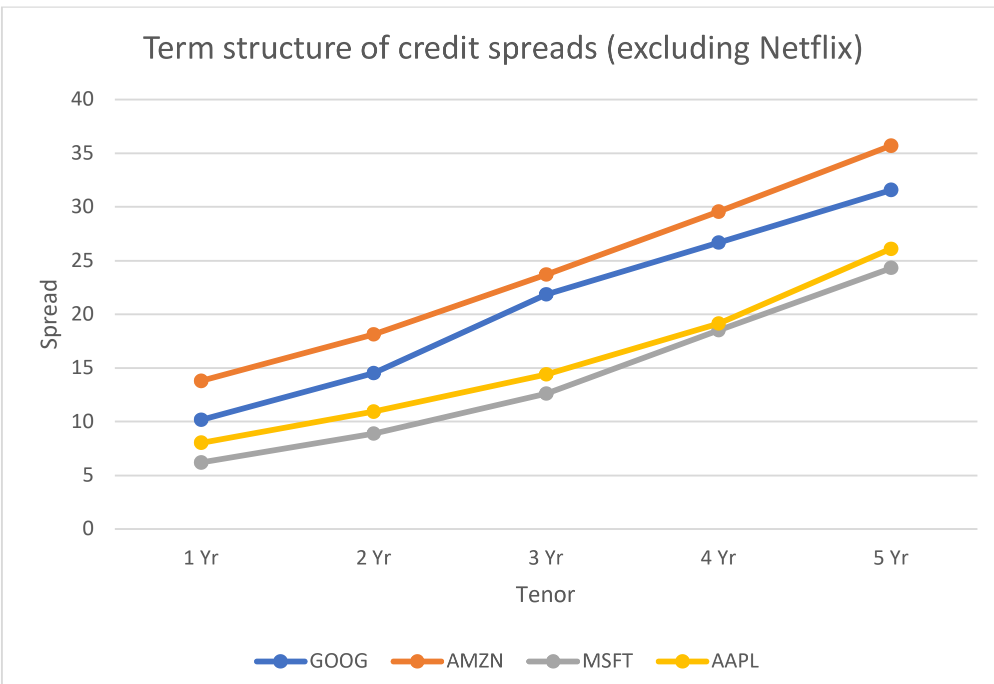
where T_ν is the CDF for the multivariate Student t distribution for ν degrees of freedom and Σ is the linear correlation matrix.



Bootstrapping hazard rates from the credit spreads

We have sourced the credit spreads of the reference names for tenors spanning from 1 year to 5 year for the reference date of 15 December 2020. We have bootstrapped the hazard rates from the credit spread term structure.





We can see that the term structure of the credit spread is upward sloping with the spreads corresponding to longer tenors are higher compared to spreads corresponding to the shorter tenors.

We can model the arrival of a credit default event as a Poisson process, which is characterised by the parameter λ (intensity of the process). In the context of a default process, λ is known as the "Hazard Rate". The default indicator function represents the arrival of a default event at a future time τ.

$$I_{Default}(t) = 1_{\{\tau > T\}} = \begin{cases} 1 & \text{if } \tau > T \\ 0 & \text{if } \tau \leq T \end{cases}$$

Survival indicator function represents the survival or the default not occurring until the future time τ.

$$I_{Survival}(t) = 1 - I_{Default}(t) \quad = \begin{cases} 1 & \text{if } \tau < T \\ 0 & \text{if } \tau \geq T \end{cases}$$

Given the nature of the default event, the default time can be represented as the first jump in a Poisson process-

- Poisson process has no memory
- The inter-arrival time of a Poisson process $(\tau_{n+1} - \tau_n)$ are exponentially distributed.
- Two or more jumps at exactly the same time have probability of zero

A homogeneous Poisson process has a constant intensity λ and the survival probability is represented as

$$P(t,T) = exp\left(-\lambda(T-t)\right)$$

A nonhomogeneous Poisson process relaxes the assumption of constant intensity and assumes the intensity to be a function of time

$$P(t,T) = exp\left(-\int_t^T \lambda(s)ds\right)$$

Assuming a piece-wise constant model, we can arrive at a hazard rate term structure to match the market spreads of the CDS of different tenors.

The following equation governs the pricing of the CDS spread (S_N) for a particular tenor N-

Equation 1 :

$$S_N = \frac{(1-R)\sum_{n=1}^{N} D(0,T_n)\left(P(T_{n-1}) - P(T_n)\right)}{\sum_{n=1}^{N} D(0,T_n)P(T_n)\left(\Delta t_n\right)}$$

Where, the numerator represents the price of the default leg and the S_N multiplied by the denominator of the above equation represents the price of the Premium leg of the CDs.

- R is the recovery rate
- D $(0,T_n)$ is the discounting rate corresponding to time T_n
- P (T_n) is the survival probability till time T_n
- P (T_{n-1}) is the survival probability till time T_{n-1}

- Δt_n is the time difference between T_n and T_{n-1}

The hazard rate term structure is bootstrapped starting with the CDS spread corresponding to the 1-year tenor. We also know that the obligor is solvent at time zero, so the following equation will hold-

$$P(T_0) = P(0) = 1$$

Assuming S_1 being the spread corresponding to the 1-tear tenor (T_1), from Equation 1 above, we get

$S_1 = (1\text{-}R)\ (1\text{-}\ P(T_1))\ /\ (P(T_1)^*\ \Delta t_1)$

Or,

$P(T_1) = (1\text{-}R)\ /\ ((1\text{-}R)+\ S_1^*\ \Delta t_1))$

As we are bootstrapping using yearly CDS prices, $\Delta tn = 1$. Loss rate is defined as L= (1-R).

So, we get,

$P(T_1) = L\ /\ (L+S_1)$ …. **Equation 2**

As the CDS spreads are known (they are part of the input data), the survival probability P(T1) can be computed based on a given Loss Rate. For hazard curve bootstrapping, we have assumed a Loss Rate of 60%, which is same as the Recovery Rate of 40%.

From Equation 1, assuming Credit Spread for 2nd-year tenor is S_2, we can represent the survival probability corresponding to year 2 as follows-

Equation 3

$$P(T_2) = \frac{D(0,T_1)\,[L(1) - (L + \Delta t_1 S_2)P(T_1)]}{D(0,T_2)(L + \Delta t_2 S_2)} + \frac{P(T_1)L}{L + \Delta t_2 S_2}$$

More generally, for a particular N (i.e., the N-th tenor), the survival probability would follow the equation below

Equation 4

$$P(T_N) = \frac{\overbrace{\sum_{n=1}^{N-1} D(0,T_n)\left[LP(T_{n-1}) - (L + \Delta t_n S_N)P(T_n)\right]}^{\text{Summative Term}}}{\underbrace{D(0,T_N)(L + \Delta t_n S_N)}_{\text{Quotient}}} + \underbrace{\frac{P(T_{N-1})L}{(L + \Delta t_N S_N)}}_{\text{Last Term}}.$$

Once P(T1) is calculated, P(T2) can be calculated algebraically and so on. We have derived the discounting rates from the US Treasury rates for different tenors as of the reference date. We have fit a quadratic equation to determine the rate for any intermediate points and derive the discounting factors from these rates.

Once the survival probability corresponding to a particular tenor Tn is calculated, the hazard rate corresponding to the tenor can be calculated as below-

$P(T_n)= \exp(-\lambda_{T_n,T_{n-1}} * (T_n - T_{n-1}))$

As we are bootstrapping using yearly CDS prices, (Tn - Tn-1) would be equal to 1.

Denoting the hazard rates for the period 0 to 1 year as λ_1, period 1 to 2 year as λ_2, and so on-

$P(T_1)=\exp(-\lambda_1)$

$P(T_2)=\exp(-(\lambda_1+ \lambda_2)= P(T_1)* \exp(-\lambda_2)$ ....

$P(T_n)=\exp(-(\lambda_1+ \lambda_2+...+ \lambda_n)= P(T_{n-1})* \exp(-\lambda_n)$

From the above, we can calculate the hazard spread as

$\lambda_1 = - \ln(P(T_1))$

$\lambda_2 = - \ln(P(T_2) / P(T_1))$ ....

$\lambda_n= - \ln(P(T_n) / P(T_{n-1}))$

Below we show one such examples where the hazard rates for the reference name GOOG (Google Inc.) is calculated based on the yearly CDS quotes for Google.

Recovery Rate 40%

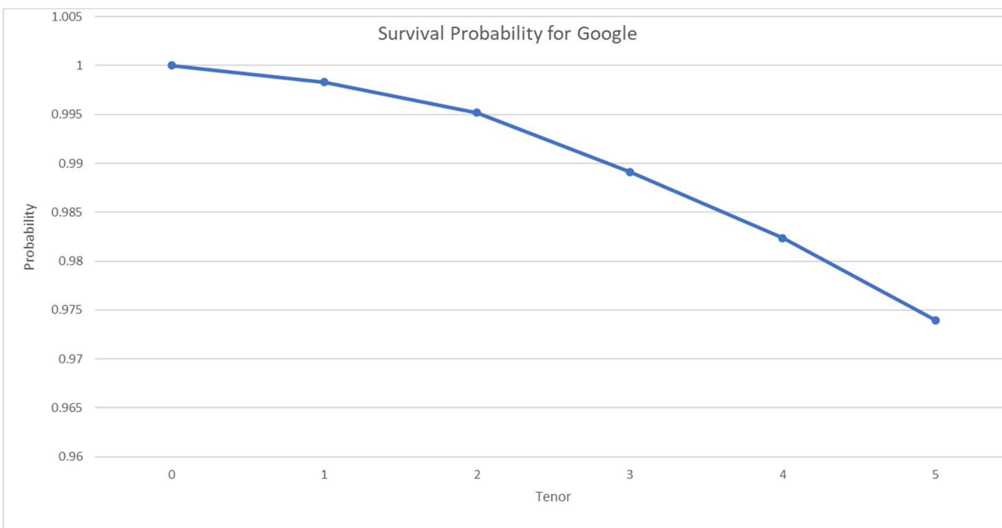
TIME (Years)	dt	SPREAD	DF	Survivavl Prob	first term	second term	third term	fourth term	Summative Term	Quotient	First Term (Summative Term/Quotient)	last term	Hazard Rate
0				100.00%									
1	1	10.18	0.9988	99.83%									0.17%
2	1	14.52	0.9974	99.52%	-0.0004				-0.0004	0.5999	-0.0007	0.9959	0.31%
3	1	21.86	0.9952	98.91%	-0.0012	-0.0003			-0.0015	0.5993	-0.0024	0.9916	0.61%
4	1	26.67	0.9912	98.24%	-0.0016	-0.0008	0.0010		-0.0014	0.5974	-0.0024	0.9847	0.69%
5	1	31.58	0.9861	97.40%	-0.0021	-0.0013	0.0005	0.0010	-0.0019	0.5948	-0.0033	0.9772	0.86%

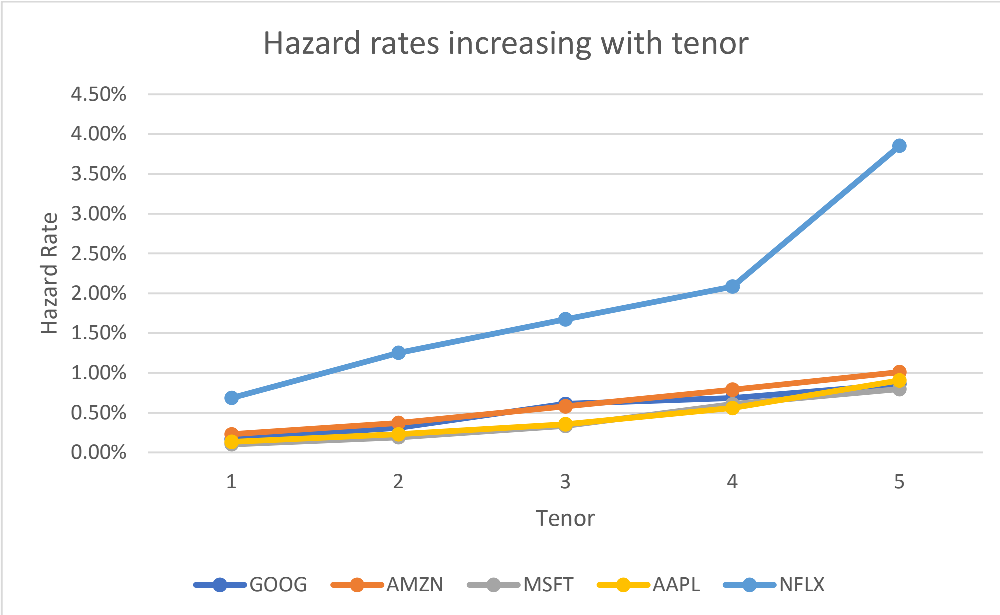
Once we have derived the hazard rates for the other reference names as well, we get a 5X5 matrix of hazard rates. We also calculate the hurdle rates for each tenor, which is calculated as the summation of the hazard rates up to that tenor i.e., hurdle rate for year 1 would be λ_1, hurdle rate for year 2 would be $\lambda_1 + \lambda_2$ and so on. The hurdle rates would be used later while sampling from Gaussian/Student t Copula in determining the exact default time. A value between two consecutive hurdle rates would signify a default occurring in that period.

Table 1

	λ_1	λ_2	λ_3	λ_4	λ_5	λ_1	$\lambda_1+\lambda_2$	$\lambda_1+\lambda_2+\lambda_3$	$\lambda_1+\lambda_2+\lambda_3+\lambda_4$	$\lambda_1+\lambda_2+\lambda_3+\lambda_4+\lambda_5$
GOOG	0.17%	0.31%	0.61%	0.69%	0.86%	0.17%	0.48%	1.09%	1.78%	2.64%
AMZN	0.23%	0.37%	0.58%	0.79%	1.01%	0.23%	0.60%	1.18%	1.97%	2.99%
MSFT	0.10%	0.19%	0.33%	0.61%	0.80%	0.10%	0.30%	0.63%	1.24%	2.03%
AAPL	0.13%	0.23%	0.36%	0.56%	0.91%	0.13%	0.36%	0.72%	1.28%	2.18%
NFLX	0.69%	1.25%	1.67%	2.09%	3.85%	0.69%	1.94%	3.61%	5.70%	9.55%

If we plot the survival probabilities against the tenor, we get a downward sloping curve as the chance of survival or the default not occurring decreases with time. As T increases to a very large value, the survival probability would be close to zero as over a very long period, the reference credit is expected to default. On the other hand, based on the credit spreads, the hazard rates i.e., the default intensity increases with time signifying that the uncertainty or the risk factors increases with time and market factors in the higher uncertainty in the CDS spreads.





Calculation of Default Time

From the hazard rates, the exact future time a default occurs i.e., the "default time" can be estimated. As discussed earlier in the report, Poisson process can be used to depict the probability distribution of the default event to occur.

Denoting the default time as τ, the Poisson Cumulative Distribution function (CDF) would denote the survival probability till the default time τ. Therefore,

Survival probability, $P(0, \tau)= \int_0^\tau e^{-\lambda t}\, dt = -e^{-\lambda \tau}+1$ ....**Equation 5**

From the above, $\tau = -\ln(1- P(0, \tau))/\lambda$

As we have seen before, λ is not constant over the period 0 to τ. Rather, we have assumed it to be piece-wise constant over different tenors. It is not possible to directly determine τ based on a given survival probability as both τ and λ (which is dependent on τ) are unknown. However, it is still possible to estimate the year during which the default occurs and the exact default time can then be either calculated or suitably estimated.

Let's assume that the default occurs during the period t_m and t_{m-1}. If we consider the survival probability, the following equation would hold-

$$P(0, t_m) \leq P(0, \tau) \leq P(0, t_{m-1})$$

We have seen before that the survival probabilities decreases over time and we have also assumed λs to be piece-wise constant over the various tenors. Representing equation 5 above in discretised form (note that we have used the symbols log and ln interchangeably),

$$\log P(0, t_m) = -\sum_{j=1}^{m} \lambda_j \Delta t_j.$$

Assuming P(0, τ) to be the survival probability up to the default time τ and given the fact that yearly time intervals have been used for bootstrapping the hazard rates, we have the following-

$-(\lambda_1 + \lambda_2 + \ldots + \lambda_{m-1}) >= \log(P(0, \tau)) >= -(\lambda_1 + \lambda_2 + \ldots + \lambda_m)$

In absolute terms,

$(\lambda_1 + \lambda_2 + \ldots + \lambda_{m-1}) <= abs(\log(P(0, \tau))) <= (\lambda_1 + \lambda_2 + \ldots + \lambda_m)$

So, once we have a survival probability, we can compare that against the threshold values of λs to determine the period during which the default occurs. This will be useful when we perform the sampling from the Gaussian or Student t copula. Suppose a simulation gives us correlated (u_1, u_2, …., u_5), our task would be to convert $u_i \rightarrow \tau_I$ for each of the reference names.

The exponential CDF is u = 1 - $e^{-\lambda\tau}$, so log (1-u)= $\lambda_\tau\ \tau$.

We iterate by adding up the hazard rates and checking for the tenor during which the above in equality holds i.e., finding $t_{m-1} <= \tau <= t_m$ such that

$(\lambda_1 + \lambda_2 + \ldots + \lambda_{m-1}) <= abs(\log(1-u)) <= (\lambda_1 + \lambda_2 + \ldots + \lambda_m)$

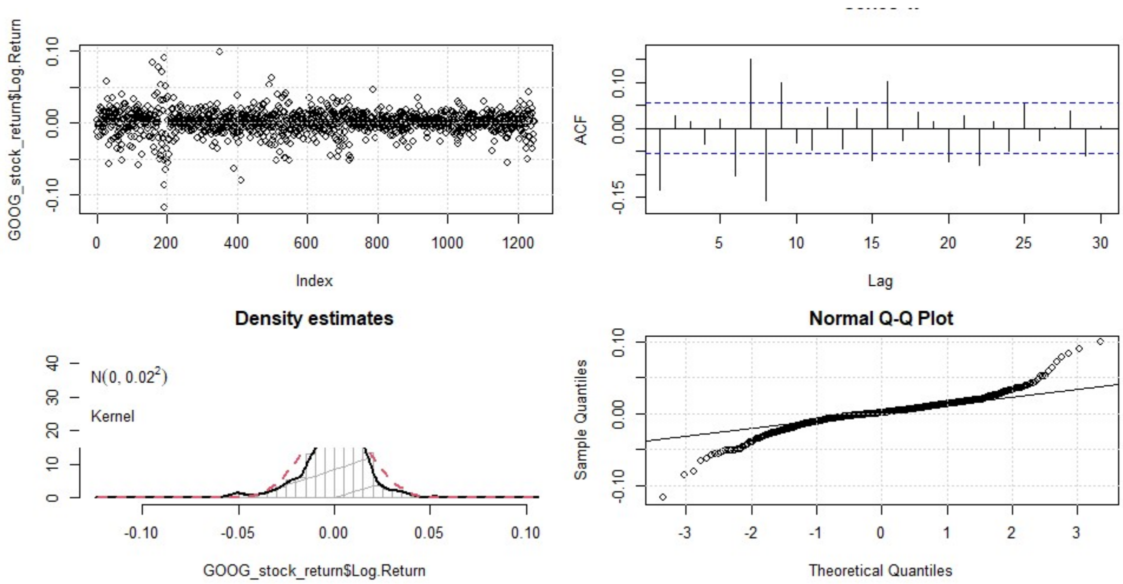
So, once we have the correlated set of u_i , by taking the absolute values of log of (1-u_i) and comparing them against the λ thresholds corresponding to the reference names (refer to Table 1 on page 10), we would be able to determine during which year a default may occur for a particular reference name for that specific simulated instance. The calculation of the exact default time is addressed in the section "sampling from the Gaussian copula".

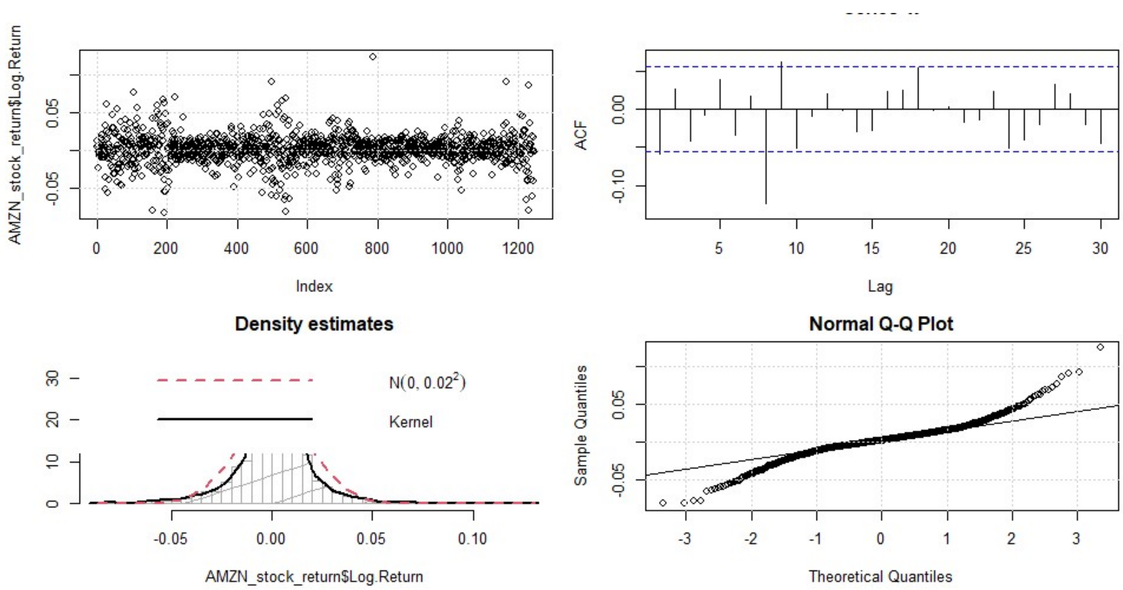
Correlation Calculation

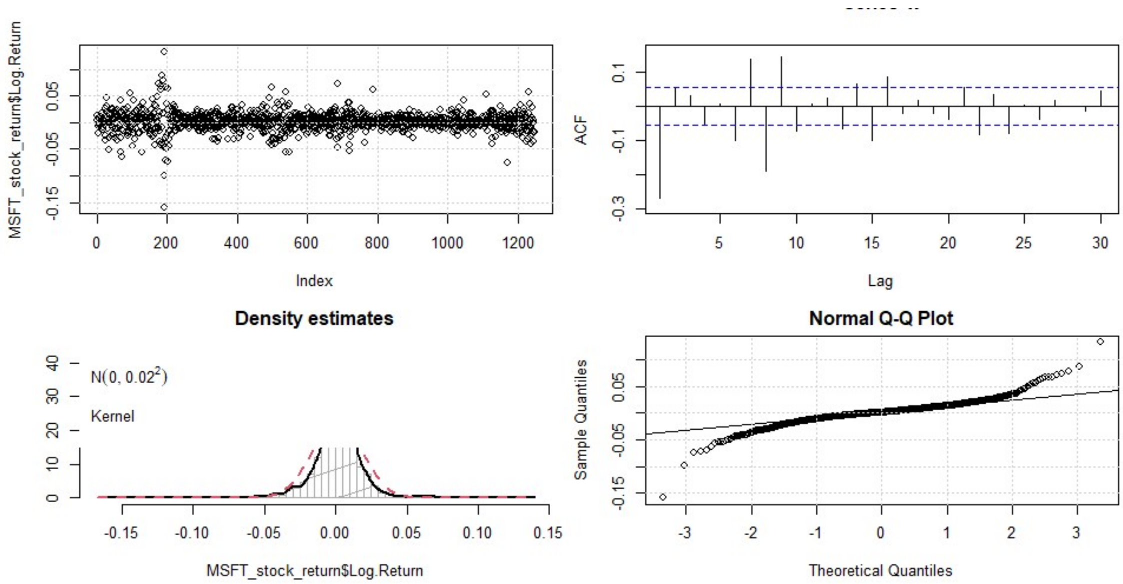
The strength and direction of the relationship between two variables are represented through correlation. In the case of basket credit derivative, we have five variables representing the default behaviour of each of the reference names. This relationship is expressed in the form of a 5X5 correlation matrix.

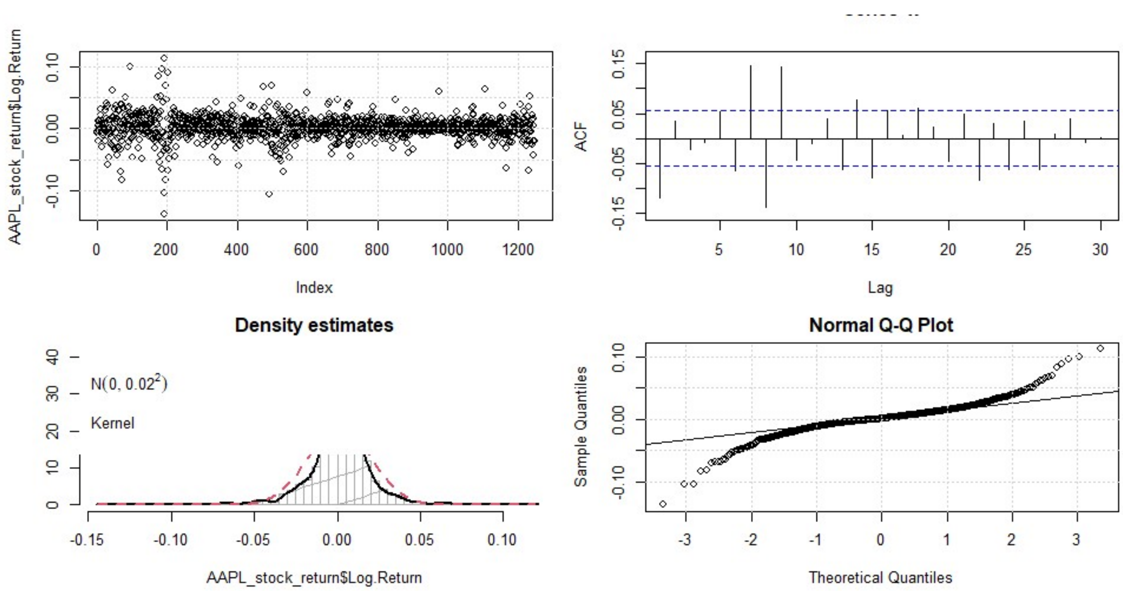
Both Gaussian and Student t copula require linear correlation as input. Pearson's correlation coefficient is derived using the linear regression method. By fitting a linear regression model between two variables and determining the goodness of fit (R^2), Pearson's correlation co-efficient can be calculated as R, which is the square root of R^2. Although the calculation is quite straightforward, it assumes the data is normally distributed.

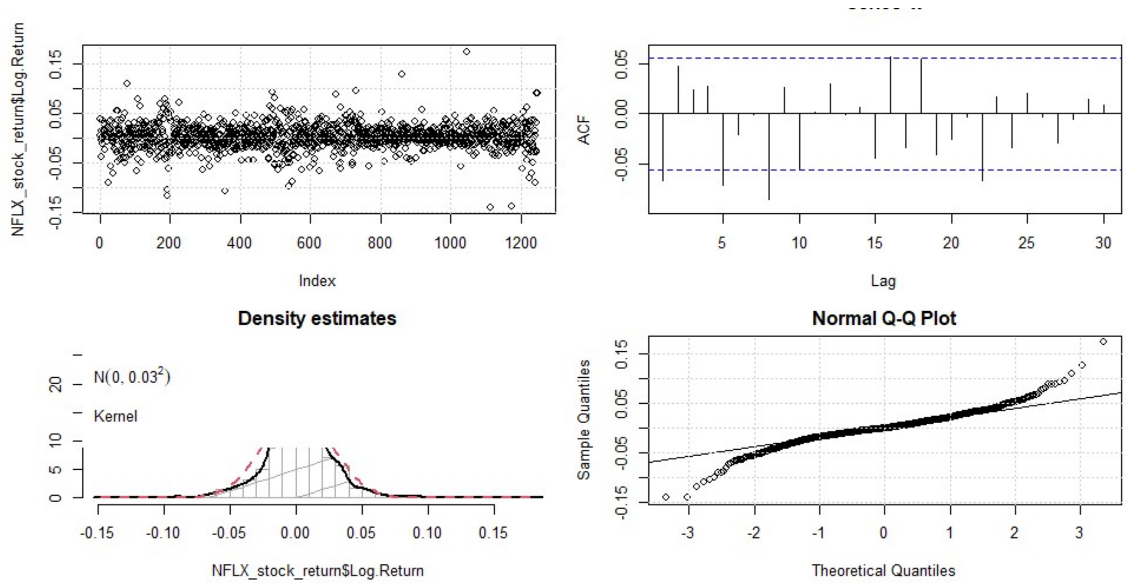
We have used the historical stock prices to determine the correlation. As it was difficult to source historical CDS prices of a sufficient length, we have used the log return of the stock prices as a suitable alternative. The stock price data for last 5 years was easily available only (data source- Yahoo Finance). However, when we looked at the log returns, we could see that they were not normally distributed.


GOOG_stock_return$Log.Return
Index
ACF
Lag
Density estimates
N(0, 0.02²)
Kernel
GOOG_stock_return$Log.Return
Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles


AMZN_stock_return$Log.Return
Index
ACF
Lag
Density estimates
N(0, 0.02²)
Kernel
AMZN_stock_return$Log.Return
Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles


MSFT_stock_return$Log.Return
Index
ACF
Lag
Density estimates
N(0, 0.02²)
Kernel
MSFT_stock_return$Log.Return
Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles


AAPL_stock_return$Log.Return
Index
ACF
Lag
Density estimates
N(0, 0.02²)
Kernel
AAPL_stock_return$Log.Return
Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles



The Shapiro test also shows that the data is not normal.

```
> ##Shapiro Wilk Test
> shapiro.test(GOOG_stock_return$Log.Return)

        Shapiro-Wilk normality test

data:  GOOG_stock_return$Log.Return
W = 0.9115, p-value < 2.2e-16

> shapiro.test(AMZN_stock_return$Log.Return)

        Shapiro-Wilk normality test

data:  AMZN_stock_return$Log.Return
W = 0.93314, p-value < 2.2e-16

> shapiro.test(MSFT_stock_return$Log.Return)

        Shapiro-Wilk normality test

data:  MSFT_stock_return$Log.Return
W = 0.88802, p-value < 2.2e-16

> shapiro.test(AAPL_stock_return$Log.Return)

        Shapiro-Wilk normality test

data:  AAPL_stock_return$Log.Return
W = 0.90279, p-value < 2.2e-16

> shapiro.test(NFLX_stock_return$Log.Return)

        Shapiro-Wilk normality test

data:  NFLX_stock_return$Log.Return
W = 0.9412, p-value < 2.2e-16
```

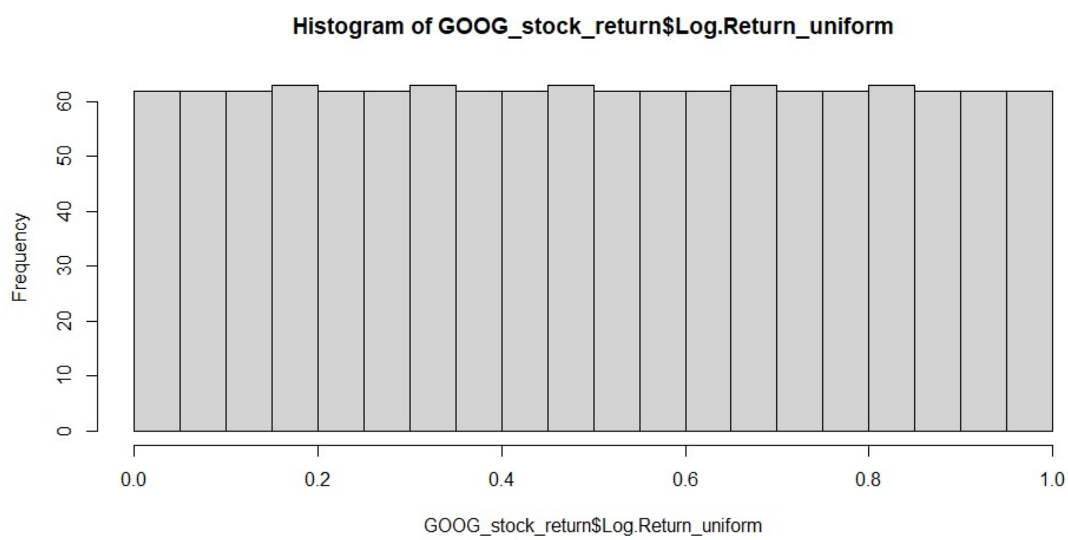
Scaling or normalisation of the data would not necessarily work as the resultant distribution may not be sufficiently uniform. We have followed the below method in estimating the Pearson correlation from the historical stock prices.

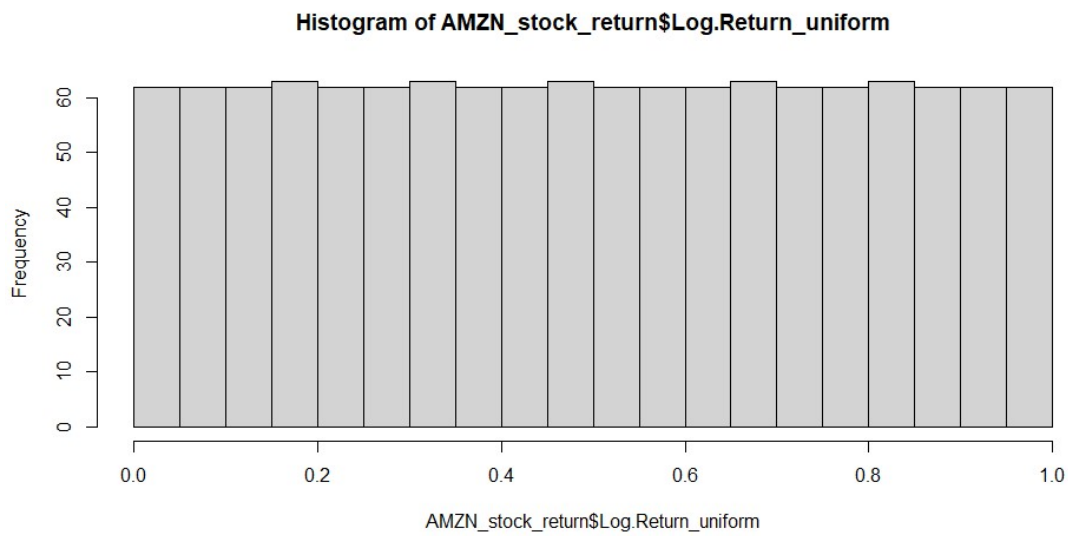
Step 1: Convert the asset returns (X) into uniform distribution U using an empirical cumulative distribution function $F\hat{}(X)$. U= $F\hat{}(X)$

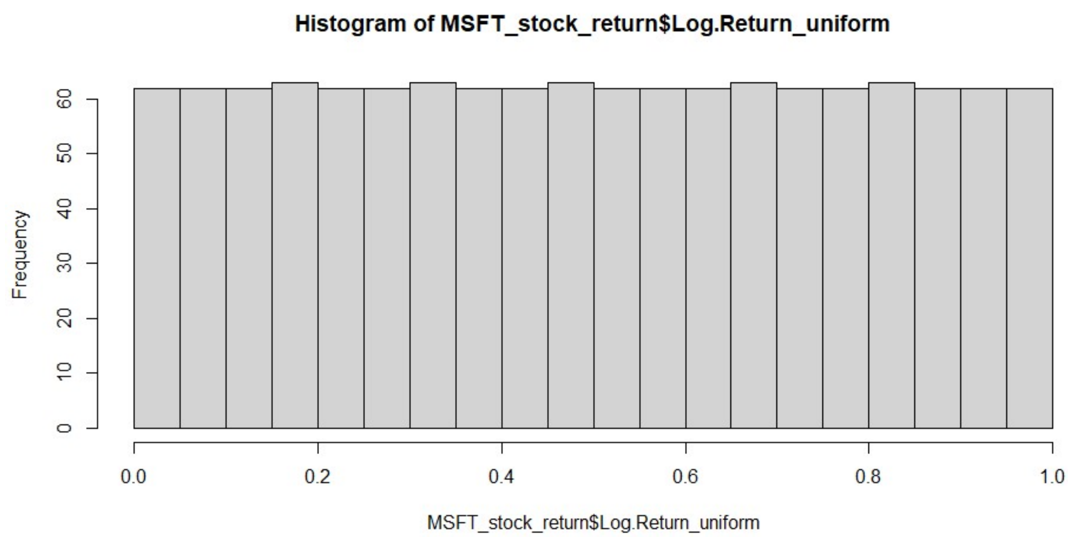
Step 2: Apply inverse normal CDF on U and obtain Z= $\emptyset\hat{}(-1)\ (X)$

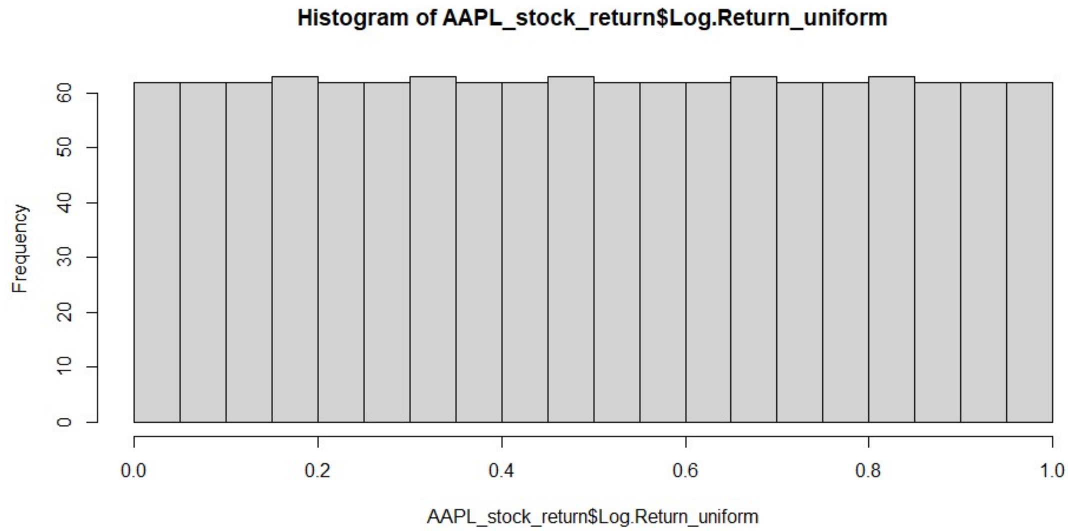
Step 3: Derive the Pearson correlation matrix from normally distributed Z values for the five reference names

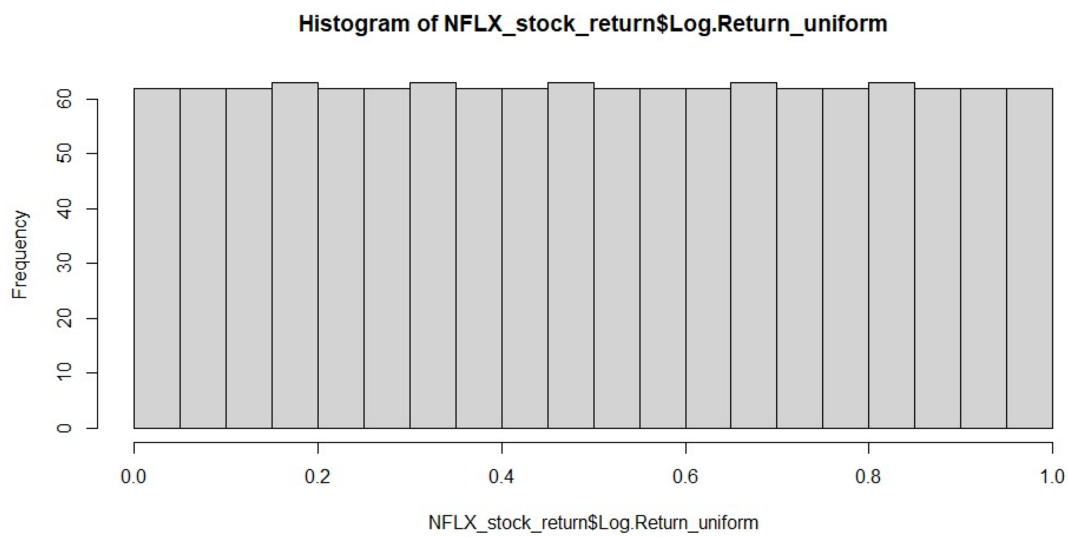
We have used the ecdf() function from R base package and the function returns an empirical cumulative distribution step function. When we look at the uniform distribution returned by the ecdf(), we found them sufficiently uniform as shown below.




Histogram of AMZN_stock_return$Log.Return_uniform
Frequency
0
10
20
30
40
50
60
0.0
0.2
0.4
0.6
0.8
1.0
AMZN_stock_return$Log.Return_uniform


Histogram of MSFT_stock_return$Log.Return_uniform
Frequency
0
10
20
30
40
50
60
0.0
0.2
0.4
0.6
0.8
1.0
MSFT_stock_return$Log.Return_uniform

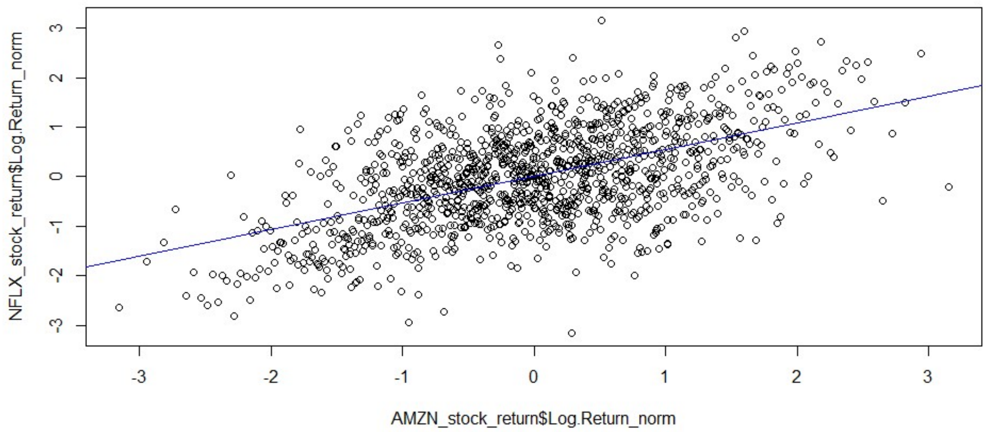


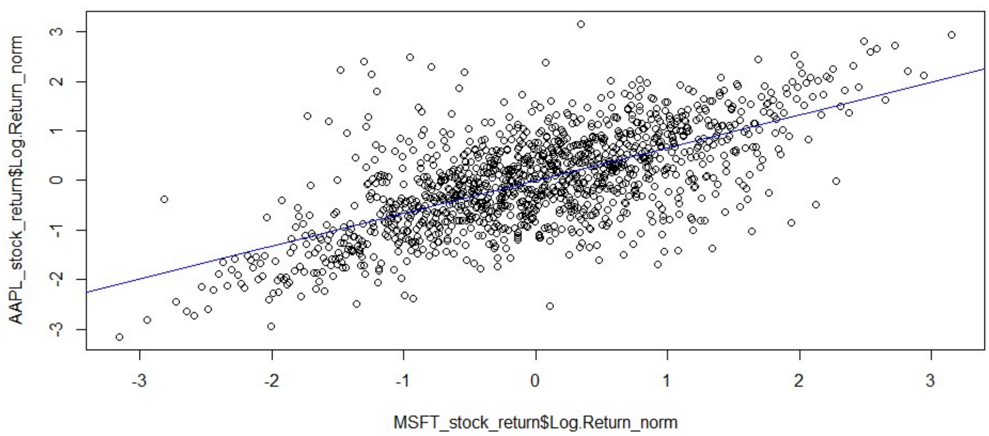


Pearson Correlation

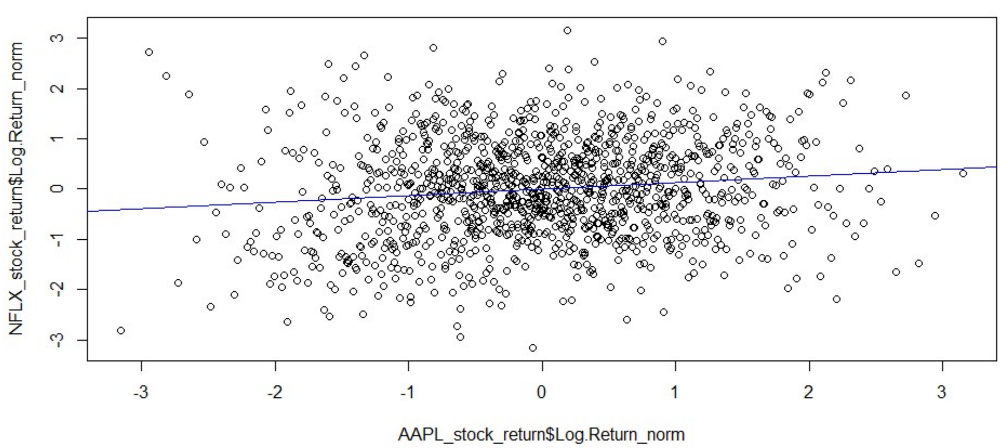
Next, we get the normal Z from the pseudo-samples U for all the reference names and use the "cor" function in the R base package with the method "Pearson" to arrive at the Pearson Correlation matrix. The matrix indicates higher correlation among Netflix and Amazon as compared to the correlation of Netflix with the other reference names. This is not surprising given both Amazon and Netflix are primarily e-commerce players. On the other hand, tech giants Google, Microsoft, and Apple show higher correlation with each other.

Scatter Plot indicating high correlation





Scatter plot indicating low correlation



Pearson Correlation Matrix	GOOG	AMZN	MSFT	AAPL	NFLX
GOOG	1	0.4370042	0.5878259	0.4872916	0.2483669
AMZN	0.4370042	1	0.3200410	0.2709753	0.5373229
MSFT	0.5878259	0.3200410	1	0.6603984	0.1507496
AAPL	0.4872916	0.2709753	0.6603984	1	0.1320469
NFLX	0.2483669	0.5373229	0.1507496	0.1320469	1

Rank Correlation

Whereas Pearson's correlation coefficient is useful for correlating normally distributed data, a rank correlation method such as "Kendall Tau" is not restricted by the normality assumption. Kendall Tau correlation represents the strength of dependence between two variables and is calculated based on the concordant-discordant method. It is scale invariant. However, the Kendall Tau measure requires linearisation by applying the following: $\rho = \sin(\frac{\pi}{2}\rho_K)$

The log returns of stock prices have been used as input for the rank correlation calculation. We have used the "cor" function in the R base package with the method "kendal" to arrive at the Kendall Tau Correlation matrix.

The linearised "Kendall Tau" correlation matrix, obtained as above and as shown below, has been used as an input while sampling from the Student T copula. As the Student T copula is very sensitive to correlation, it is imperative that we use a scale-invariant correlation measure that is not constrained by the shape of the underlying distributions.

Kendall Tau Correlation Matrix	GOOG	AMZN	MSFT	AAPL	NFLX
GOOG	1	0.473643	0.616994	0.522641	0.234526
AMZN	0.473643	1	0.357154	0.294933	0.509496
MSFT	0.616994	0.357154	1	0.656408	0.150757
AAPL	0.522641	0.294933	0.656408	1	0.12908
NFLX	0.234526	0.509496	0.150757	0.12908	1

Rank Correlation and Degrees of Freedom for Student T Copula

In the Copula method, we are describing the dependence structure of multivariate distributions in terms of some simple mathematical construction that is useful at describing the tail-dependence among variables. Multivariate Student T distributions are a family of distributions, which has a particular degrees of freedom parameter and also a correlation parameter that between them describe the distribution and hence, the dependence structure among the variates in the corresponding copula (Student T Copula). Therefore, to correctly represent the Copula, we would need to calibrate both the correlation matrix and the degrees of freedom.

We have used Maximum Likelihood Estimation (MLE) method to estimate the degree of freedom. For the Student's t copula, the density is represented by the following formula-

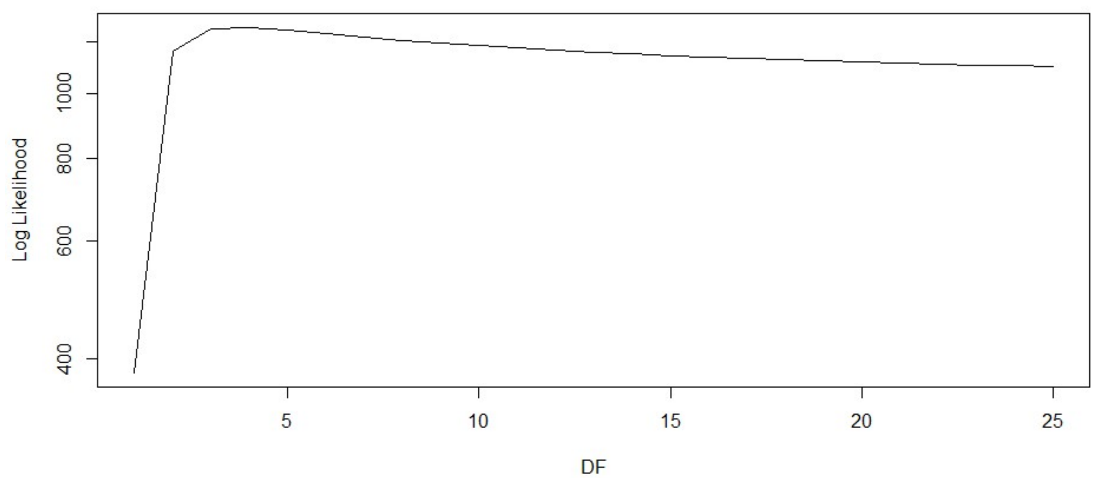
$$c(\mathbf{u};\nu,\widehat{\boldsymbol{\Sigma}}) = \frac{1}{\sqrt{|\Sigma|}}\frac{\Gamma\left(\frac{v+n}{2}\right)}{\Gamma\left(\frac{v}{2}\right)}\left(\frac{\Gamma\left(\frac{v}{2}\right)}{\Gamma\left(\frac{v+1}{2}\right)}\right)^{n}\frac{\left(1+\frac{T_v^{-1}(\boldsymbol{u}')\boldsymbol{\Sigma}^{-1}T_v^{-1}(\boldsymbol{u})}{v}\right)^{-\frac{v+n}{2}}}{\prod_{i=1}^{n}\left(1+\frac{T_v^{-1}(u_i)^2}{v}\right)^{-\frac{v+1}{2}}}$$

and we would try to maximise the following-

$$\underset{\nu}{\operatorname{argmax}}\left\{\sum_{t=1}^{T}\log c(\mathbf{u}_t^{Hist};\nu,\widehat{\boldsymbol{\Sigma}})\right\}$$

Where, u_t is a row vector representing the historical log return of stock prices for the five reference names, n is the number of observations (we have used last 5 years data up to 15 December 2020 as observations) and $\hat{\Sigma}$ is the correlation used in the sampling from Student T copula.

We calculate the log likelihood for different degrees of freedom (ν) and choose the particular ν that gives us the maximum log likelihood value. The graph of the log likelihood against the degrees of freedom (ν) is shown below:

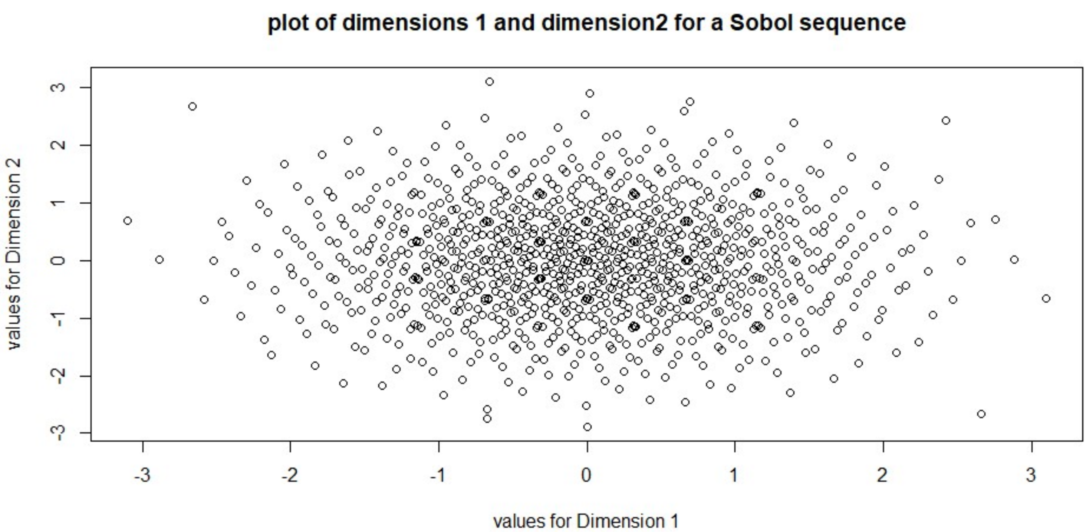


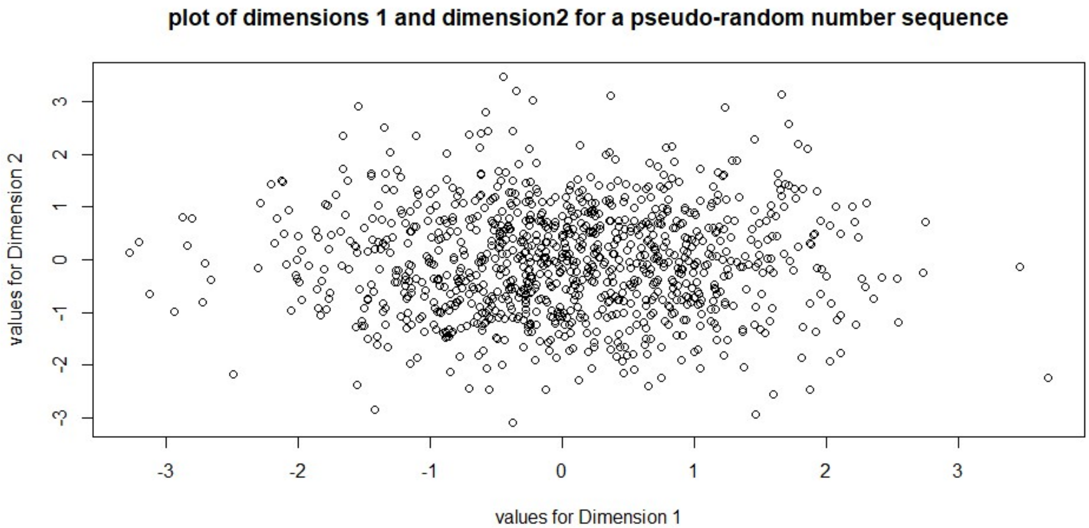
After checking the log likelihood values, the optimum degrees of freedom (ν) is 4

Using Sobol sequence to simulate the Standard Normal Variate for sampling from the Copula

A Sobol sequence is a quasi-random number series that evenly fills the probability space. As a result, Sobol sequence leads to less clustering as compared to using pseud-random number sequences such as the base rnorm() implementation in R. Use of the Sobol sequence in the Monte Carlo simulation speeds up convergence i.e., the solution for the spread of the basket credit derivatives converges to the fair value in a much smaller number of iterations as compared to the case where the pseudo-random numbers from the R base package is used. We will discuss this in more details in the "Convergence" section of the report.

In our implementation, we have used the "sobol" function from the R package "randtoolbox" to generate the Sobol sequences. We have built a customised function around it to generate data corresponding to 5 dimensions (corresponding to the five names in the basket) and up to the desired numbers of iterations to be used in MC simulation. Below we compare the “standard normal” samples drawn using the “Sobol Sequence” with the sample values drawn using the pseudo-random number generator in R for a sample size of 1000 and for two dimensions (i.e., comparing the samples drawn for two different reference names) . The graphs show that the Sobol sequence generate numbers which are more uniformly spread across the probability space of a standard normal variate as compared to the pseudo-random sequence which tends to show some clustering, which would delay convergence and therefore require a larger number of iterations.





Sampling from the Gaussian Copula

We have followed the steps below to perform the sampling from the Gaussian copula

- We have computed the Cholesky decomposition of the Pearson correlation matrix. We have checked that the Pearson Correlation matrix is positive define and the Cholesky decomposition yields a lower triangular matrix (A) and its conjugate transpose (A'). This is represented as Σ_{Gaussian} = AA'
- For each simulation, we generate a 5-dimensional vector (corresponding to the 5 names in the basket CDS) Z of n=5 uncorrelated random normal variables Z = (Z_1, … , Z_n)'. The sample is drawn from the Sobol sequence we have generated (refer to the previous section)
- Next, we compute a vector of correlated variables by computing X = AZ i.e., by imposing the correlation matrix A on the vector Z
- We use the Normal CDF to transform the vector X into a variable with a Uniform distribution (U) i.e. U = ϕ (X)
- We convert each uniform variable to default time ($u_i \rightarrow \tau_i$) using the hazard rate term structure of each name. As we have seen in the "Default Times" section of the report, by comparing abs (log(1- u_i)) with the cumulative hazard rates ($\sum_{k=1}^{T} \lambda_k$), we can determine the tenors during which the default occurs (here T=5 as we have used hazard rate term structure of 1 to 5 years). Once we know this, the exact default time can be determined as the midpoint of the tenor. For example, if we find that default occurs between t_{m-1} and t_m, the exact default time would be t_{m-1}+0.5. Here we have assumed that the default event is determined and settled based on accruals. The computation is performed for each of the reference names i.e., for i=1 to n (in our case, n=5 as we have five reference names in the basket)
- Once the default time is determined, the same is fed into the spread calculation to determine the value of the default leg as well as the premium leg. We would discuss this in the "Spread Computation" part of the report.

Sampling from Student T Copula

We have followed the steps below to perform the sampling from the Student t copula

- We have computed the Cholesky decomposition of the Kendall Tau correlation matrix. We have checked that the Kendall Tau Correlation matrix is positive define and the Cholesky decomposition yields a lower triangular matrix (A) and its conjugate transpose (A'). This is represented as Σ_{Studentt} = AA'
- For each simulation, we generate a 5-dimensional vector (corresponding to the 5 names in the basket CDS) Z of n=5 uncorrelated random normal variables. Z = (Z_1, ... , Z_n)'. The sample is drawn from the Sobol sequence we have generated
- Draw an independent Chi-squared random variable **s** ~ χ^2_ν
- We have earlier computed degrees of freedom as 4. So, the Chi-squared random variable would be determined with degrees of freedom of 4. We have used the "rchisq" function of the R base package to generate **s**. The Chi-squared random variable is computed by drawing ν squared random normal variables separately. So, in our case, s = $z_1^2 + \cdots + z_4^2$
- We compute a n-dimensional Student's t vector by calculating Y= Z / $\sqrt{s}/\nu$
- Next, we compute a vector of correlated variables by computing X = AY i.e., by imposing the correlation matrix A on the vector Y
- We use the Student t CDF to transform the vector X into a variable with a Uniform distribution (U) i.e. U = ϕ (X)
- We convert each correlated uniform variable to default time ($u_i \rightarrow \tau_i$) using the hazard rate term structure of each name.
- Once the default time is determined, the same is fed into the spread calculation to determine the value of the default leg as well as the premium leg.

Fair Spread Calculation

In the simulation of correlated default events, the marginal distributions τ_I ~ Exp ($\hat{\lambda}$) are kept separate from the dependence structure represented by the linear correlation matrix $\hat{\Sigma}$. The joint distribution of k-th default time across the n different reference names is given by τ_k ~ F_k (t_1, t_2, ... , t_n) ≡ C (u_1, u_2, ..., u_n), where C is the factorised Copula (Cholesky linear system). This joint distribution remains unknown and is likely different for each k-th to default instrument.

Par-spread of the k-th to default swap is calculated by equating the default leg with the premium leg under the risk neutral measure.

$$s = \frac{< DL >}{< PL_{\$} >} = \frac{(1-R)\sum_{i=1}^{m} Z(0,t_i)\ (F_k(t_i) - F_k(t_{i-1}))}{\Delta t \sum_{i=1}^{m} Z(0,t_i)\ (1 - F_k(t_i))}$$

Where m is the number of periods or tenors (in our case it is 5).

To simplify the calculation of the fair spread, the concept of Total Expected Loss, which is the expectation over the joint distribution, is used. With $\mathbb{E}[F_k(t)] = L_k$, the equation above can be represented as

$$\mathbb{E}[s] = \frac{(1-R)\sum_{i=1}^{m} Z(0,t_i)\,(L_i - L_{i-1})}{\Delta t \sum_{i=1}^{m} Z(0,t_i)\,(NP - L_i)}$$

The fair spread calculation is performed multiple times and averaged at the end across all iterations (Monte Carlo method). In the above equation,

- R is the recovery rate
- Z (t, T) is the discount factor obtained from the US treasury rate as of 15 December 2020.
- We assume NP or the notional principal is 1 and also assume 1/5th of the notional is invested in each name

We assume loss function per time period to be $L_i - L_{i-1}$ = 1/5 X NP = 1/5 to simplify the computation. So, based on the above, for each of the simulation, the price or the fair spread of the first-to-default swap would be

$$\frac{(1-R)Z(0,\tau_1)\times\frac{1}{5}}{Z(0,\tau_1)\,\tau_1\times\frac{5}{5}}$$

, where τ_1 is the default time for the first default.

To achieve convergence quicker, instead of calculating the spread for each iteration, we calculate the average of the numerator (default leg) and the denominator across all iterations and then calculate the final spread as the average of the default leg (numerator) divided by the average of the premium leg (denominator).

Where $\tau_1 > 5$, the default leg would have the value of 0, however, the premium leg would still accrue. We have assumed annual payment of premium, so the discretised discounted premium would be

DP_5 = 1 X Z (0,1) + 1 X Z (0,2) + … + 1 X Z (0,5)

Across all the iterations of the Monte Carlo simulation, the calculation would look like below-

Iteration	**1**	**2**	**3**	**4**	**…**	**100000**
Default Leg Payout	(1-R) X 1/5 X Z (0, τ_1)	0	0	(1-R) X 1/5 X Z (0, τ_1)	…	0
Premium Leg Payout	τ_1 X Z (0, τ_1)	DP_5	DP_5	τ_1 X Z (0, τ_1)	…	DP_5

We have priced each **k-th to default basket** as separate instruments. We have re-used the simulations in pricing the instruments, however the default times would correspond to the k-th default.

For a 2nd-to-default instrument that protects against the default of a single name, fair spread would be

$$s=\frac{(1-R)Z(0,\tau_2)\times\frac{1}{5}}{Z(0,\tau_1)(\tau_1-0)\times\frac{5}{5}+Z(0,\tau_2)(\tau_2-\tau_1)\times\frac{4}{5}}$$

We make a simplifying assumption that reference entities that have defaulted before 2nd-default are not removed from the portfolio. So, the above equation would simplify to

$$S=\frac{(1-R)Z(0,\tau_2)\times\frac{1}{5}}{Z(0,\tau_2)\tau_2}$$

The other k-th to default instruments are priced based on the above formula with τ_k replacing τ_2 above. The price of a k-th to default instrument would be less than the price of a (k-1)-th to default instrument. This is because the probability of at least 1 default occurring is higher than the probability of at least 2 default occurring. Hence the number of iterations or simulation instances where a default payout would occur would be higher for a (k-1)-th to default instrument compared to a k-th to default instrument. The Premium Leg anyway would accrue till the time of the default and to 5 years in case there are no defaults. So, we would have more instances in the K-th to default cases where the premium would accrue until 5 years. So, in case of a k-th default instruments, we would end up with a default leg average that is lower than and a premium leg average that is higher than the corresponding averages for a (k-1)-th to default instrument. As a result, the spread of a (k-1)-th instrument would be higher than a k-th to default instrument.

This is what we have seen in the pricing as well. For both Gaussian and Student t Copula sampling, we can see that the fair spread for the 1st-to-default swap is the highest and spread for the 5th-to-default swap is the lowest among the instruments.

Fair spread calculated based on sampling from Gaussian Copula

The fair spread is calculated based on a recovery rate (R) of 40% and a total of 100,000 iterations

Instrument	**Fair Spread**
1st-to-default	37.67909388
2nd-to-default	7.48463771
3rd-to-default	1.60611377
4th-to-default	0.31195144
5th-to-default	0.04996653

Fair spread calculated based on sampling from Student T Copula

Instrument	**Fair Spread**
1st-to-default	32.7985198
2nd-to-default	8.5288394
3rd-to-default	3.0872167
4th-to-default	1.0778433
5th-to-default	0.2768468

Student t Copula signifies stronger co-movement. If there is a high value occurring for a particular name, the probability of a high value for other names would be quite high as well. As a result, under Student t Copula, instances of multiple defaults occurring together would be greater than the instances of multiple defaults under the Gaussian Copula. The higher co-movement would result in a higher spread for K value higher than 1. The spread would also vary as a more precise correlation matrix (Kendall Tau) is used for sampling from Student t copula. This is evidenced from the fair spread prices we have obtained for K=1 to 5 for Gaussian and Student t Copula.

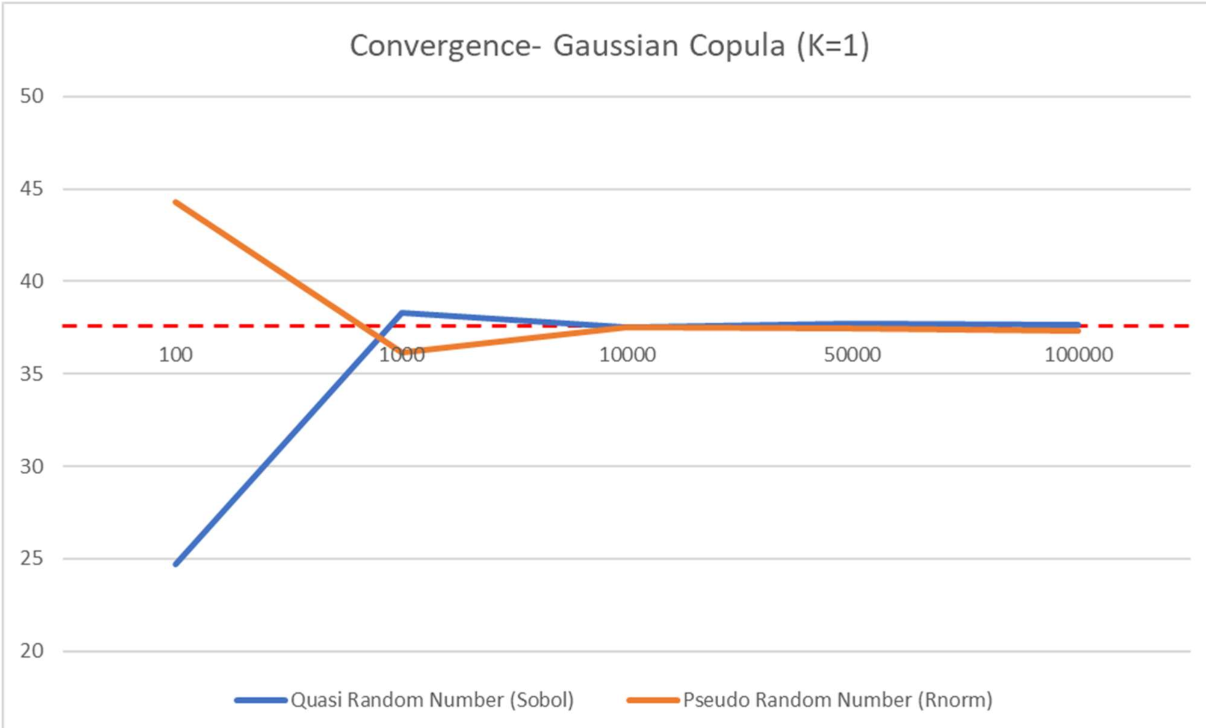
Convergence of Fair Spread

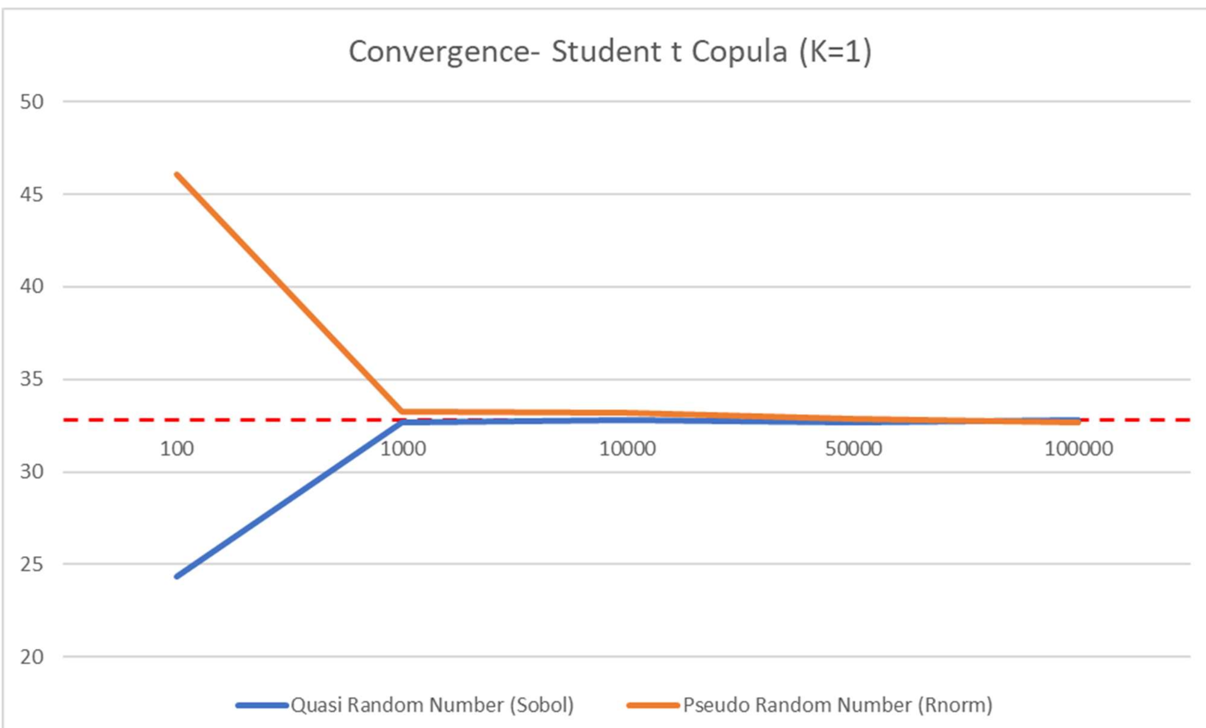
We have performed the fair spread computations for K=1 to 5 for different number of iterations- 100, 1000, 10000, 50000, and 100000. For Both Gaussian and Student t Copula sampling, we see that the spreads tend to converge faster (i.e., converging to a stable value with a smaller number of iterations) for the low-discrepancy numbers generated using the Sobol sequence. Simulations using pseudo-random number generated using the rnorm() function of the R package was slower to converge. This was more clearly visible for higher values of K (K>2).

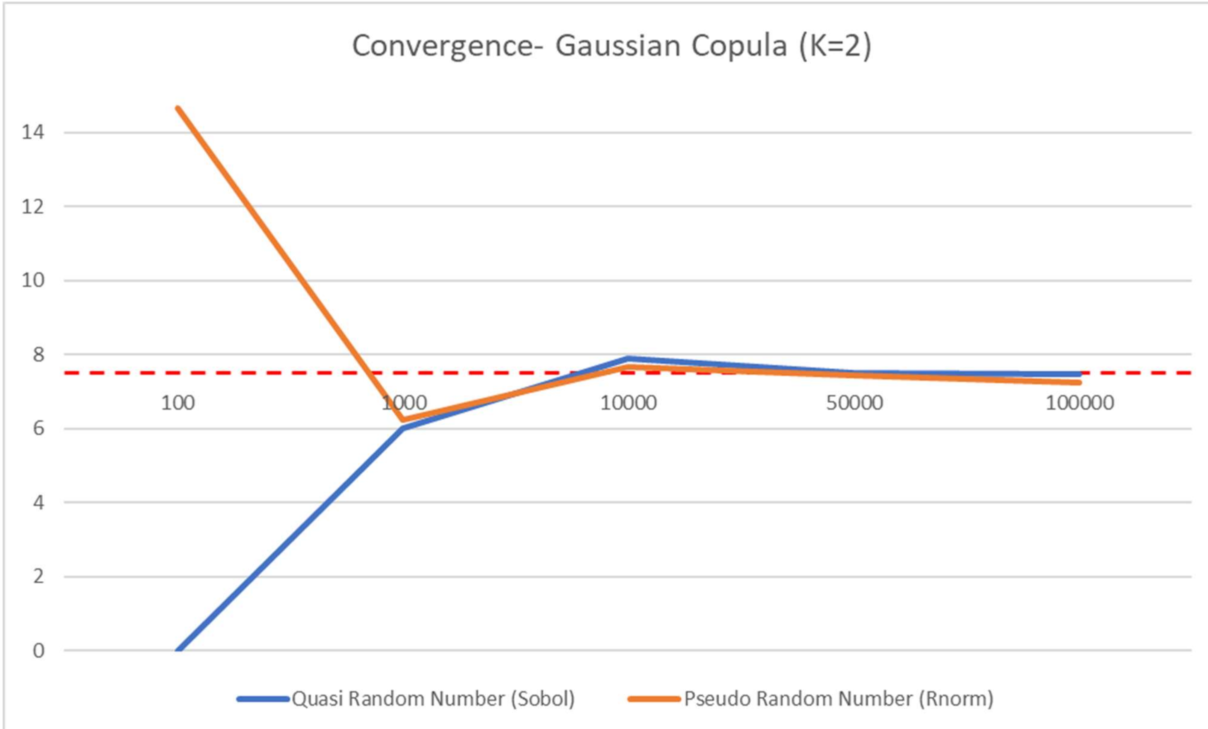
The points in the quasi-random sequence (also called low-discrepancy sequence) such as the Sobol sequence are correlated to provide greater uniformity. The resulting simulation i.e., the Quasi-Monte Carlo has a convergence rate of approximately $O((\log N)^k N^{-1})$, where k is a constant.

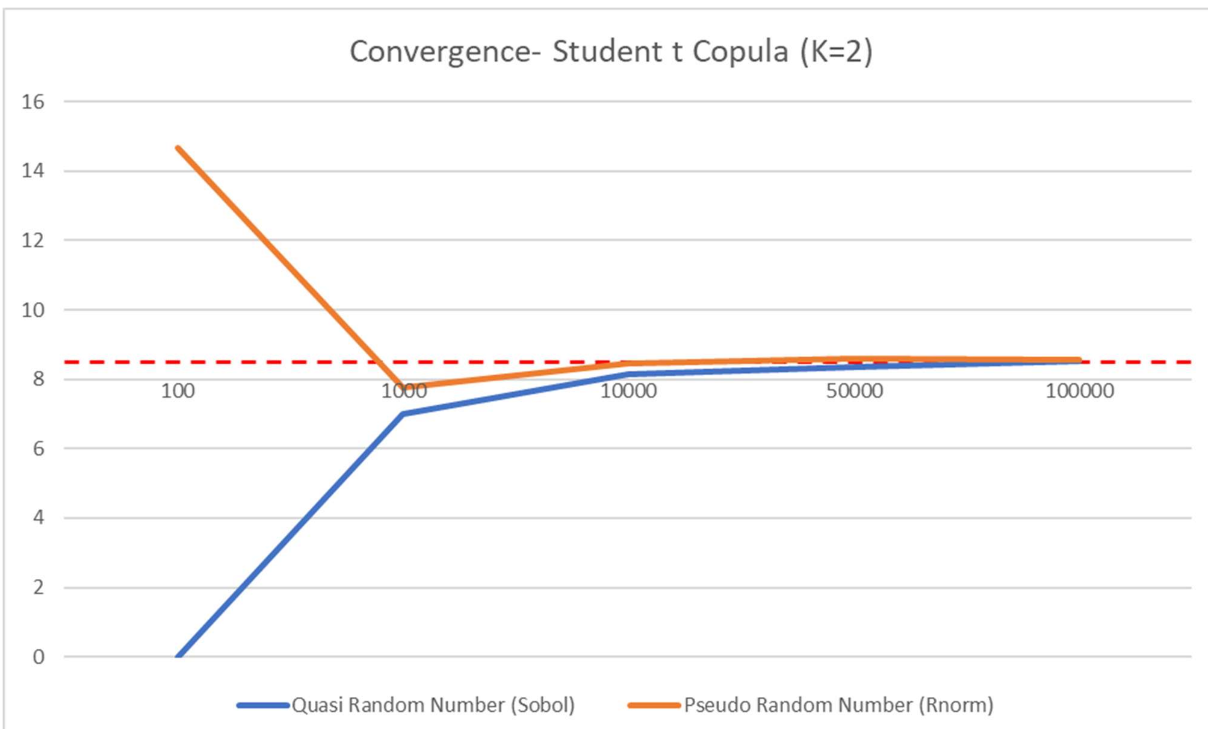
A Monte Carlo simulation using random or pseud-random sequences on the other hand has a convergence rate of $O(N^{-1/2})$. This means that the Monte Carlo simulation using quasi-random numbers converge more rapidly compared to Monte Carlo sequence using random/pseudo-random numbers. For Pseudo-random number generators such as rnorm(), a draw of numbers typically does not have a knowledge of the previous draw and hence suffer for lower sampling efficiency. This can lead to clustering of data points and data gaps that may prevent achieving a stable price with a smaller number of iterations.

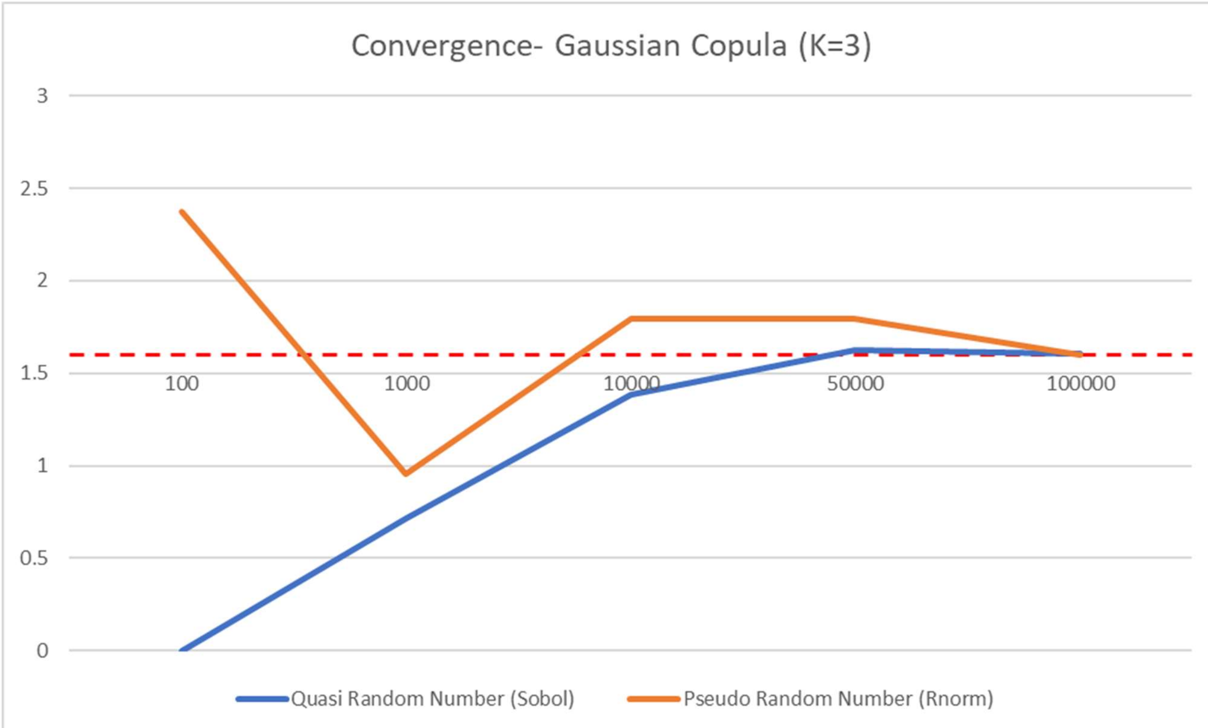
Convergence diagrams for different Ks and for both Gaussian and Student t Copulas are shown below-

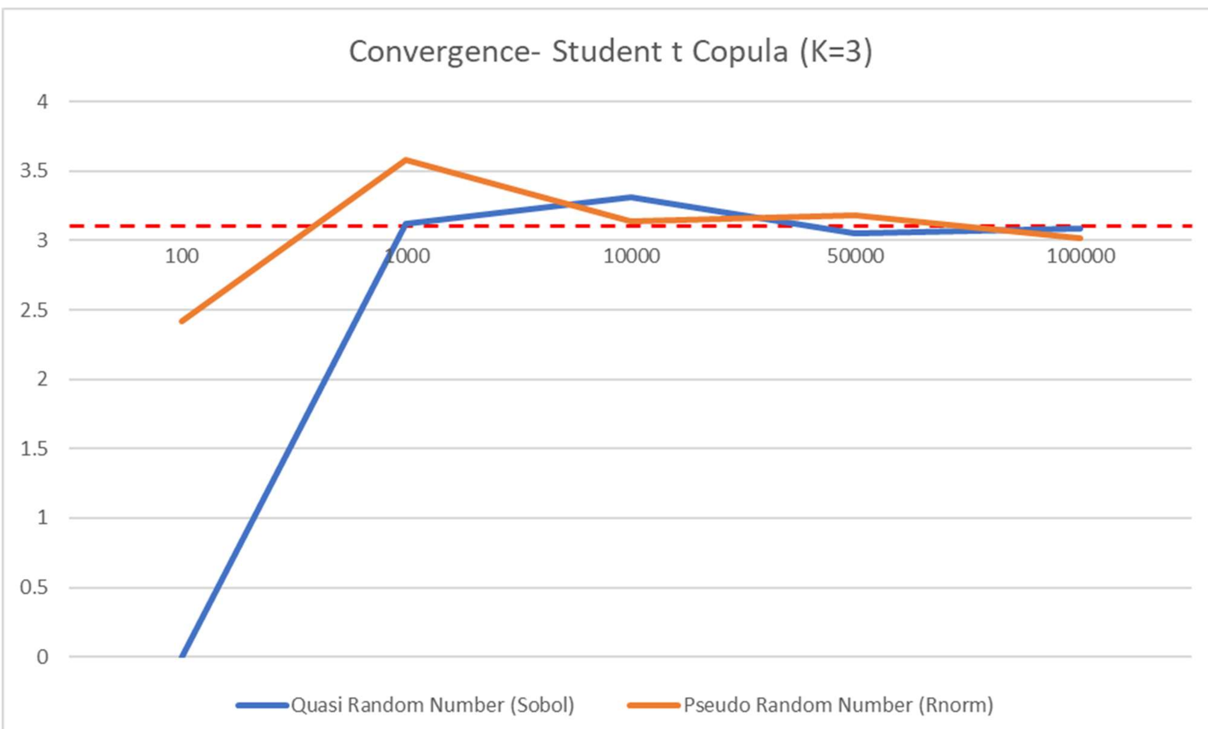

Convergence- Gaussian Copula (K=1)
50
45
40
35
30
25
20
100
1000
10000
50000
100000
Quasi Random Number (Sobol)
Pseudo Random Number (Rnorm)

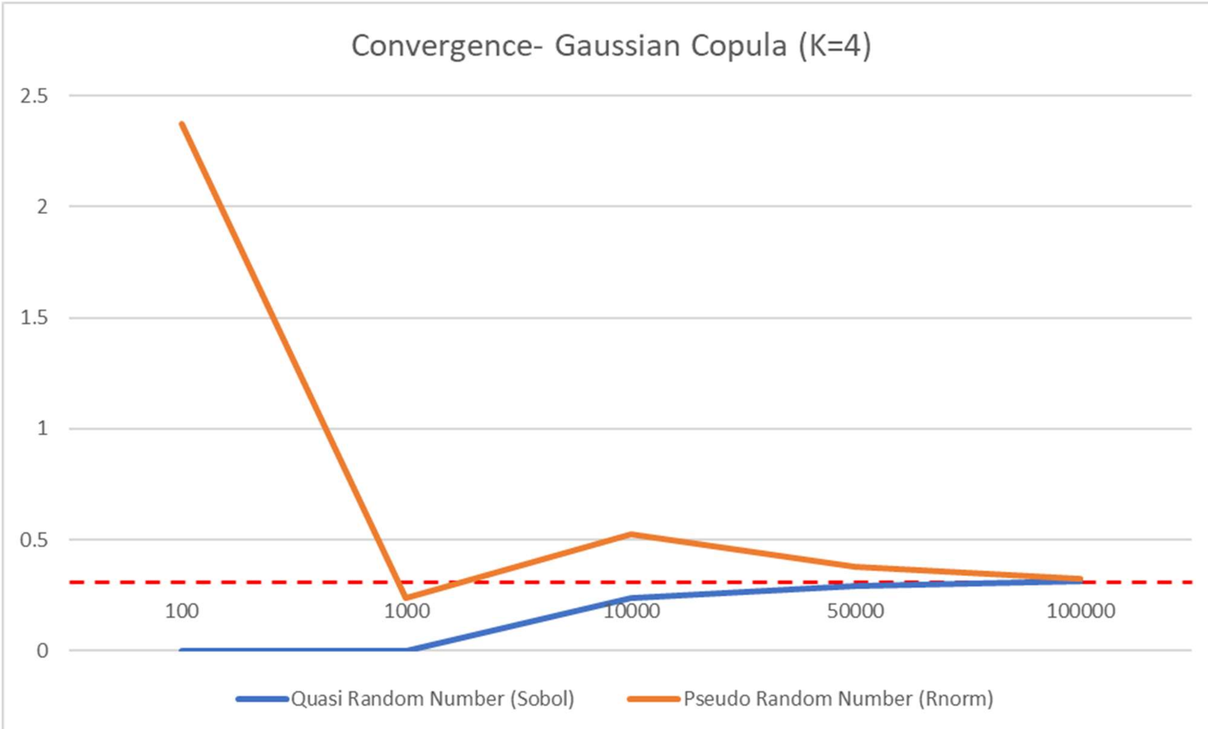

Convergence- Student t Copula (K=1)
50
45
40
35
30
25
20
100
1000
10000
50000
100000
Quasi Random Number (Sobol)
Pseudo Random Number (Rnorm)

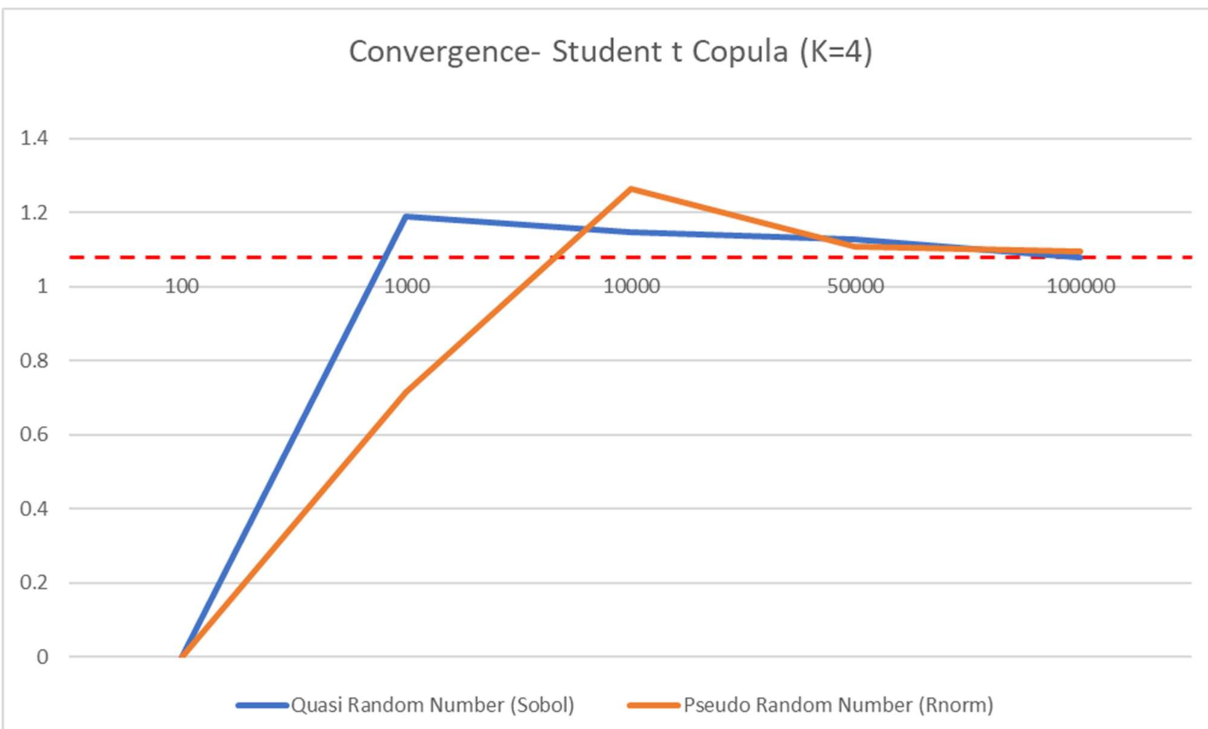

Convergence- Gaussian Copula (K=2)
14
12
10
8
6
4
2
0
100
1000
10000
50000
100000
Quasi Random Number (Sobol)
Pseudo Random Number (Rnorm)

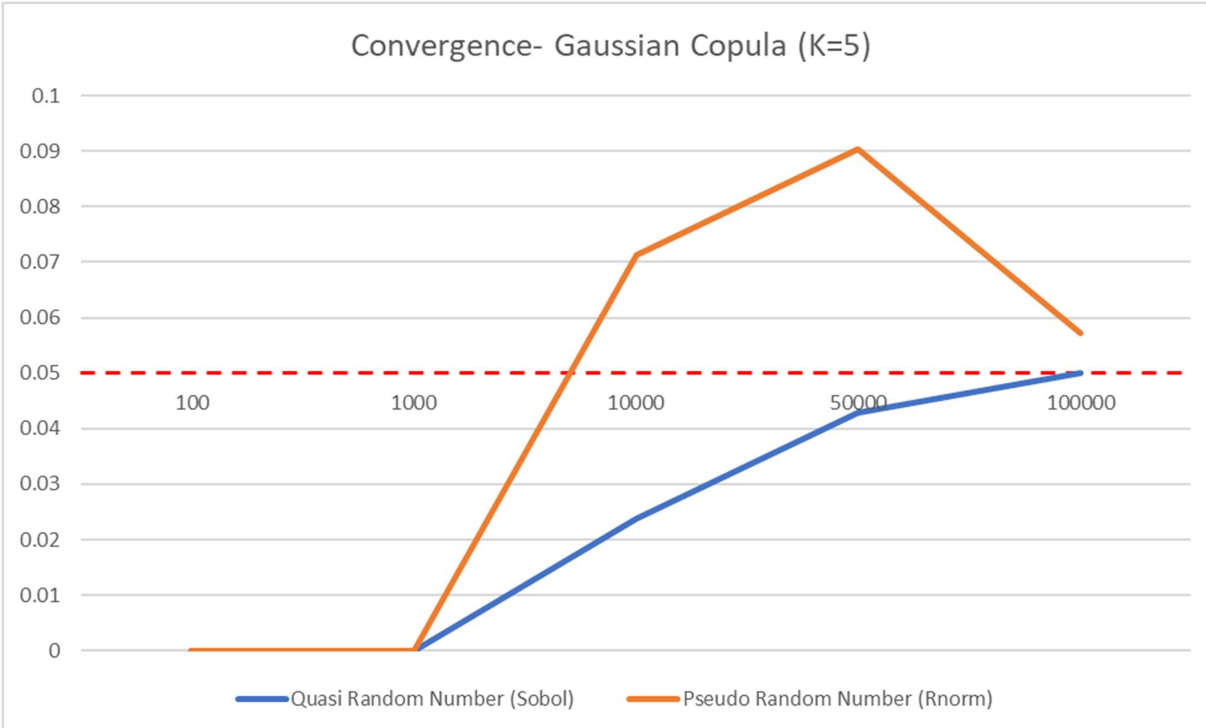

Convergence- Student t Copula (K=2)
16
14
12
10
8
6
4
2
0
100
1000
10000
50000
100000
Quasi Random Number (Sobol)
Pseudo Random Number (Rnorm)

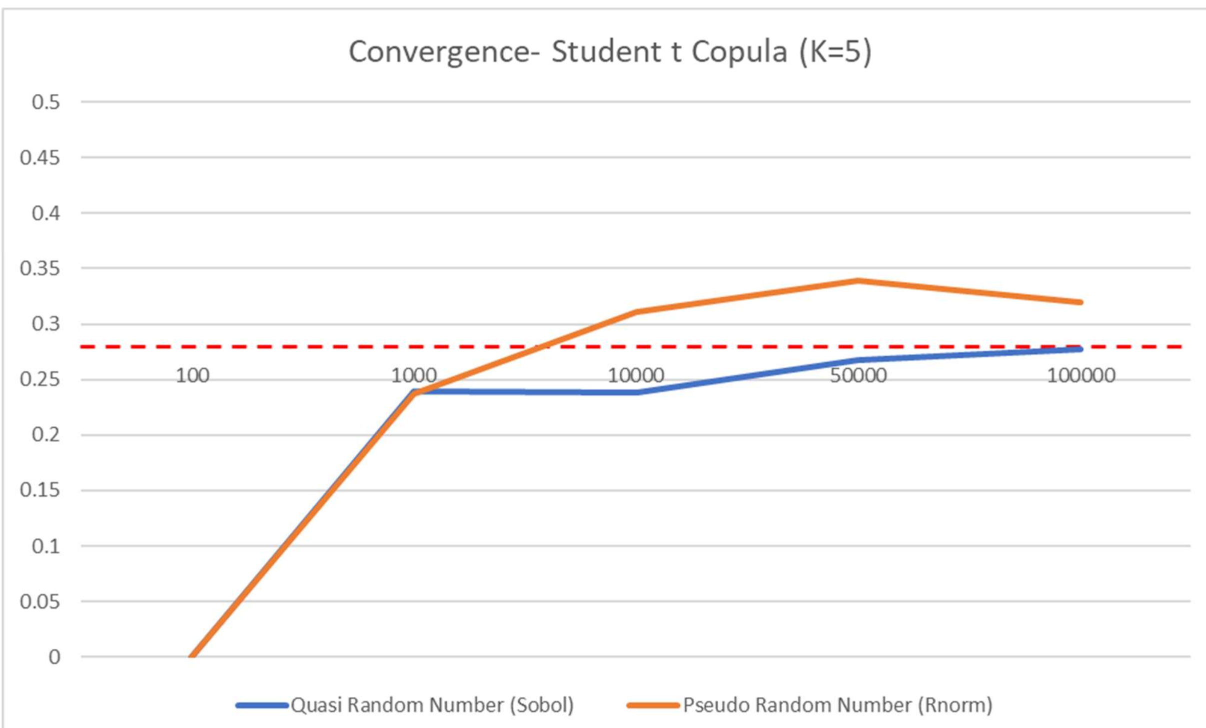

Convergence- Gaussian Copula (K=3)
3
2.5
2
1.5
1
0.5
0
100
1000
10000
50000
100000
Quasi Random Number (Sobol)
Pseudo Random Number (Rnorm)


Convergence- Student t Copula (K=3)
4
3.5
3
2.5
2
1.5
1
0.5
0
100
1000
10000
50000
100000
Quasi Random Number (Sobol)
Pseudo Random Number (Rnorm)


Convergence- Gaussian Copula (K=4)
2.5
2
1.5
1
0.5
0
100
1000
10000
50000
100000
Quasi Random Number (Sobol)
Pseudo Random Number (Rnorm)


Convergence- Student t Copula (K=4)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
100
1000
10000
50000
100000
Quasi Random Number (Sobol)
Pseudo Random Number (Rnorm)





Sensitivity Analysis

In this section we explore the impact of the input parameters i.e., reference credit spreads, correlation, and recovery rate on the fair spread of the basket credit swap. We vary one parameter at a time keeping others at their base value used in the pricing. Sensitivity analysis would tell us how the pricing would vary in the stressed situations.

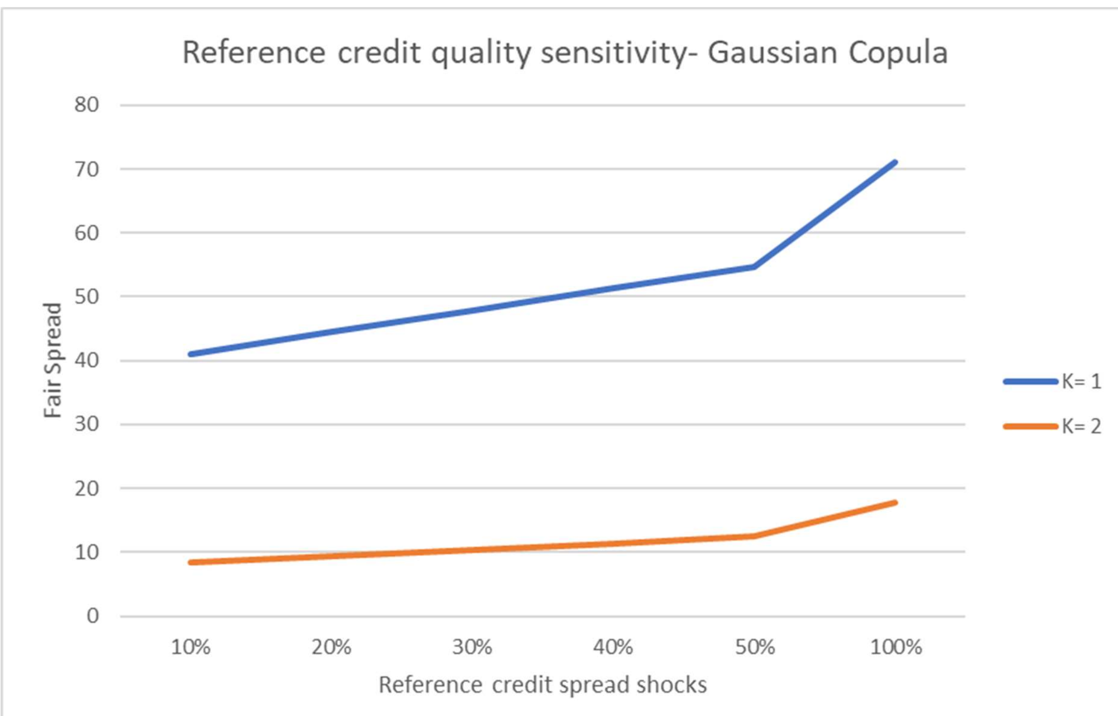
A. Impact of credit quality of individual reference names

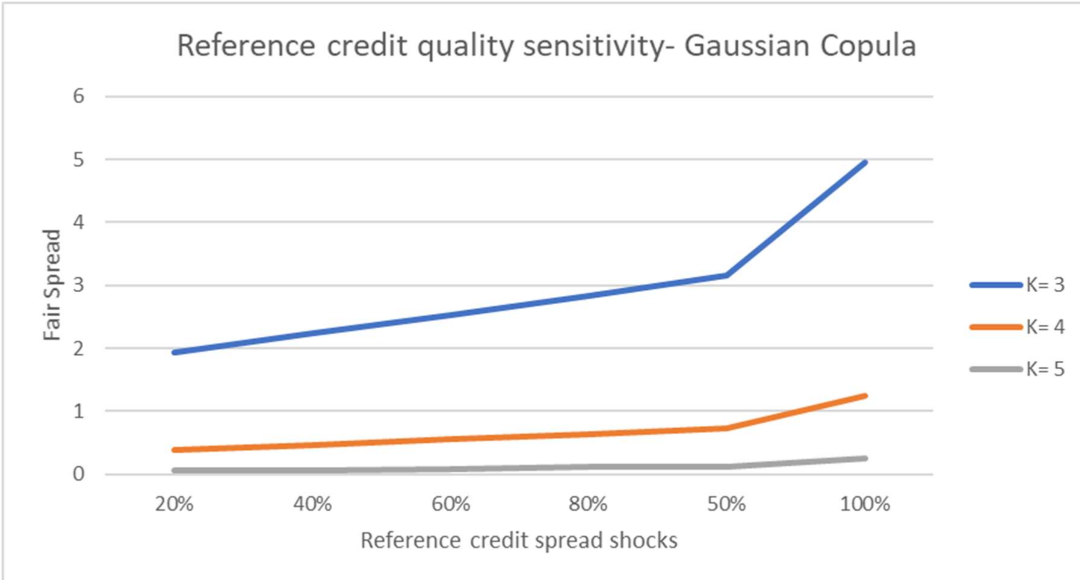
The sensitivity of the fair spread against the credit quality of the reference names is determined by applying shocks (multipliers) to the credit spread of the reference names. The other parameters i.e., recovery rate (40%) and correlation are kept at their base values used in the pricing of the fair spread. Number of iterations used is 100,000. The credit spread shocks applied are 10%, 20%, 30%, 40%, 50%, and 100%.

The decrease in the credit quality of the reference names (i.e., the increase in their CDS spreads) leads to an increase in the fair spread of the basket credit swap. This is not surprising as the reference names are positively correlated and a decrease in the credit quality of individual names would result in a decrease of credit quality of the basket as a whole. However, we can see that the impact of the credit quality change is more pronounced at higher Ks. With a relatively lower occurrence of corresponding default events, a change in credit quality results in a more pronounced impact on the fair spread of the instrument. The results are consistent across both Gaussian and Student t copula fair spreads.

Fair spread using Gaussian Copula

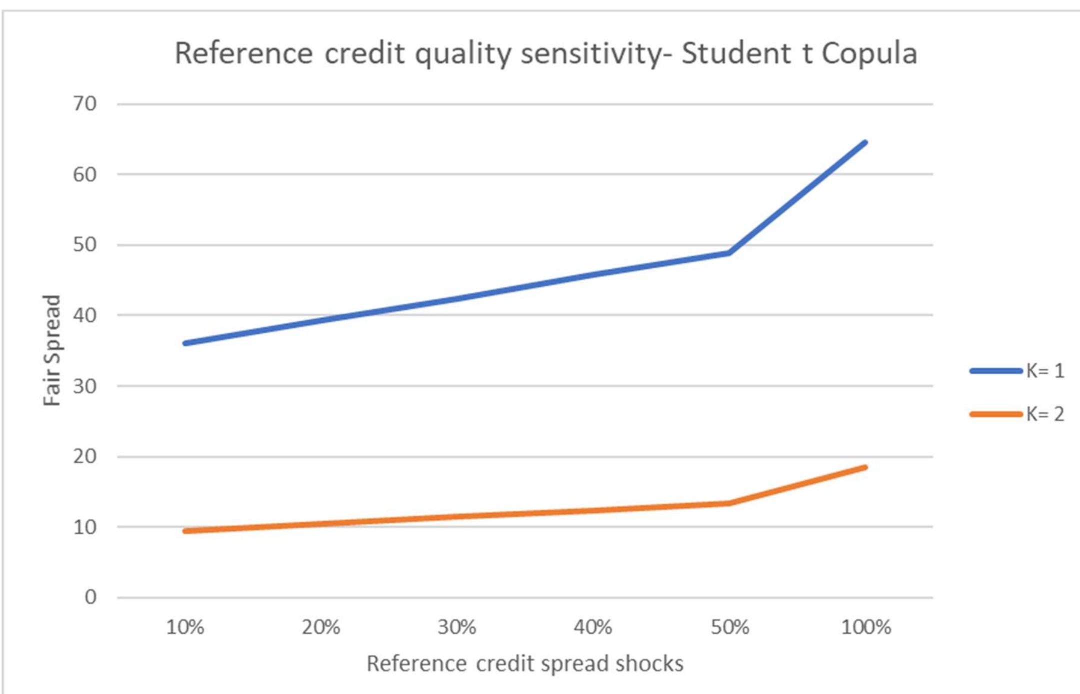
Credit Spread Shock	10%	20%	30%	40%	50%	100%
K= 1	41.0929125	44.57088831	47.94356974	51.3153548	54.598962	71.0526794
K= 2	8.3999165	9.39691244	10.40956004	11.3843355	12.4506528	17.8613402
K= 3	1.9431000	2.23982641	2.52939602	2.8241245	3.1551471	4.9488998
K= 4	0.3905278	0.46676252	0.55257502	0.6454642	0.7312774	1.2466271
K= 5	0.0595336	0.06673356	0.08574652	0.1118553	0.1261327	0.2593005

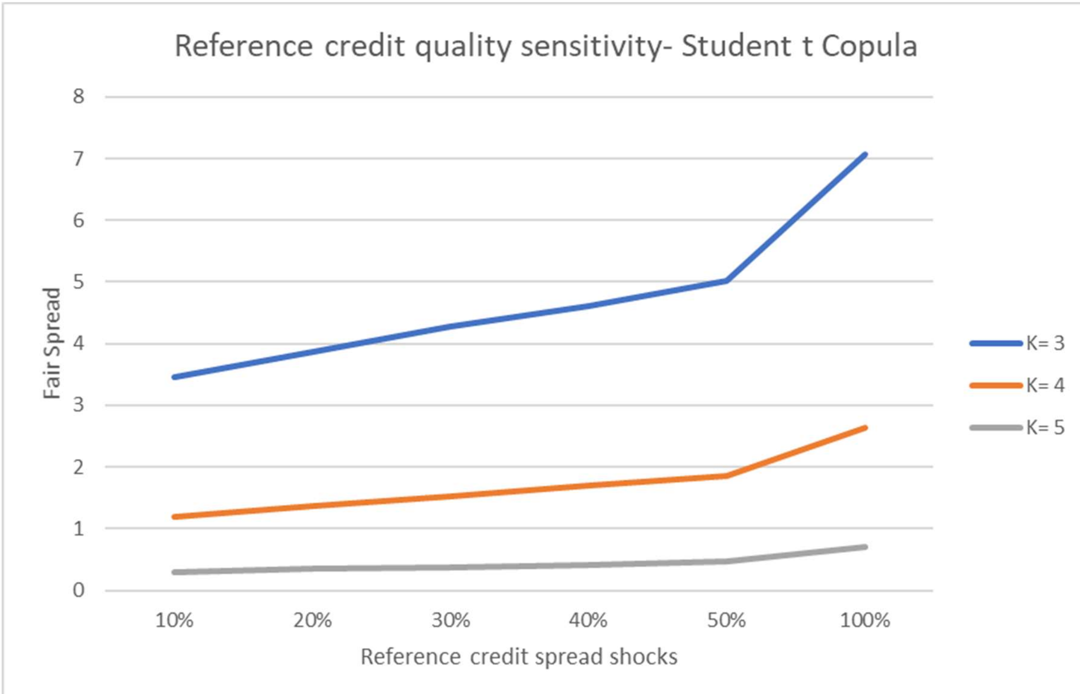




Fair spread using Student t Copula

Credit Spread Shock	10%	20%	30%	40%	50%	100%
K= 1	35.9989446	39.3344034	42.4059605	45.7388316	48.7975061	64.5784821
K= 2	9.4791321	10.4886844	11.4067945	12.3177785	13.3566265	18.5079209
K= 3	3.4652147	3.8672603	4.2769321	4.6177271	5.0261024	7.0743774
K= 4	1.2000844	1.3720849	1.5179906	1.6925687	1.8529597	2.6444614
K= 5	0.2888789	0.3484078	0.3817978	0.4175984	0.4629796	0.7016213





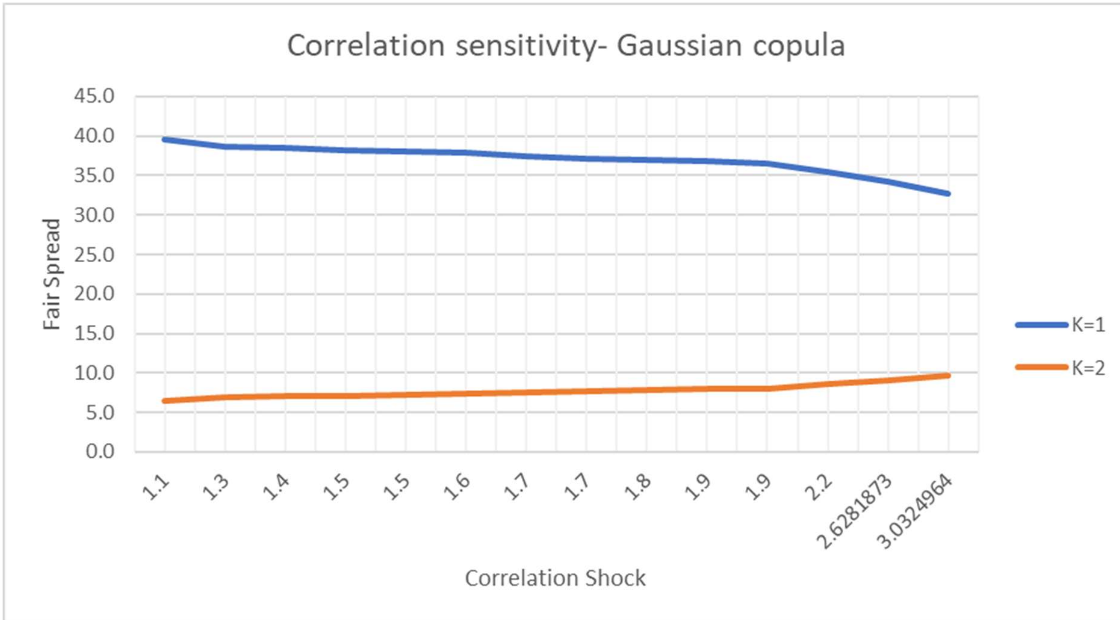
B. Impact of correlation on fair spread

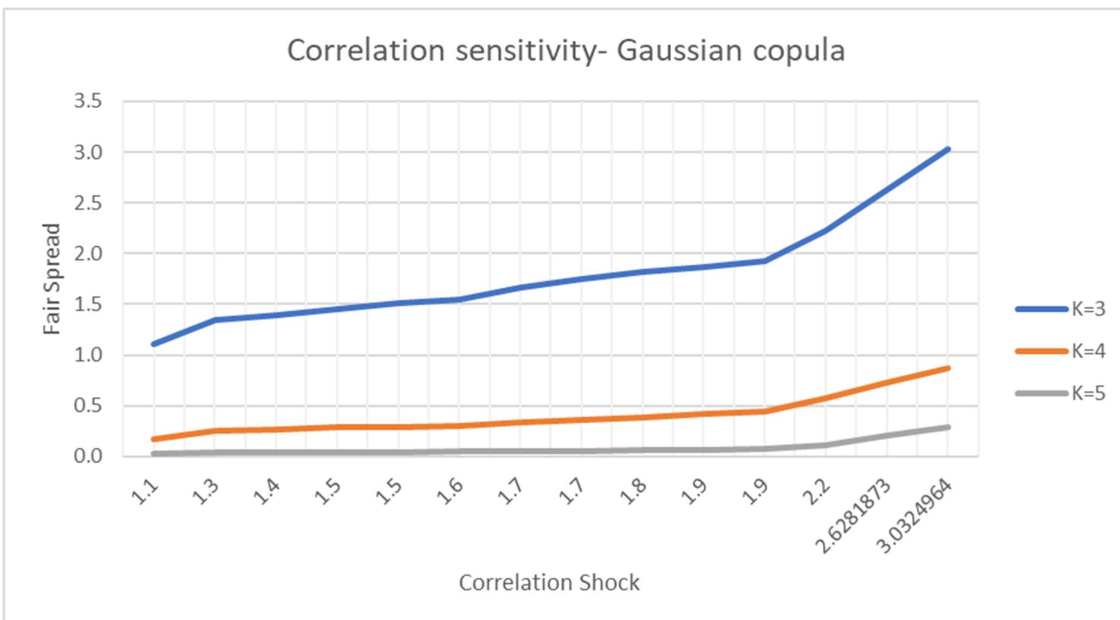
The correlation sensitivity has been calculated by stressing the linear correlation matrix used for Gaussian and Student T copula. We have used shocks ranging from -20% to 40% as shown in the tables below. The recovery rate has been kept at 40%, reference credit spreads have been kept at the base value used for pricing the basket derivatives, and number of iterations used is 100,000.

From the plots below, we can see that for 1st-todefault swap the fair spread decreases as correlation increases. However, from 2nd-to default swap onwards fair spread increases with increasing correlation. Where the reference names are entirely uncorrelated (i.e., correlation of 0), most of the time, only one reference name will default and the instances of multiple reference names defaulting within the same iteration of the simulation would be quite rare. As a result, the fair spread of the basket CDS would be close to the CDS spread of the reference names. However, where the reference names are correlated, we would see increasing intersections i.e., multiple names defaulting together. As a result, the number of only first-to-default instances within the simulation relative to instances with two or more defaults would decrease. This will lead a lowering of the fair spread of first-to-default swaps and increase of the spread of 2nd-to-default swap onwards (i.e., for K>1). We have observed this for both Gaussian and Student t Copula spreads.

Fair spread using Gaussian Copula

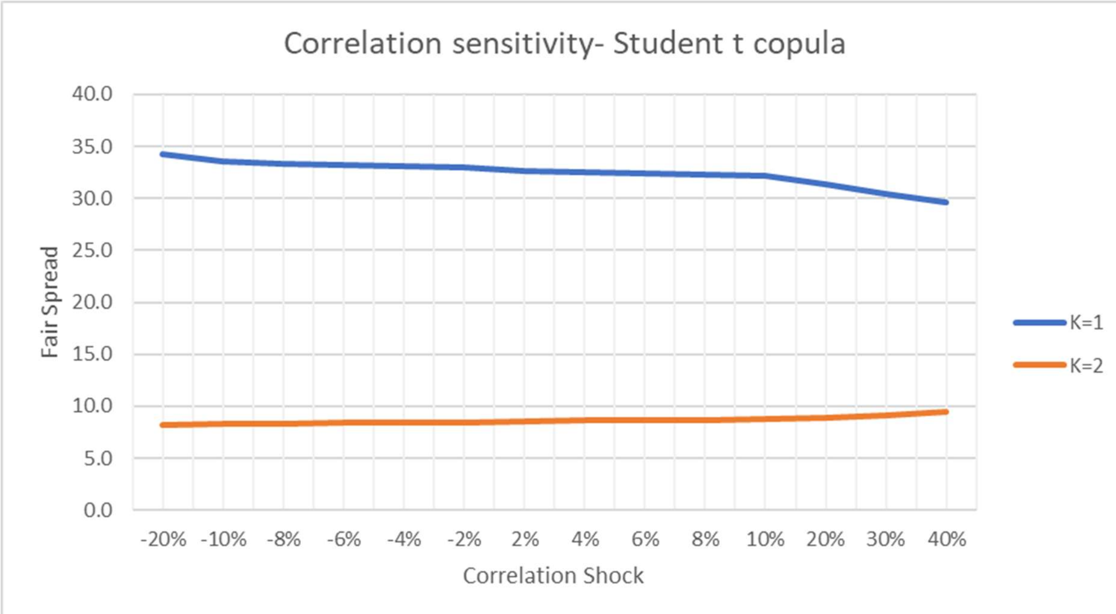
Correlation shock	-20%	-10%	-8%	-6%	-4%	-2%	2%	4%	6%	8%	10%	20%	30%	40%
K=1	39.6	38.7	38.5	38.3	38.1	37.9	37.5	37.2	37.0	36.8	36.6	35.5	34.2	32.7
K=2	6.4	6.9	7.0	7.1	7.2	7.4	7.6	7.7	7.8	7.9	8.0	8.6	9.1	9.7
K=3	1.1	1.3	1.4	1.5	1.5	1.6	1.7	1.7	1.8	1.9	1.9	2.2	2.6	3.0
K=4	0.2	0.2	0.3	0.3	0.3	0.3	0.3	0.4	0.4	0.4	0.4	0.6	0.7	0.9
K=5	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.3

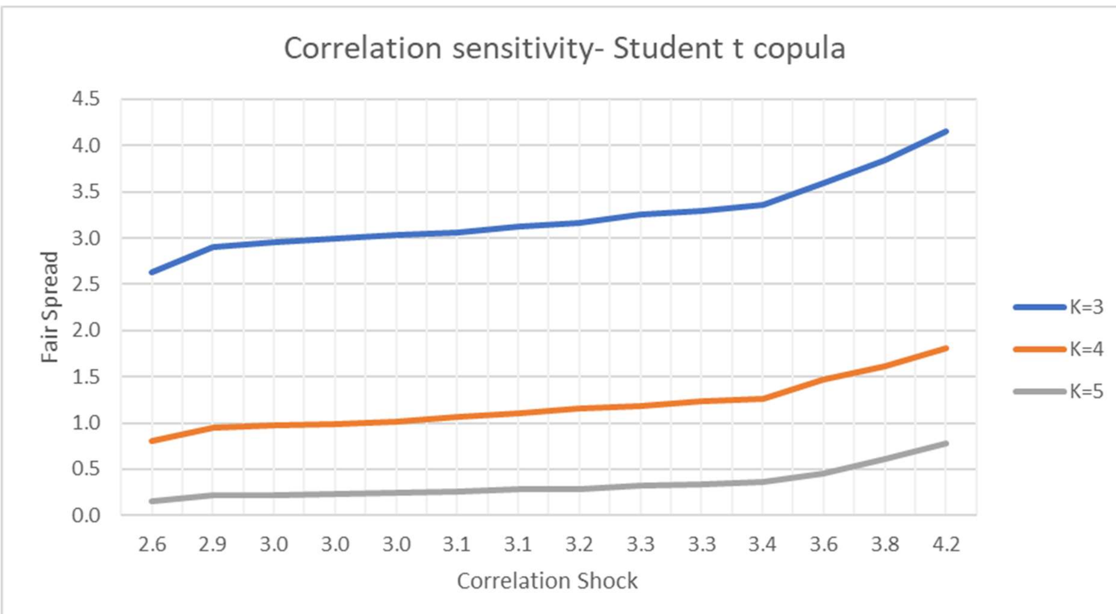




Fair spread using Student t Copula

Correlation shock	-20%	-10%	-8%	-6%	-4%	-2%	2%	4%	6%	8%	10%	20%	30%	40%
K=1	34.3	33.5	33.4	33.2	33.1	32.9	32.7	32.6	32.4	32.3	32.1	31.3	30.5	29.6
K=2	8.2	8.3	8.3	8.4	8.4	8.5	8.6	8.6	8.6	8.7	8.7	8.9	9.2	9.5
K=3	2.6	2.9	3.0	3.0	3.0	3.1	3.1	3.2	3.3	3.3	3.4	3.6	3.8	4.2
K=4	0.8	0.9	1.0	1.0	1.0	1.1	1.1	1.2	1.2	1.2	1.3	1.5	1.6	1.8
K=5	0.1	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3	0.3	0.4	0.5	0.6	0.8





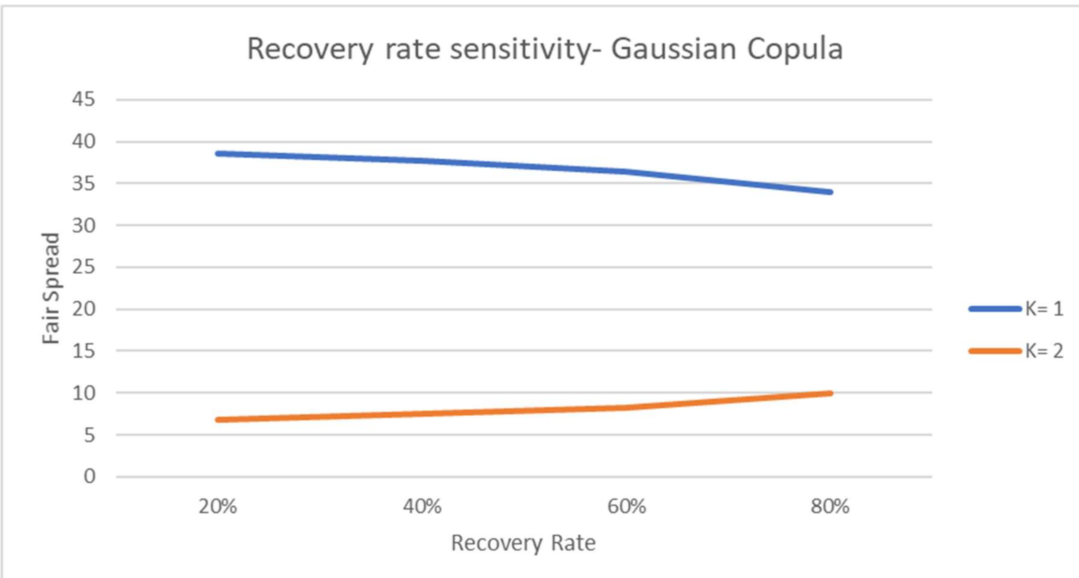
C. Impact of the recovery rate on fair spread

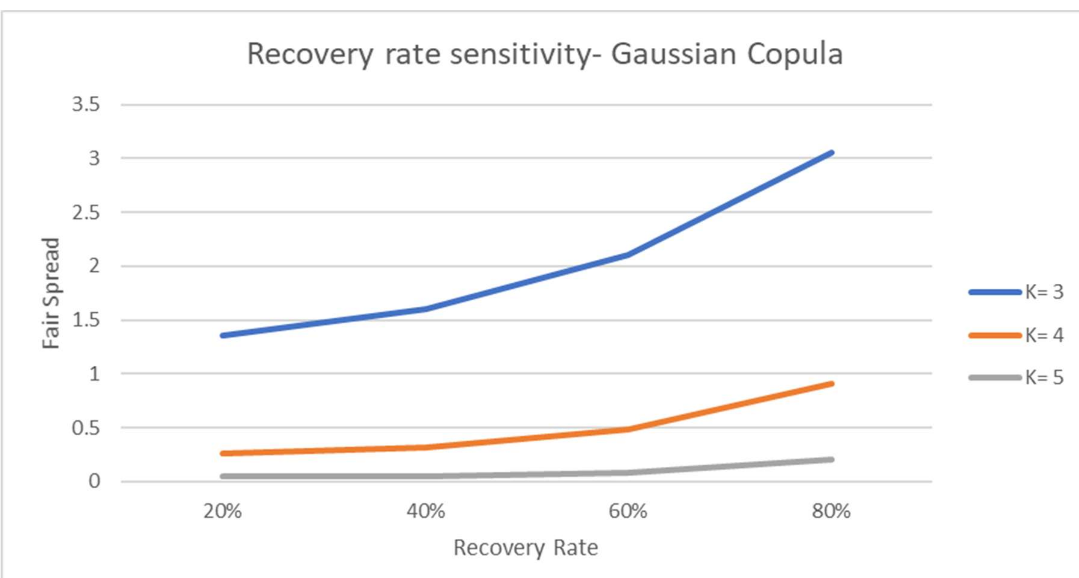
We have calculated the recovery rate sensitivity by varying the recovery rate from 20% to 80% in the increment of 20%. The number of iterations used is 100,000. The resultant change in the fair spread shows that the fair spread decreases for 1st-to default instrument with increasing recovery rate. We have observed this for both Gaussian and Student t spreads. However, for 2nd-to default instrument onwards (i.e., for K>1), the fair spread decreases with increasing recovery rate.

As the recovery rate increases, the pay-out at the time of default would decrease. This leads to a lowering of the spread. However, at the same time, a higher recovery rate would lead to a larger intensity or hazard spread in the bootstrapping process. This would have an impact of overall lowering of the default time i.e., defaults occurring sooner in the process and therefore having an impact of increasing the spread. The net impact would vary depending on the instrument being priced. So, for k=1, we can see that the effect of lower pay-out outweighs the effect of lowering of default time and as a result, we see a decrease in fair spread with increasing recovery rate. For K>1, the effect reverses.

Fair spread using Gaussian Copula

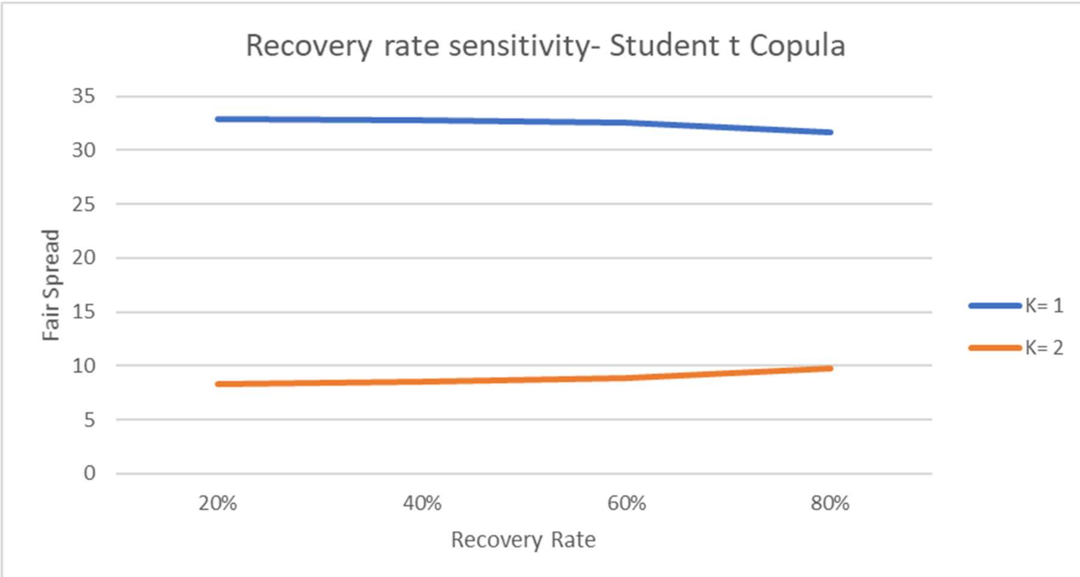
Recovery Rate	20%	40%	60%	80%
K= 1	38.53672715	37.67909388	36.399308	34.0160382
K= 2	6.86492332	7.48463771	8.30043517	9.9071017
K= 3	1.35777333	1.60611377	2.10343138	3.0554376
K= 4	0.25691392	0.31195144	0.48751830	0.9061487
K= 5	0.04754324	0.04996653	0.08408843	0.2007779

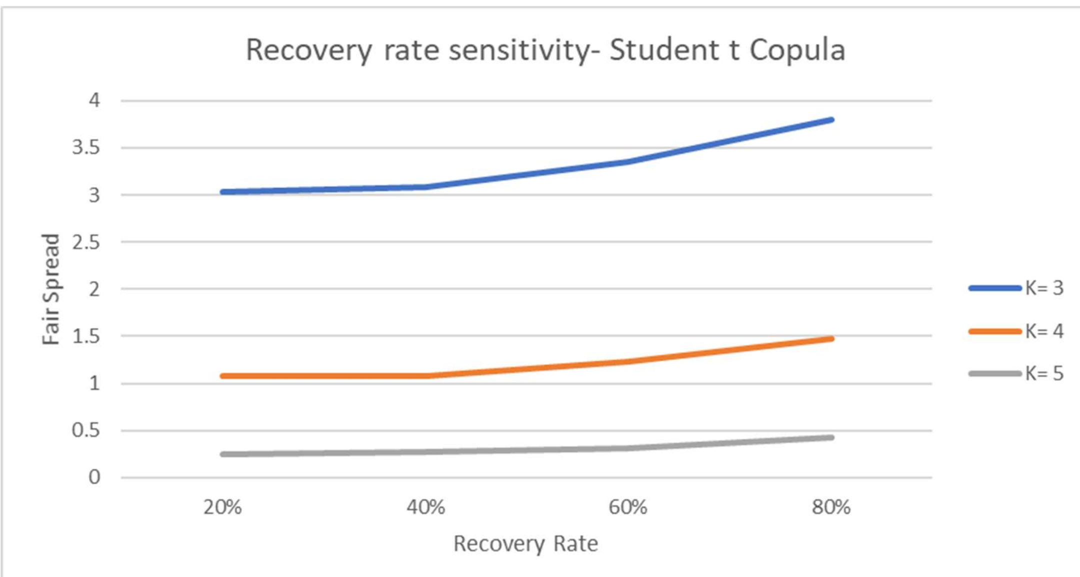




Fair spread using Student t Copula

Recovery Rate	20%	40%	60%	80%
K= 1	32.9167839	32.7985198	32.531671	31.6378726
K= 2	8.3005377	8.5288394	8.904418	9.7110162
K= 3	3.0356988	3.0872167	3.350735	3.7978721
K= 4	1.0731787	1.0778433	1.235306	1.4741796
K= 5	0.2482228	0.2768468	0.308653	0.4225071





Appendix

A. Data used in the process

- Five separate CSV files containing the stock price data corresponding to the five different names. The data include the log return of the stock prices for the last five years, ending on 15 December 2020
- A CSV file with the US Treasury Rates on 15 Dec 2020. We have fit a quadratic equation based on these rates to derive the discount factors corresponding to various tenors
- The credit spreads for the five reference names on 15 Dec 2020 for the tenors from 1 to 5 years. This is an Excel file. The data is used to bootstrap the hazard rates term structure

B. R code structure

We have arranged the code in three different R files-

- The first R script is used to determine the Pearson and Kendall Tau linear correlation matrices used in the Gaussian and Student t Copula sampling respectively. This piece of code uses the stock price return data as input
- The second R file is used to determine the "degree of freedom" corresponding to the Student t Copula. The correlation matrices from the previous step are used as an input for this piece of code.
- The final file contains rest of the code and consists of the following key component
  - Initialisation of variables
  - Generating pseudo and quasi-random normal numbers and generating Ch-square values corresponding to the degree of freedom calculated in Step 2 as above
  - Determining the discounting factors by fitting a quadratic equation on the USD rates data for the reference date of 15 Dec 2020.
  - Bootstrapping of hazard rates based on the credit spread data as of the reference date
  - Sampling from Gaussian and Student t copula
  - Calculation of fair spreads for different Ks
  - Convergence/stability test
  - Sensitivity analysis
    - Sensitivity against correlation
    - Sensitivity against credit spreads
    - Sensitivity against recovery rate

References

1. Pricing basket credit default swaps. Avtar Sehra. ResearchGate
2. Copula based simulation procedures for pricing basket credit derivatives. Fathi, Abid and Nader, Naifar. Munich Personal RePec Archive (MPRA)
3. On bootstrapping hazard rates from CDS spreads. Giuseppe Castellacci. http://ssrn.com/
4. Introducing Monte Carlo methods with R. Christian P. Robert and George Casella. ResearchGate.
5. Pricing Kth to default basket CDS. Zhang, Xuchen.
6. Basket credit default swap. Bertrand Le Nezet.
7. Stochastic processes and advanced mathematical finance. Steven R. Dunbar.
8. Monte Carlo and quasi-Monte Carlo methods. Russel E. Caflisch. Cambridge University Press.
9. Maximum likelihood estimation of the correlation parameters for elliptical copulas. Lorenzo Hernandez, Jorge Tejero and Jaime Vinuesa. Quantitative Risk Research S.L. Madrid, Spain.