Pricing of kth-to-default Basket CDS

June 2023 Cohort

Sergei Beshenov

Abu Dhabi, 2024

Contents

Introduction

Description: Kth-to-Default Basket CDS

A kth-to-Default Basket CDS is a structured credit product that consolidates multiple individual credit default swaps. In contrast to a conventional credit default swap, which triggers a payout upon a credit event of a single reference entity, the kth-to-Default swap is activated only when the basket encounters its kth credit event.

To illustrate, if a trader purchases protection on a 5th-to-Default swap encompassing 20 entities, they will only receive a payout if 5 entities in the basket undergo a credit event before the swap's maturity. Should fewer than 5 entities default, the swap will conclude in the protection seller's favor. It's worth noting that the payout is determined solely by the notional value of the 5th defaulting entity, rather than the cumulative value of all defaulting entities.

The pricing of fair spreads necessitates consideration of both the credit risk characteristics of each constituent and the interdependency of risk profiles among constituents. Consequently, the fair spread is defined as an expectation across the joint distribution of default times. Default times are simulated through the sampling of correlated pseudo-samples from a copula using Monte Carlo methods, and these samples are then transformed into default times by utilizing individual hazard rate term structures. The fair spread is subsequently computed by averaging the default legs and premium legs for each simulation separately. Each kth-to-default tranche is priced separately, and the interrelationships between tranches are elucidated in this paper.

Here in this paper, we attempt to price the Basket CDS on five sovereign names: UAE, QATAR, Kingdom of Saudi Arabia (KSA), Oman, Bahrain. The choice of constituents is inspired by the current place of residence and the job of the author.

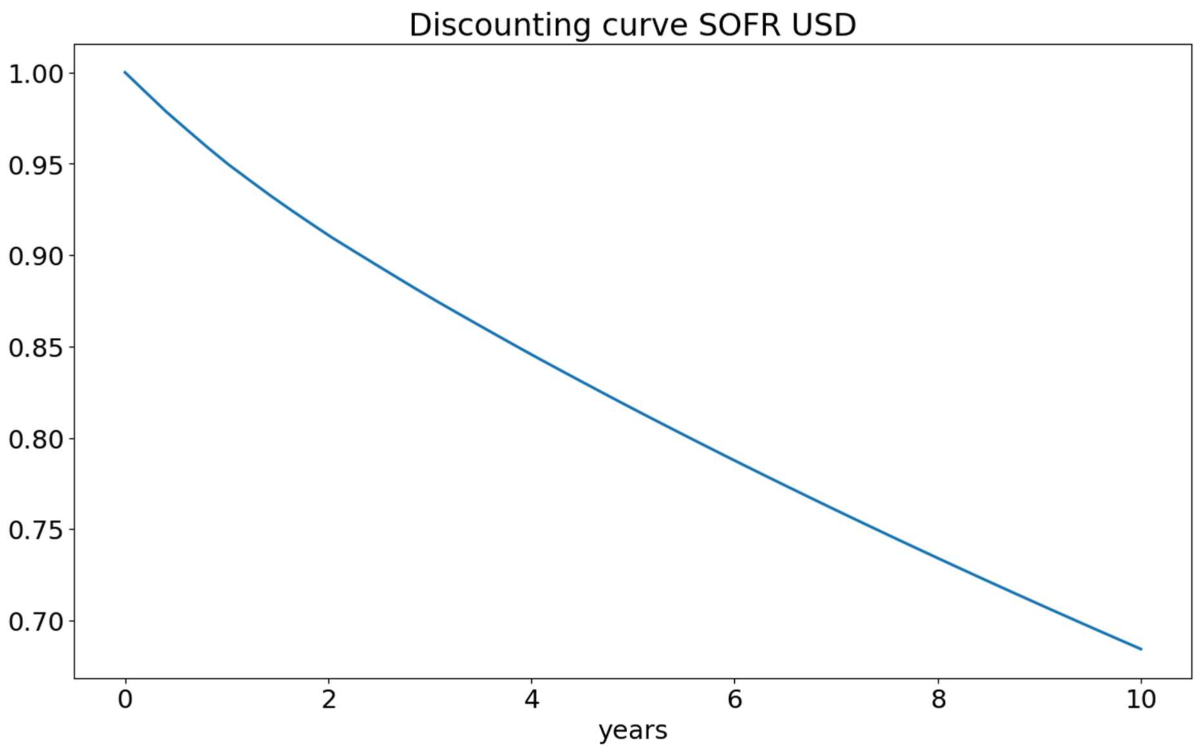
Data for the project

The data were collected from two main data sources: Datagrapple.com and Bloomberg.

Credit spread term structure and discounting curve were taken from the Bloomberg terminal.

Credit spread term structure can be seen in the Annex 1. All CDS curves are monotonically increasing, and have bid-ask spread, which varies by term. The narrowest bid-ask spread can be observed for 5Y maturities as these are the most traded contracts. This is why we will be using 5Y historical spreads for correlation analysis.

For discounting, we use the SOFR forward curve as current market practice given the recent Libor transition. Forward rates were transformed to discounting rates, while discounting rates would be interpolated by log-linear interpolation as it was shown in CQF Python lab 17. Below you can see a chart with an interpolated discounting curve:



The interpolated curve is monotonically decreasing which is consistent with the idea of no-arbitrage pricing.

Datagrapple.com is a CDS spreads provider containing price information on credit default swaps on more than 600 companies and credit derivatives indices over the last 8 years. We were able to source sovereign data on 5Y CDS spreads with up to 20 years of history. For this paper, we are using only the last 5 years, which are shown on the chart below:

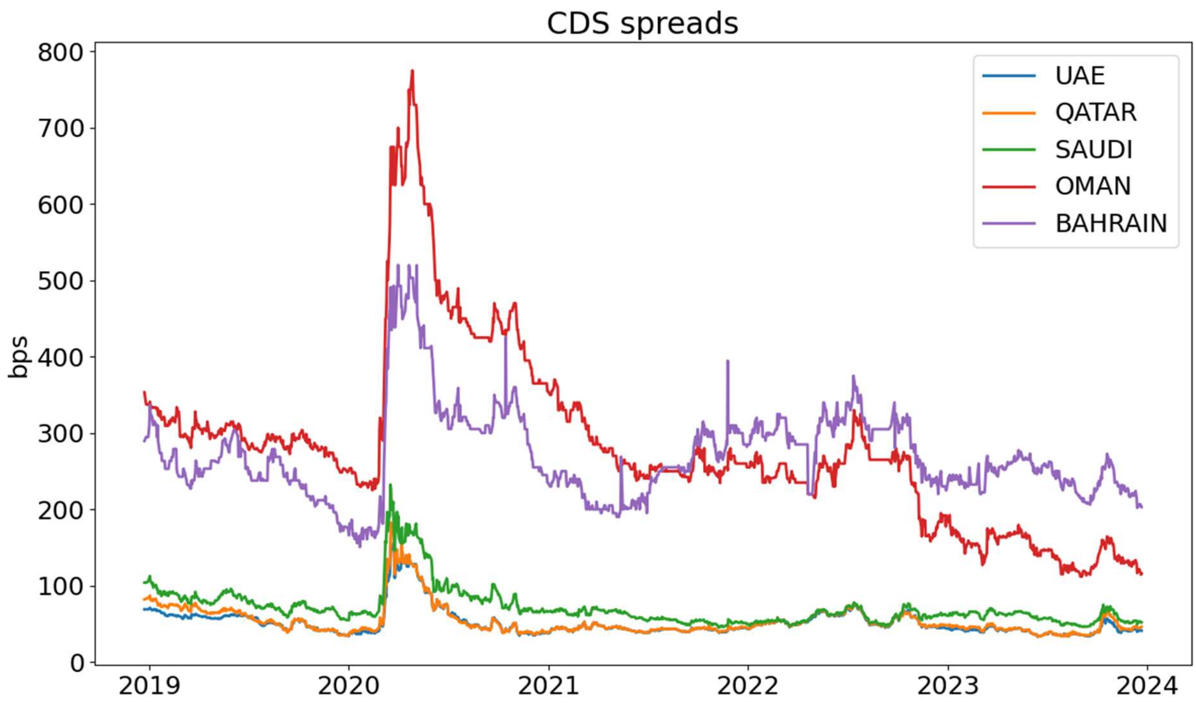


Figure 1. Historical data for 5Y sovereign CDS. Produced in Python.

From the chart below, we see that three countries have very similar levels: QATAR, UAE, KSA. These countries are historically very stable, have the highest ratings in the region, strong economies and high GDP per capita. UAE and QATAR seem to be highly correlated as they behave in a very similar manner, while KSA's spread is somewhat around but not the same. Oman and Bahrain are highly volatile and seem to be driven by their idiosyncratic credit risk factors rather than simply following the dynamics of the region.

We also checked the marginal distribution of the spreads as well as drew scatter plots for country pairs (Figure 2). The observed marginal distributions are leptokurtic which means fatter tails compared to normal distribution. Hence, the computation of Pearson's correlation straight away would undermine the actual possibilities of extreme events. Based on scatter plots, High correlations are seen for Qatar, UAE and KSA.

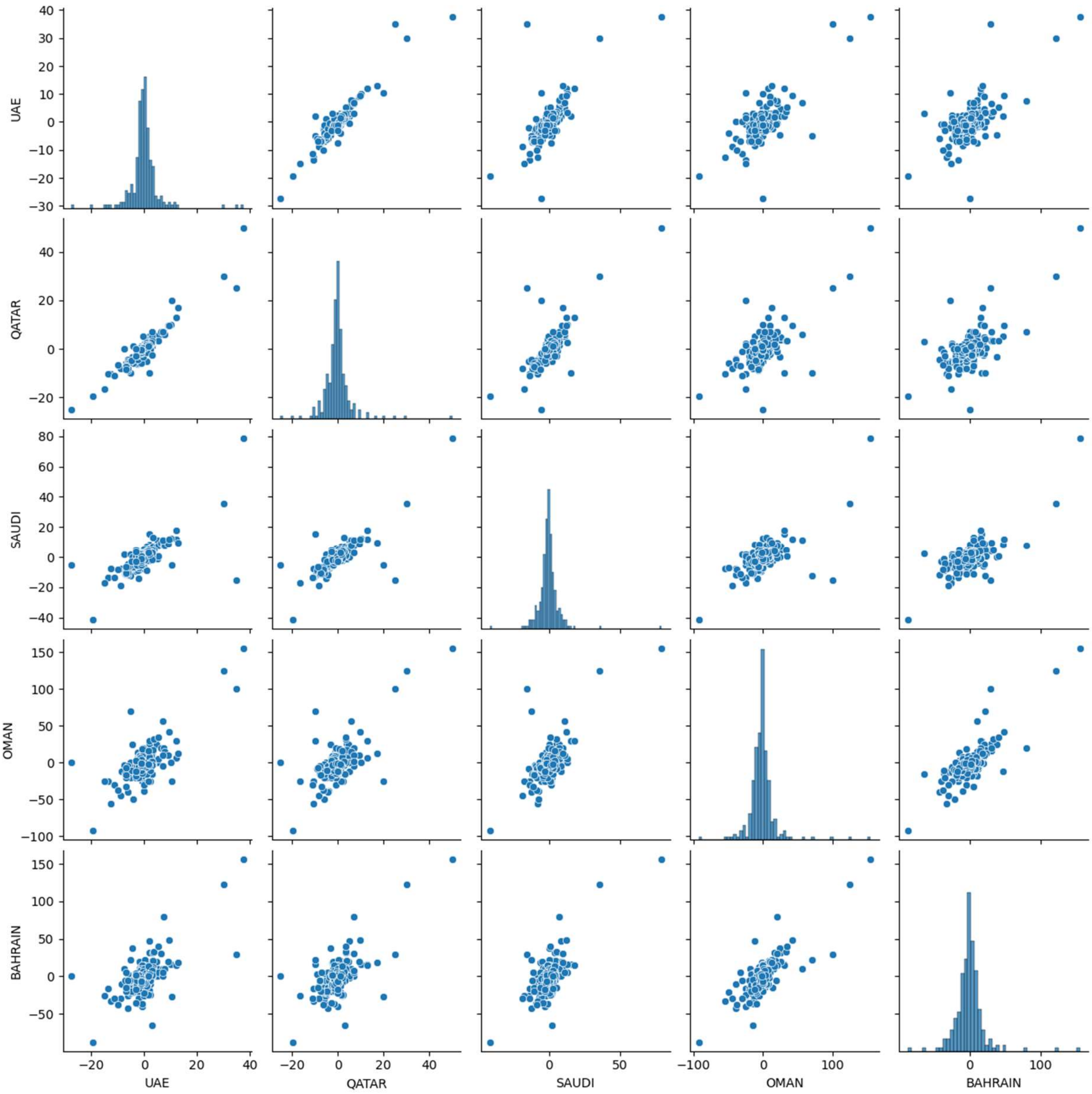


Figure 2. Histograms (on the diagonal) and pair scatter plots for weekly 5Y CDS differences. Produced in Python.

Definitions

Before proceeding to mathematical formulas of underlying concepts we briefly go through their meanings:
Default Probability: The default probability, often denoted as PD (Probability of Default), is the likelihood that a debtor will fail to meet its debt obligations over a specified time horizon. It represents the risk of a borrower defaulting on a loan or bond within a given period.
Survival Probability: The survival probability refers to the likelihood that a debtor or entity will continue to fulfil its debt obligations and remain solvent over a specific period. It is essentially the opposite of the default probability and indicates the probability of not defaulting within a certain timeframe.
Hazard Rate: The hazard rate, also known as the conditional default probability, is the instantaneous rate at which defaults occur, given that no default has occurred until a particular point in time. It measures the likelihood of default per unit of time, provided that the entity has survived up to that point.
These concepts are fundamental in the assessment of credit risk and are widely used in financial modelling, risk management, and credit valuation.

Survival Analysis

Poisson process

A Poisson process is a random process that counts the occurrences of events and the timing of these occurrences within a specified time frame. The time between consecutive events follows an exponential distribution with the parameter λ while these interarrival times are assumed to be uncorrelated with one another. The probability of k events (e.g., defaults) occurring within a given interval can be formulated as follows:

$$\mathbb{P}[N(t+\tau) - N(t) = k] = \frac{(\lambda\tau)^k}{k!} e^{-\lambda\tau}$$

where $N(t_n)$ is the number of defaults up to time t and λ is the rate parameter (hazard rate), and $\tau = t_n - t_{n-1}$ is the interarrival time, where:

$$t_n = \sum_{i=1}^{n} \tau_i$$

As $\tau \to 0$ we find that $\mathbb{P}(k)$ becomes

- For $k = 0$ $\mathbb{P}(0 \text{ Events}) = 1 - \lambda\tau$
- For $k = 1$ $\mathbb{P}(1 \text{ Event}) = \lambda\tau$
- For $k > 1$ $\mathbb{P}(> 1 \text{ Even}) = 0$

Now we may define $F(\tau)$ as the Cumulative Distribution Function (CDF) of τ, and define the probability of no event occurring in the interval τ as the survival function:

$$S(\tau) = 1 - F(\tau)$$

Hence, we can write the surviving probability up to time t and instant defaulting as

$$\mathbb{P}(t < \tau \leq t + dt \mid \tau > t) = (1 - \lambda dt)^{n_t} \lambda dt = \lambda dt + O(dt^2).$$

Then using the relationship from conditional probability, $P(B \mid A) = \frac{P(A \cap B)}{P(A)}$, this equation can be rewritten as

$$\begin{aligned}\lambda &= \lim_{dt \to 0} \frac{\mathbb{P}(t < \tau \leq t + dt)}{dt \mathbb{P}(\tau > t)} \\ &= \lim_{dt \to 0} \frac{[1 - S(t + dt)] - [1 - S(t)]}{dt S(t)} \\ &= \lim_{dt \to 0} \frac{S(t) - S(t + dt)}{dt S(t)} \\ &= -\frac{d\log\, S(t)}{dt}\end{aligned}$$

By solving the above Ordinary Differential Equation for S with boundary condition $S(0) = 1$ (100% survival rate at time 0), we obtain the survival probability function as

$$S(t) = e^{-\int_0^t \lambda(u) du},$$

And the probability of default as

$$F(t) = 1 - S(t) = 1 - e^{-\int_0^t \lambda(u) du}.$$

Default time

From function F(t) we have $U = F(t) = 1 - S(t) = 1 - e^{-\int_0^t \lambda(u) du}$. Thus, we can rearrange it as follows

$$\int_0^t \lambda(u) du = -\log\,(1 - U)$$

Hence, the default time τ can be determined as

$$\tau = \inf\left\{t > 0 \colon \log\,(1 - U) \geq -\int_0^t \lambda(u) du,\right\}$$

Now we can discretize this and set Δt to a constant from 0 to t, we get

$$\tau = \inf\left\{t > 0 \colon \log\,(1 - U) \geq -\Delta t \sum_{i=1}^{t} \lambda_i\right\}$$

We will iterate this inequality to calculate default times by summing up λ_i until the inequality begins to hold, i.e. if the inequality holds after adding λ_n then default occurs between t_{n-1} and t_n.

CDS pricing and Bootstrapping

CDS pricing

The pricing of CDS baskets relies on the principle of no-arbitrage, which necessitates that the present value of premium payments equals the present value of loss given default. In simpler terms, it means that the present value of the spread multiplied by the premium leg is equal to the present value of the default leg. This also results in the inference that the spread is equal to the present value of the default leg divided by the present value of the premium leg.

The expected present value of the premium leg payments can be expressed as below:

$$PV(\text{Premium leg}) = \sum_{i=1}^{N} \text{Spread}_N \text{ Notional } D(0,T_i)S(T_i)\Delta t_i$$

Then, the expected presented value of the default leg payments will be defined in a similar way

$$PV(\text{Default leg}) = \sum_{i=1}^{N} (1-R)D(0,T_i)[S(T_{i-1}) - \mathrm{S(T_i)}],$$

where R is a recovery rate.

Now equating the Premium and Default legs we can compute the fair spread for a N period CDS as

$$\text{Spread}_N = \frac{\sum_{i=1}^{N} (1-R)D(0,T_i)[S(T_{i-1}) - \mathrm{S(T_i)}]}{\sum_{i=1}^{N} \text{ Notional } D(0,T_i)S(T_i)\Delta t_i}$$

Written in terms of the default probability, $F(T_i)$, this can be defined as

$$\text{Spread}_N = \frac{\sum_{i=1}^{N} (1-R)D(0,T_i)[F(T_i) - \mathrm{F(T_{i-1})}]}{\sum_{i=1}^{N} \text{ Notional } D(0,T_i)\big(1-F(T_i)\big)\Delta t_i},$$

This will be the structure used in pricing the k-th to default swap.

Bootstrapping CDS Survival Probabilities and Hazard rates

CDS spreads and discount factors are observed on the market, making the equation for Spread_N applicable for bootstrapping survival/default probabilities, which are not directly observable. In the case of N periods, employ N CDSs with progressively longer maturities, each featuring a spread S_N for the iterative bootstrapping process, resulting in the following equation.

$$S(T_N) = \frac{\sum_{i=1}^{N-1} D(0,T_i)\big[LS(T_{i-1}) - \big(L + \Delta t_i \text{ Spread}_N\big)S(T_i)\big]}{\sum_{i=1}^{N} \text{ Notional } D(0,T_N)\big(L + \Delta t_i \text{ Spread}_N\big)} + \frac{S(T_{N-1})L}{L + \Delta t_N \text{ Spread}_N},$$

Where $L = 1 - R$ is the loss given default.

By leveraging this equation, we can start with $S(T_1)$ and run the bootstrapping process to finish at $S(T_N)$.

Then, using the survival probabilities, we can derive the hazard rate term structure $\lambda_i \ where \ i = 1, \dots, N$. Which can be performed by discretizing formula for survival probability function $S(t) = e^{-\int_0^t \lambda(u)du}$: iterate starting with $S(T_1)$ and moving up to $S(T_N)$ as shown below

$$\lambda_N = \frac{1}{(\mathrm{T_i} - T_{i-1})}\left(-\ln\ S(T_N) - \sum_{i=1}^{N-1} \lambda_i(\mathrm{T_i} - T_{i-1})\right)$$

The results of bootstrapping procedure for our data can be seen below.

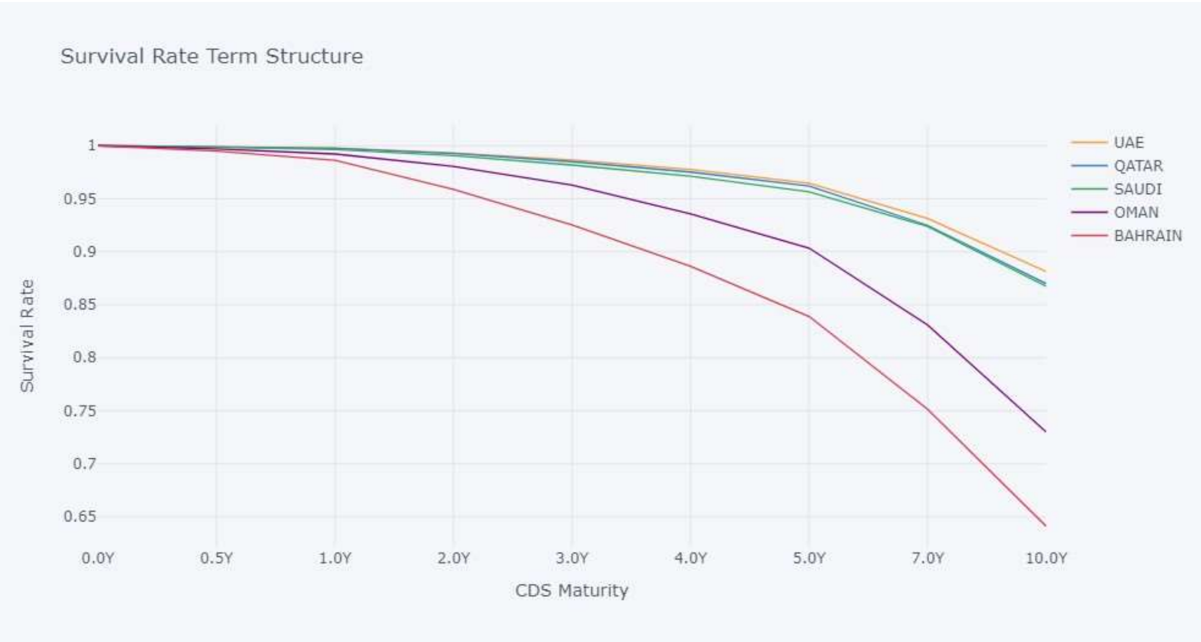


Figure 3. Survival rate term structure

Figure 3 shows us bootstrapped survival rates for all constituents. There is nothing surprising here as UAE, KSA and QATAR again possess the same behaviour while the highest risk of default or the shortest survival rate is associated with Bahrain and then Oman.

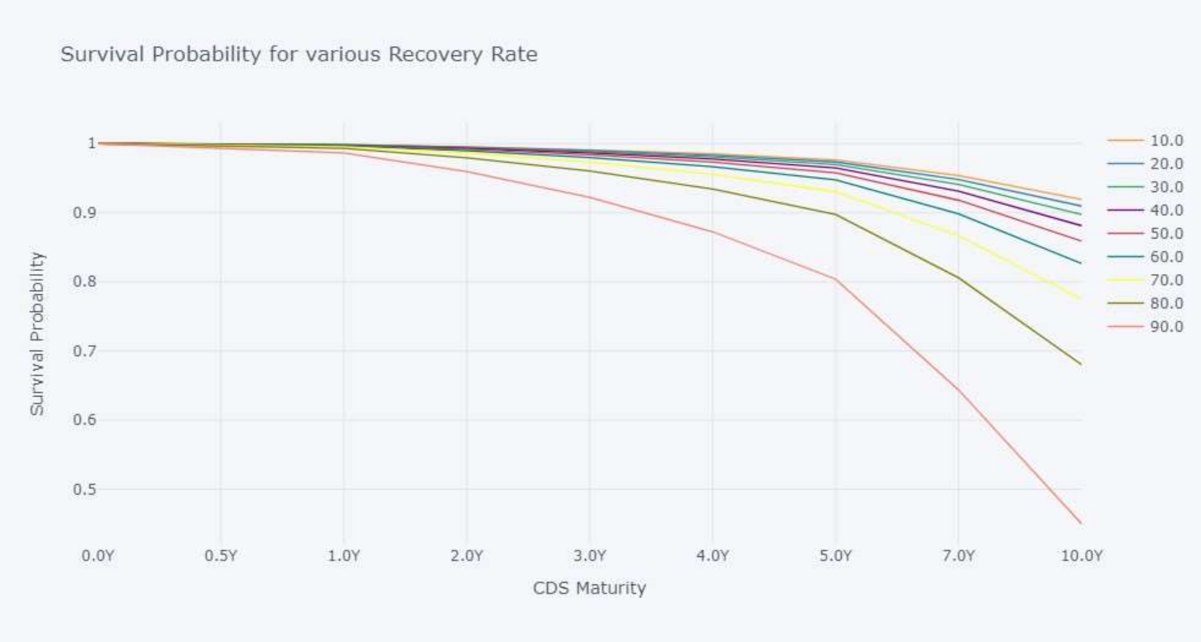


Figure 4. UAE survival rate term structure for different recovery rates

Figure 4 highlights the idea that bootstrapped rates are dependent on the level of recovery assumed for each particular entity: the higher the recovery rate the lower the implied survival probability. This comes straight from the above formulas for spread calculation.

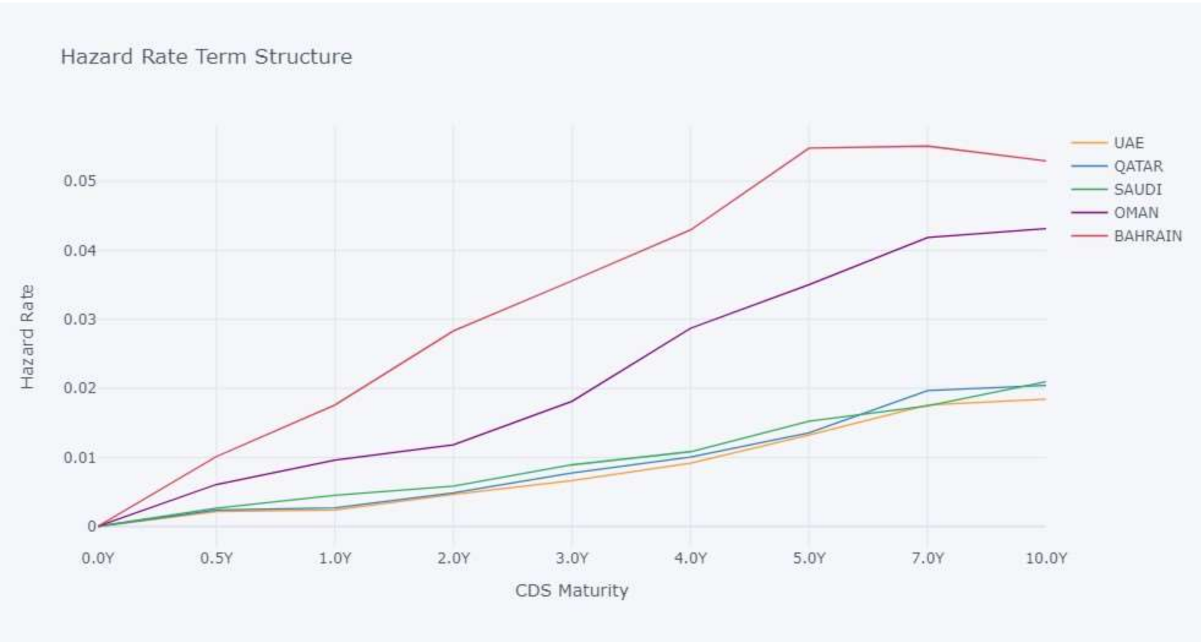


Figure 5. UAE hazard rate term structure for different recovery rates

Figure 5 depicts the hazard rate term structure of all constituents. All the curves except for Bahrain's curve are monotonically increasing. This can be explained by the fact that Bahrain' CDS spread is very high and if it can remain solvent next 5-7 years then there is a good change to be solvent for upcoming years after.

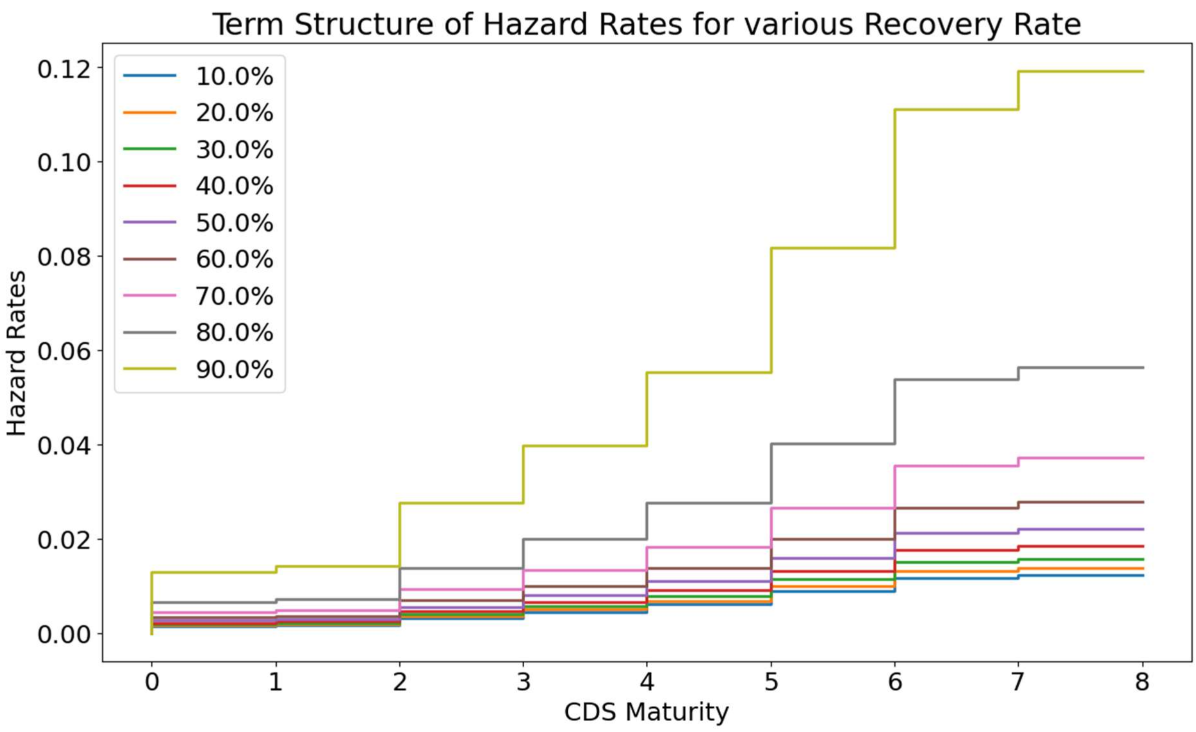


Figure 6. UAE hazard rate term structure for different recovery rates

Figure 6 is the opposite to Figure 4, the higher recovery rate we assume the higher default intensity we may expect.

Copula Overview

A copula is used in the pricing of basket CDS spreads because it allows for the modeling of the joint distribution of default times for multiple entities in the basket. Basket CDS involves multiple reference entities, and their default events are not independent. We noted above that the underlying's' distribution are far from normal ones, so we cannot use Pearson's correlation straight away as we would do for stocks.

Copulas are essential for modeling the dependence structure between random variables, independent of their marginal distributions. Understanding the marginal distributions is crucial for assessing the credit risk associated with each entity separately.

As there are no closed form solutions for marginals, Kernel Density Estimation (KDE) is the way to estimate the joint probability density function of the data. KDE provides a means to estimate the joint distribution needed for copula modeling, enabling the assessment of dependencies, tail behavior, and extreme events.

Definition

For n uniform random variables, $U_1, U_2, \cdots, U_n$, with correlation $\mathbf{\Sigma}$ the joint distribution 'copula function', $\boldsymbol{C}$, is defined as

$$\boldsymbol{C}(U_1, U_2, \cdots, U_n \mid \mathbf{\Sigma}) = \mathbb{P}(u_1 \leq U_1, u_2 \leq U_2, \dots, u_n \leq U_n),$$

A copula function can link univariate marginal distributions to their full multivariate distribution. Sklar's theorem (1973) shows that if $F(x_1, x_2, \cdots, x_m)$ is a joint multivariate distribution function with univariate marginal distribution functions $U_1 = F_1(x_1), U_2 = F_2(x_2), \cdots, U_n = F_n(x_n)$, then there exists a copula function $\boldsymbol{C}(U_1, U_2, \cdots, U_m \mid \boldsymbol{\Sigma})$ such that

$$\boldsymbol{F}(x_1, x_2, \cdots, x_n) = \boldsymbol{C}(F_1(x_1), F_2(x_2), \cdots, F_n(x_n) \mid \boldsymbol{\Sigma}),$$

Where $\boldsymbol{\Sigma}$ is the correlation between the univariate distributions. Thus, copula functions provide a unifying and flexible way to work with multivariate distributions. That is if we know the marginal distributions and their correlations, we can construct the joint distribution through use of an appropriate copula function.

We will be modelling the joint default probabilities using Gaussian and Student T copula functions.

Kernel Smoothing

Kernel density estimation (KDE) is a non-parametric method used to estimate the probability density function of a random variable. It's particularly useful when the underlying distribution of the data is unknown or when the data might not conform to a specific parametric distribution.

In KDE, a kernel function, typically a smooth, symmetric function such as the Gaussian kernel, is placed at each data point, and the sum of these kernels forms the estimated density function.

The bandwidth of the kernel determines the smoothness of the resultant density estimate, with larger bandwidths resulting in smoother estimates and smaller bandwidths capturing more detailed fluctuations in the data. KDE provides a flexible and versatile approach for estimating probability distributions from empirical data, enabling the visualization of the underlying density and facilitating further statistical analyses.

We use KDE from MATLAB software as it allows to precisely assess PDF and smoothen Empirical CDF (ECDF) of marginal distribution. The below chart is produced in Matlab for UAE CDS spreads.

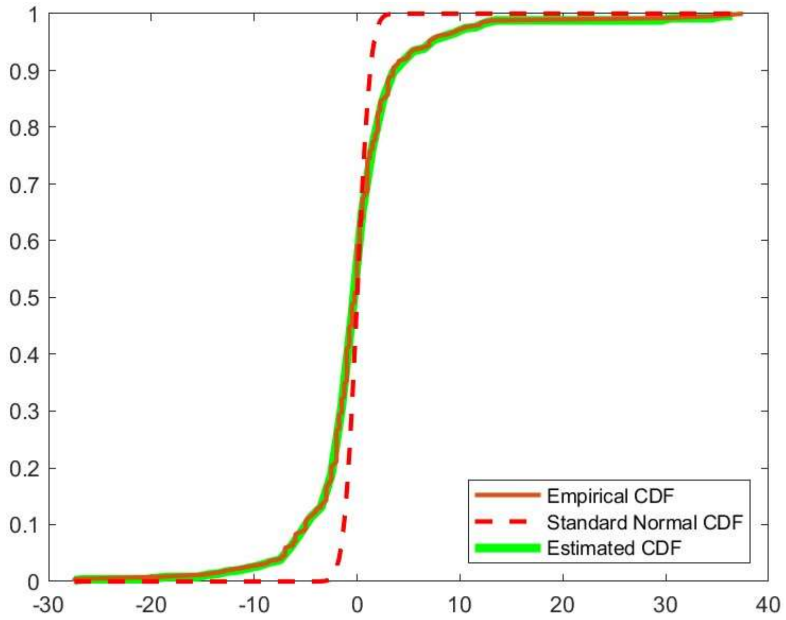


Figure 7. Estimated CDF in Matlab for weekly UAE 5Y CDS differences

Figure 8 below is the pair plot of uniforms drawn in python based on the data that we estimated in Matlab software. Matlab utilizes KDE (kdensity function) to achive the best possible result. The only choise that we make is bandwidth, which by default (Silverman's rule) gives a way too smoothend distribution. The optimal choise would be a bandwith within 0.1 to 0.001 values, although there is no good iterative process to dermine the level that gives best uniforms.
We were changing the bandwidth value while checking uniformity via histograms and the Kolmorogov-Smirnov test. The Matlab code for this test would look as follows:

```
testCDF = makedist('Uniform');
kstest(u1,'CDF',testCDF,'Alpha',0.01);
```

While after we transform uniforms to standard normals, we can do the kstest again by executing the following:

```
n1 = norminv(u1);
kstest(n1,'Alpha',0.01)
```

From the Figure 8 we see that bars of historgam are not even altough change of bandwidth could not change the picture drastically, so we ended up using the level of 0.001.

All Matlab produced unifroms and standard normals were checked by Kolmogorov-Smirnov test where the null hypothesis that the data comes from a respective distibution has been rejected.

Python Quantile transformer

The other way to map an Empirical CDF to uniforms is to use so-called Quantile Transformer (QT). Please note that this method is not based on KDE.

The transformation is applied on each feature independently. Firstly, an estimate of the cumulative distribution function of a feature is used to map the original values to a uniform distribution. The obtained values are then mapped to the desired output distribution using the associated quantile function. Features values of new/unseen data that fall below or above the fitted range will be mapped to the bounds of the output distribution. Note that this transform is non-linear. It may distort linear correlations between variables measured at the same scale but renders variables measured at different scales more directly comparable.

The results of Python quantile transformer are given in Annex 2. The historgams look better, evenly distributed.

Nonetheless, we proceed with Matlab KDE as this method is superior, as major drawback of QT is shrinkage of the data. The biggest concern is that this may affect tail correlations which are crucial for rare-event joint probability estimation.

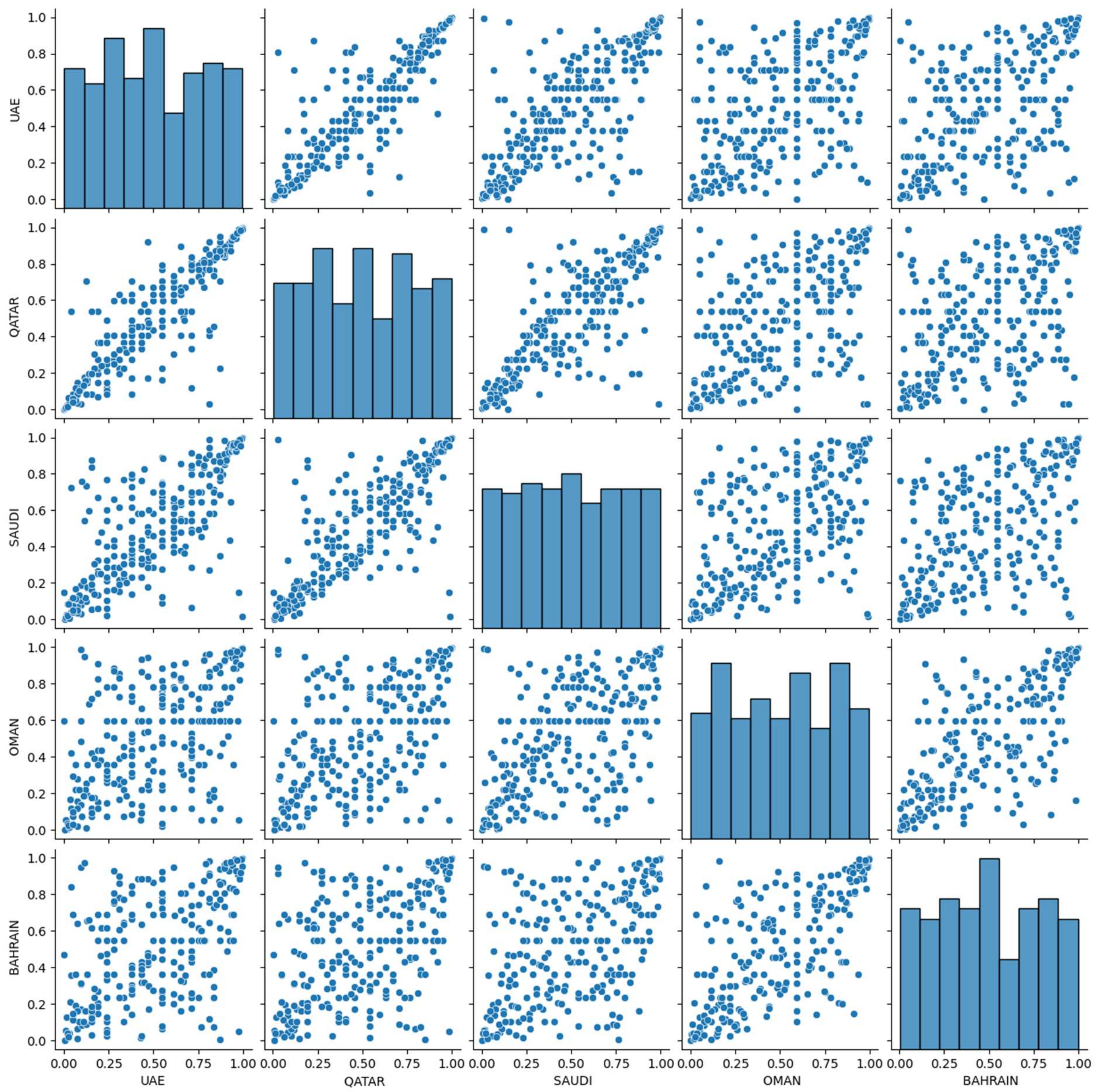


Figure 8. Estimated CDF in Matlab for weekly UAE 5Y CDS differences

Gaussian Copula

The multivariate Gaussian Copula Function can be formulated as below:

$$\boldsymbol{C}(U_1, U_2, \dots, U_n \mid \boldsymbol{\Sigma}) = \boldsymbol{\Phi}_n(\Phi^{-1}(U_1), \Phi^{-1}(U_2), \dots, \Phi^{-1}(U_n) \mid \boldsymbol{\Sigma}),$$

Where $\boldsymbol{\Phi}_n$ is the CDF of a multivariate standard Normal distribution. Then, a n-dimensional Gaussian Copula density (in terms of uniform variables U) is given by

$$\boldsymbol{C}(U_1, U_2, \dots, U_n \mid \boldsymbol{\Sigma}) = \frac{1}{\sqrt{|\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}\boldsymbol{\Phi}^{-1}(\boldsymbol{U})'(\boldsymbol{\Sigma}^{-1}-\boldsymbol{1})\boldsymbol{\Phi}^{-1}(\boldsymbol{U})}$$

Sampling from Gaussian copula:

Using the Gaussian Copula preserves the underlying distribution of the individual random variables but the joint distribution is a multidimensional Gaussian.

1. Compute spectral decomposition (no requirement to be positive definite, refer to CQF Lecture on CDO) of correlation matrix $\hat{\Sigma} = AA^{\top}$
2. Draw an n-dimensional vector of independent standard Normal variables $Z = z_1, \dots, z_n{}^{\top}$.
3. Compute a vector of correlated variables by X = AZ

Map to a correlated uniform vector by $U = \Phi(X)$.

Student T Copula

In real life tail events occur in the financial markets more frequently than are modelled by the normal distribution. The Student T Copula provides a joint distribution which has fatter tails but preserves the same bell-shaped characteristics of the Gaussian. With fatter tails and parameterized with degrees of freedom v the multivariate Student T Copula function can be expressed as

$$\boldsymbol{C}(U_1, U_2, \dots, U_n; v; \boldsymbol{\Sigma}) = \mathbf{T}_v(\mathrm{T}_v{}^{-1}(U_1), \mathrm{T}_v{}^{-1}(U_2), \dots, \mathrm{T}_v{}^{-1}(U_n); \boldsymbol{\Sigma}),$$

Where $\mathbf{T}_v$ and T_v is the multivariate and univariate CDFs for the standard Student's T distribution, with the degrees of freedom parameter v. The n dimensional Student T Copula density (in terms of uniform variables U) is given by

$$\boldsymbol{C}(U_1, U_2, \dots, U_n; v; \boldsymbol{\Sigma}) = \frac{1}{\sqrt{|\boldsymbol{\Sigma}|}} \frac{\Gamma\left(\frac{v+n}{2}\right)}{\Gamma\left(\frac{v}{2}\right)} \left(\frac{\Gamma\left(\frac{v}{2}\right)}{\Gamma\left(\frac{v+1}{2}\right)}\right)^n \frac{\left(1 + \frac{\boldsymbol{T}_v^{-1}(\boldsymbol{U})'\boldsymbol{\Sigma}\boldsymbol{T}_v^{-1}(\boldsymbol{U})}{v}\right)^{-\frac{v+n}{2}}}{\prod_{i=1}^{n}\left(1 + \frac{T_v^{-1}(U_i)^2}{v}\right)^{-\frac{v+1}{2}}}$$

Where $\Gamma(v)$ is a Gamma Function.

Sampling from T copula:

1. Compute spectral decomposition of correlation matrix $\hat{\Sigma} = AA^{\top}$
2. Draw an n-dimensional vector of independent standard Normal variables $Z = z_1, \dots, z_n{}^{\top}$.

3. Drawn an independent chi-squared random variable $s \sim \chi^2_v$. Compute n-dimensional Student's t vector $Y = \frac{Z}{\sqrt{\frac{s}{v}}}$
4. Compute a vector of correlated variables by $\mathrm{X} = \mathrm{AY}$
5. Map to a correlated uniform vector by $U = T_v(X)$ using T dist. CDF.

Estimating Degrees of Freedom (d.f.) of Student T distribution

The degrees of freedom, v, are estimated using the Maximum Likelihood procedure. This is achieved by the two-stage process:

1. Calculate log-likelihood function of Student T Copula density for a set of uniform we computed via KDE in Matlab;
2. Sum the log-likelihood functions and maximize w.r.t v.

This can be summarized as

$$\max_{v} \left\{ \sum_{i=1}^{N_{\text{obs}}} \log\ c(\boldsymbol{U}_i; v; \boldsymbol{\Sigma}) \right\}$$

Figure 9 below visualizes the above-described procedure when we calculate log likelihood for degrees of freedom from 1 to 30 and then find the maximum at d.f. = 3. This value will be used further for all calculations. The correlation matrix used for the computation is based on Kendall's tau correlation we describe in the next section.

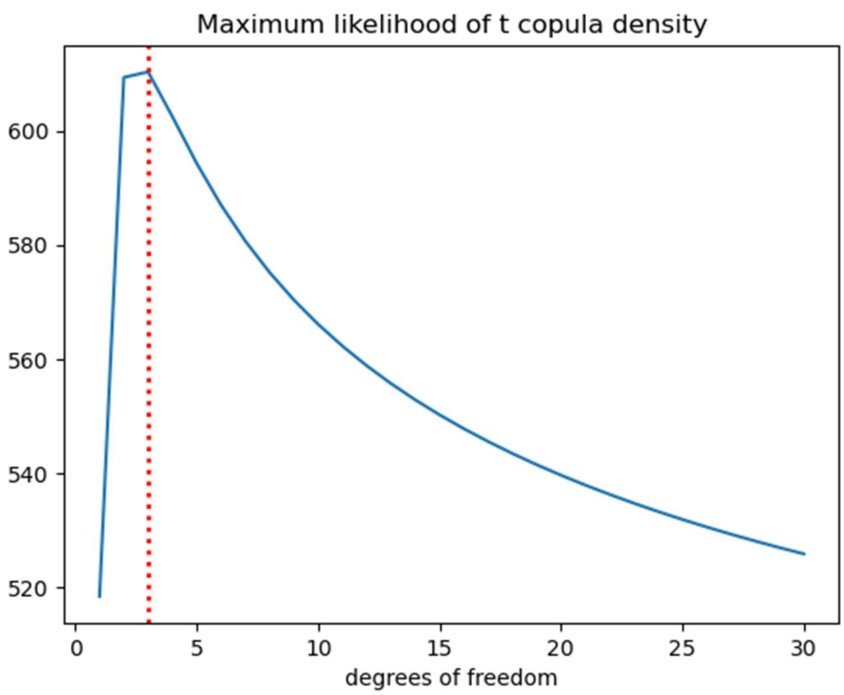


Figure 9. Estimated Maximum log likelihood of T-copula density in order to find best fit for d.f.

Correlation analysis

Correlation is a metric used to quantify the extent to which two variables are related to each other. The correlation coefficient varies between -1.0 and 1.0, indicating the strength and direction of the relationship.

While correlation assesses the degree of association between variables and their co-movement behavior, it does not indicate causation; it doesn't establish whether one variable causes the other, or vice versa, or if the association is influenced by a separate, third factor.

Pearson's Correlation

Pearson's correlation coefficient is the covariance of the two variables divided by the product of their standard deviations.

$$\rho(X_1, X_2) = \frac{\text{Cov}\,(X_1, X_2)}{\sqrt{\mathbb{V}ar(X_1)\text{Var}\,(X_2)}}$$

Below is the correlation matrix derived by this method:

	UAE	QATAR	SAUDI	OMAN	BAHRAIN
UAE	1.000000	0.892030	0.739856	0.549111	0.537390
QATAR	0.892030	1.000000	0.774736	0.536499	0.497217
SAUDI	0.739856	0.774736	1.000000	0.565004	0.544185
OMAN	0.549111	0.536499	0.565004	1.000000	0.722417
BAHRAIN	0.537390	0.497217	0.544185	0.722417	1.000000

Table 1. Pearson's correlation

Spearman's rho

Spearman's rho is equivalent to Pearson's Linear Correlation Coefficient applied to the rankings of the columns X_a and Y_b.
If all the ranks in each column are distinct, the equation simplifies to:

$$rho(a, b) = 1 - \frac{6\sum\ d^2}{n(n^2-1)},$$

where d is the difference between the ranks of the two columns, and n is the length of each column.

Below is the linearized correlation matrix derived by this method:

	UAE	QATAR	SAUDI	OMAN	BAHRAIN
UAE	1.000000	0.873733	0.736480	0.511791	0.510970
QATAR	0.873733	1.000000	0.789894	0.519192	0.470053
SAUDI	0.736480	0.789894	1.000000	0.545302	0.512188
OMAN	0.511791	0.519192	0.545302	1.000000	0.683988
BAHRAIN	0.510970	0.470053	0.512188	0.683988	1.000000

Table 2. Spearman's correlation matrix (linearized)

Kendall's tau

There exist three versions of Kendall's τ. Here we present so-called Kendall's τ_b version. The same is used by both Python and Matlab.

For column X_a in matrix X and column Y_b in matrix Y, Kendall's tau coefficient is defined as:

$$\tau = \frac{2K}{n(n-1)}$$

where $K = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \xi^*\left(X_{a,i}, X_{a,j}, Y_{b,i}, Y_{b,j}\right)$, and

$$\xi^*\left(X_{a,i}, X_{a,j}, Y_{b,i}, Y_{b,j}\right) = \begin{cases} 1 & \text{if} \quad \left(X_{a,i} - X_{a,j}\right)\left(Y_{b,i} - Y_{b,j}\right) > 0 \\ 0 & \text{if} \quad \left(X_{a,i} - X_{a,j}\right)\left(Y_{b,i} - Y_{b,j}\right) = 0. \\ -1 & \text{if} \quad \left(X_{a,i} - X_{a,j}\right)\left(Y_{b,i} - Y_{b,j}\right) < 0 \end{cases}$$

relationship between the columns.

Below is the linearized correlation matrix derived by this method:

	UAE	QATAR	SAUDI	OMAN	BAHRAIN
UAE	1.000000	0.905063	0.767307	0.523773	0.525724
QATAR	0.905063	1.000000	0.830408	0.533447	0.477372
SAUDI	0.767307	0.830408	1.000000	0.557546	0.518068
OMAN	0.523773	0.533447	0.557546	1.000000	0.700398
BAHRAIN	0.525724	0.477372	0.518068	0.700398	1.000000

Table 3. Kendall's tau correlation matrix (linearized)

Spearman's and Kendall's tau correlations were linearalized before sampling form copulas[1]. Linearization facilitates the simulation of joint distributions using copulas. Once the rank correlation coefficients are linearized, the transformed variables can be generated using the copula, and the inverse of the marginal distribution functions can be applied to obtain the final variables.
All three correlation martices exhibit higher calculation between Qatar,UAE and KSA, as well as between Bahrain and Oman. On average Kendall's tau possess slightly higher correlations. We will use this matrix for further sampling from T copula.

Basket CDS pricing and Sensitivity analysis

Pricing Methodology

Pricing a k-th to default swap is done in a similar way as was shown for pricing a CDS. The spread of a k-th to default swap is calculated by equating the expected value of the discounted Premium Leg (PL) with the expected value of the discounted Default Leg (DL), under the risk neutral measure:

$$PL = \text{ Spread }_k \text{ Notional } \Delta t \sum_{i=1}^{m} Z(0, \boldsymbol{t}_i)\big(1 - F_k(\boldsymbol{t}_i)\big)$$

$$DL = (1 - R) \text{ Notional } \sum_{i=1}^{m} Z(0, t_i)\big(F_k(\boldsymbol{t}_i) - F_k(\boldsymbol{t}_{i-1})\big)$$

Assuming the instruments exist over m periods, the fair spread of the k-th to default swap is

$$\text{Spread }_k = \frac{DL}{PL} = \frac{(1-R) \text{ Notional } \sum_{i=1}^{m} Z(0, \boldsymbol{t}_i)\big(F_k(\boldsymbol{t}_i) - F_k(\boldsymbol{t}_{i-1})\big)}{\text{Notional } \Delta t \sum_{i=1}^{m} Z(0, \boldsymbol{t}_i)\big(1 - F_k(\boldsymbol{t}_i)\big)}$$

Here $F_k(\boldsymbol{t}) = F_k(t_1, t_2, \dots, t_n)$ is the unknown function giving the k-th to default probability for the n reference entities. To simplify the calculation process we can use the expected loss function

$$L_k = \text{ Notional } \mathbb{E}[F_k(\boldsymbol{t})]$$

Which then gives the fair spread for the K$^{\text{th}}$ to default swap as

$$\text{Spread }_k = \frac{DL}{PL} = \frac{(1-R)\sum_{i=1}^{m} Z(0, \boldsymbol{t}_i)(L^k{}_i - L^k{}_{i-1})}{\Delta t \sum_{i=1}^{m} Z(0, \boldsymbol{t}_i)(\text{ Notional } - L^k{}_i)}$$

To satisfy the loss distribution expectation, the fair spread calculation needs to be performed using a large number of simulated default times, and averaged across simulations. For each

[1] Copulas: Generate Correlated Samples https://in.mathworks.com/help/stats/copulas-generate-correlated-samples.html

Monte Carlo simulation, where default times $\boldsymbol{\tau} = (\tau_1, \tau_2, \tau_3, \tau_4, \tau_5)$ are randomly sampled, gives the k-th to default swap spread as

$$\text{Spread}_1 = \frac{(1-R)Z(0,\tau_1)\frac{1}{5}}{\tau_1 Z(0,\tau_1)\frac{5}{5}} = \frac{(1-R)\frac{1}{5}}{\tau_1}$$

$$\text{Spread}_2 = \frac{(1-R)Z(0,\tau_2)\frac{1}{5}}{\tau_1 Z(0,\tau_1)\frac{5}{5} + (\tau_2-\tau_1)Z(0,\tau_2)\frac{4}{5}},$$

Further on, we assume that we price 2nd to default without leg removal simplifying the calculation as follows

$$\text{Spread}_2 = \frac{(1-R)Z(0,\tau_2)\frac{1}{5}}{\tau_2 Z(0,\tau_2)\frac{5}{5}}$$

The same is applied for $Spread_3, Spread_4$ and $Spread_5$.

The further implementation of Monte Carlo model is staged as below:

1. Correlated uniform random variables are generated using either Gaussian distribution or Student T distributions (with d.f. v), $\boldsymbol{U}_c = \Phi(\boldsymbol{Z}_c)$ or $\boldsymbol{U}_c = \mathrm{t_v}(\boldsymbol{Z}_c)$. The detailed procedure was descried earlier for both copulas;
2. Correlated uniform random variables are converted to default times $\tau_i = F_i^{-1}(U_{c_i})$;
3. If τ_i is greater than the maturity of the Basket CDS, it is equated to 999 as an indicator as it does not default within maturity of the product. If τ_i is less than 0.25 than we set it to 0.25;
4. An ordered default time vector is constructed using the calculated default times, $\boldsymbol{\tau} = (\tau_1, \tau_2, \tau_3, \tau_4, \tau_5)$;
5. Based on $(\tau_1, \tau_2, \tau_3, \tau_4, \tau_5)$ calculate the discounted values of premium and default legs as per above formulas;
6. The above steps are performed a large number of times. Average premium and default legs across simulations separately. Calculate the fair spread s.

Results and Validation

Following the procedure specified in the previous section, the k-th to default Basket CDS spreads have been computer for both Gaussian and T copula and are presented below (50 000 iterations performed):

in bps	Gaussian copula	T copula
1st to default	59.6	55.3
2nd to default	23.0	22.9
3rd to default	8.4	9.5
4th to default	3.7	5.2
5th to default	1.5	2.8

Table 4. An estimation of Basket CDS spreads (50 000 iterations)

The fair spread for kth-to-default Basket CDS should be less than k-1 to default. Why?

This is due to a very simple reason - If the kth default occurs, it means that the k-1 to default has already defaulted. Let's imagine there is only 2 names in a basket, then it's fairly easy to say that the probability of at least one default occurring is higher compared to the probability of at least two defaults. Consequently, there would be less risk associated with kth-to-default Basket CDS rather than for a kth-1 to default instrument.

This pricing behaviour is observed consistently for both Gaussian and Student t Copula sampling. In both cases, the fair spread for the 1st-to-default swap is the highest, and the spread for the 5th-to-default swap is the lowest among the instruments.

At the same time, we see in interesting observation, 1st to default is cheaper for T copula, 2nd to default is about the same value, and the rest three (3rd,4th,5th) – to default are pricier under T copula. The answer to this observation is that T Copula possesses stronger co-movement, which means that there is higher probability of joint defaults.

For the purpose of comfort, starting from here and onwards we will call 1st to default swap – equity tranche, 2nd – mezzanine tranche, 3rd to 5th – senior tranches. These names are commonly used for CDO and MBS products which structure is somewhat similar to the CDS basket.

Price Sensitivity Analysis

Here we are going to investigate price sensitivity with respect to changes in credit spreads, correlations and hazard rates.

- Correlation matrices were stressed by multipliers from 0 to 1.5 while keeping all diagonal members equal to 1, and non-diagonal below 0.95 (higher values cause errors in the spectral decomposition);
- Credit spreads were shifted approximately. We would shift hazard rate curve by parallel shocks of +25bps to +400bps and then recalculate survival rates accordingly
- Effect of recovery rate change is assessed on a range from 0.1 to 0.9.

Below you can see these effects visualized and commented for better experience and navigation. However, the detailed results and figures are also available in Annex 3.

Correlation matrix effect

The correlation effect for both copulas is rather similar: The lower correlations between constituents the higher price of equity trance and the lesser prices of other trances.

The higher correlations between tranches the closer to each other prices become.

Mezzanine tranche is barely affected by change in correlation levels: its value increased slightly for both copulas.

Senior tranches undergo a significant increase as correlation increases.

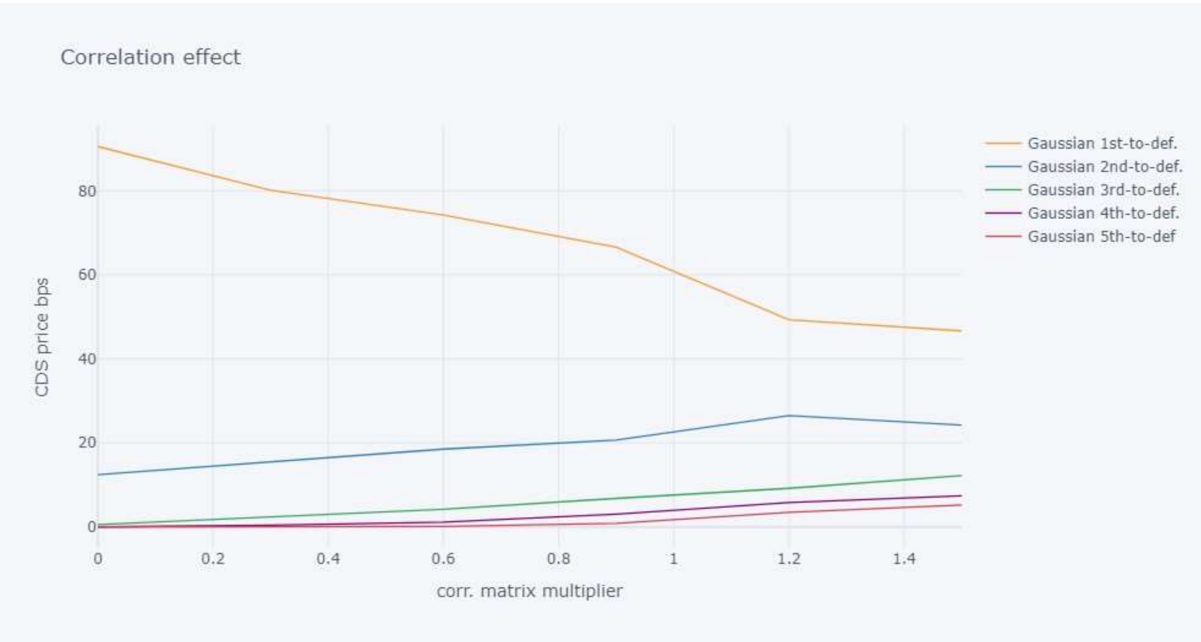


Figure 10. Correlation effect of pricing of Basket CDS (Gaussian copula)

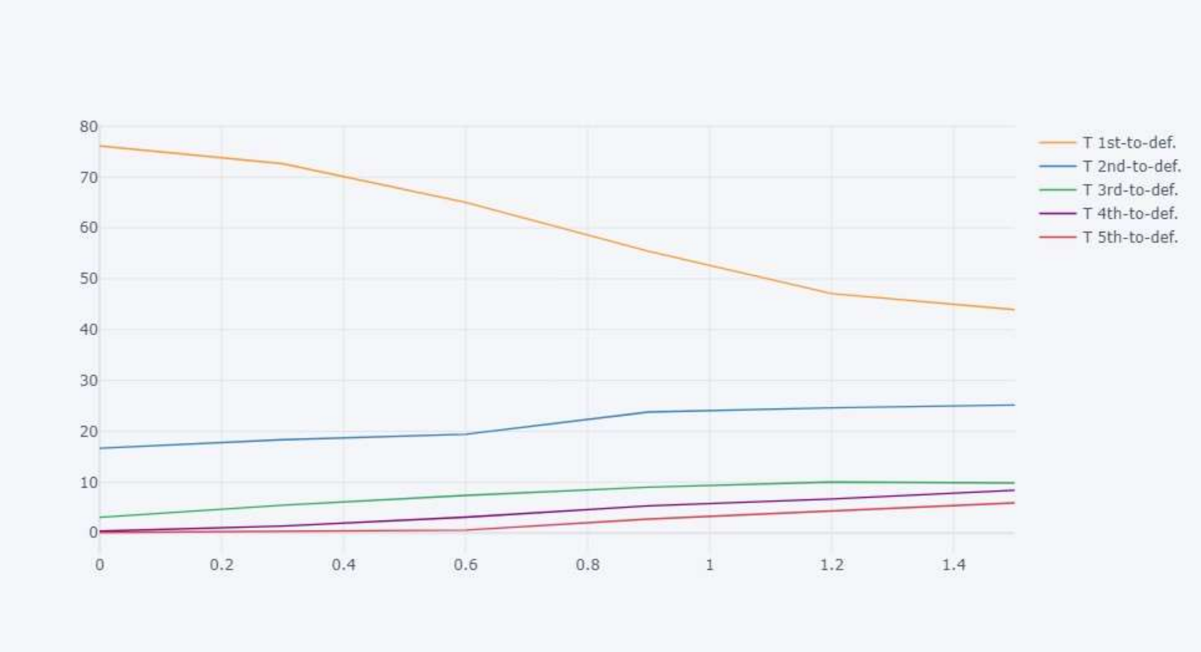


Figure 11. Correlation effect of pricing of Basket CDS (T copula)

Credit spread effect (Shift all constituents)

For both copulas we see a somewhat linear effect from hazard rate increase (survival rate decrease). What is worth noting is that the equity tranche has a greater slope (the most sensitive), which can be explained mathematically as the probability of a single default is increasing faster than the probability of joint default.

T copula has lesser slopes which may be caused by a higher correlation of default.

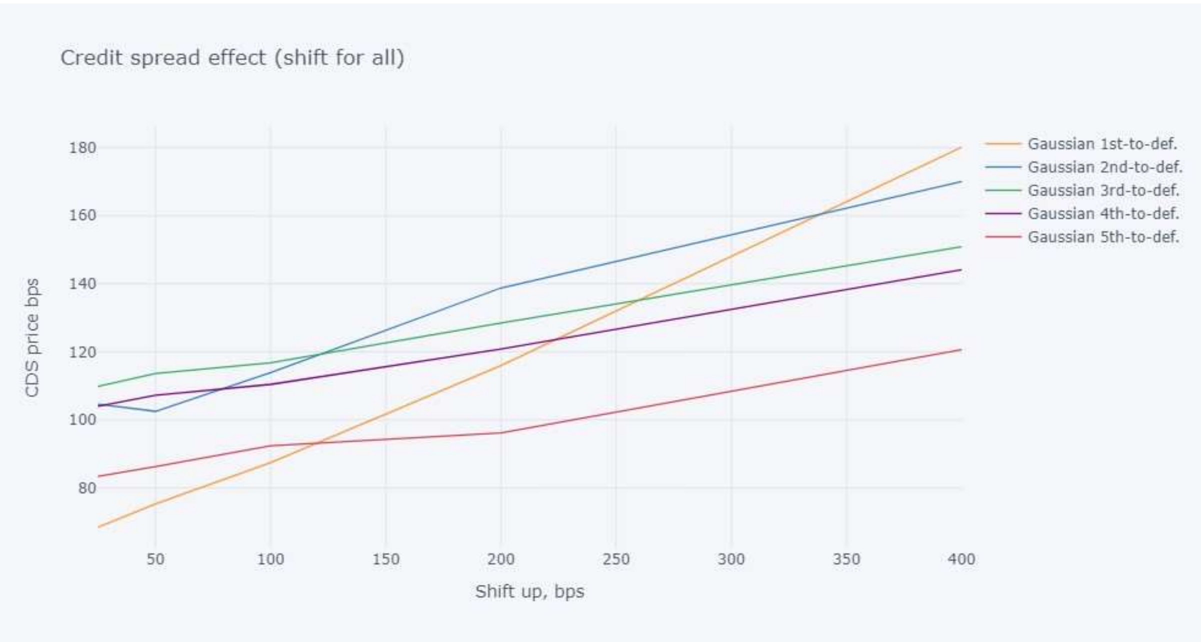


Figure 12 Credit spread effect of pricing of Basket CDS (Gaussian copula)

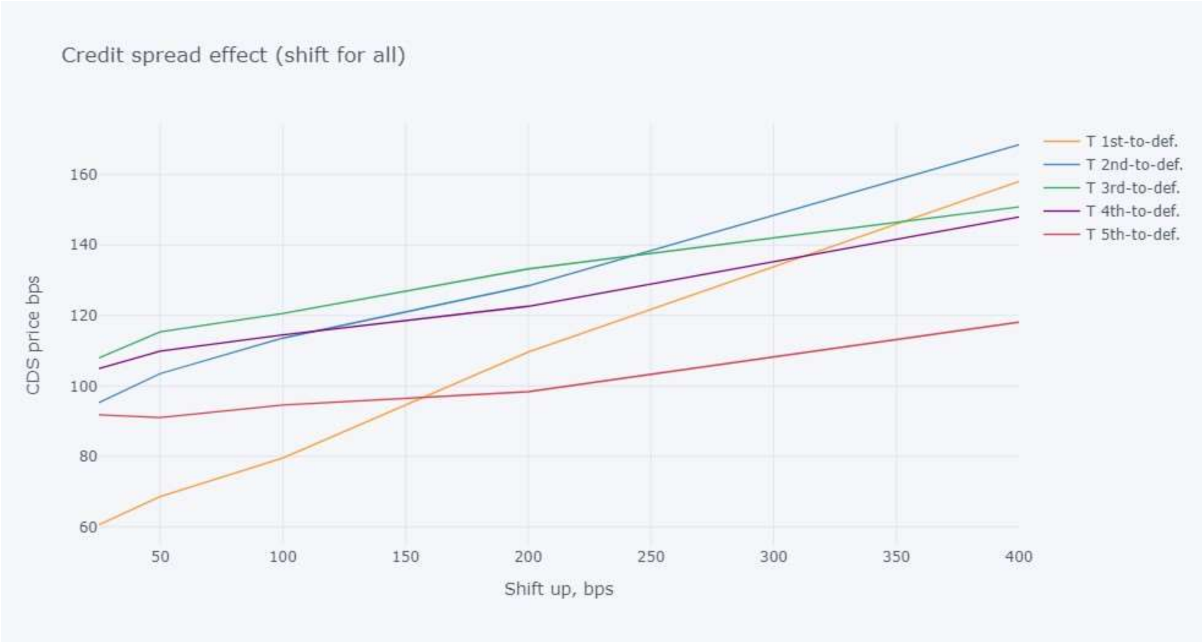


Figure 13. Credit spread effect of pricing of Basket CDS (T copula)

Credit spread effect (Shift a single constituent)

Now we stress credit spread of a single name in the basket (UAE in theis case). Here again, the behavior is rather similar for both copulas. The equity tranche is increasing drastically given that the probability of occurring of one default is increasing. The mezzanine tranche is growing moderately while senior tranches barely pay attention to a singe name increase. If one constituent is having hard time, the default of 3^{rd}, 4^{th} and 5^{th} constituents are still unlikely.

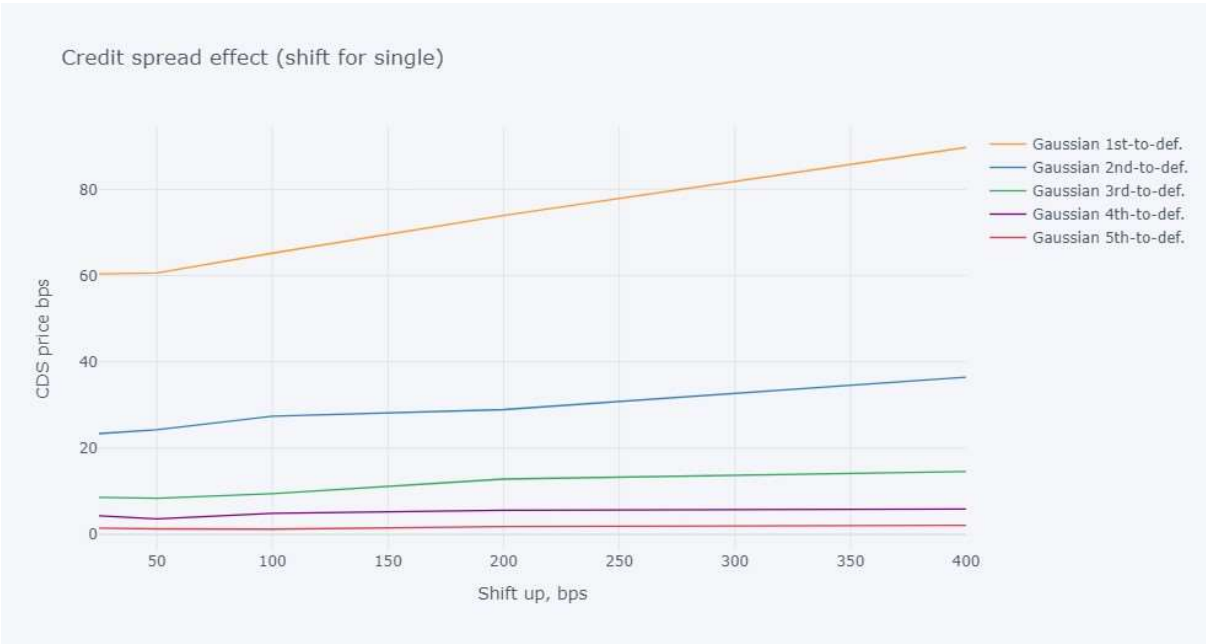


Figure 14. Credit spread effect of pricing of Basket CDS (Shift a single name, Gaussian copula)

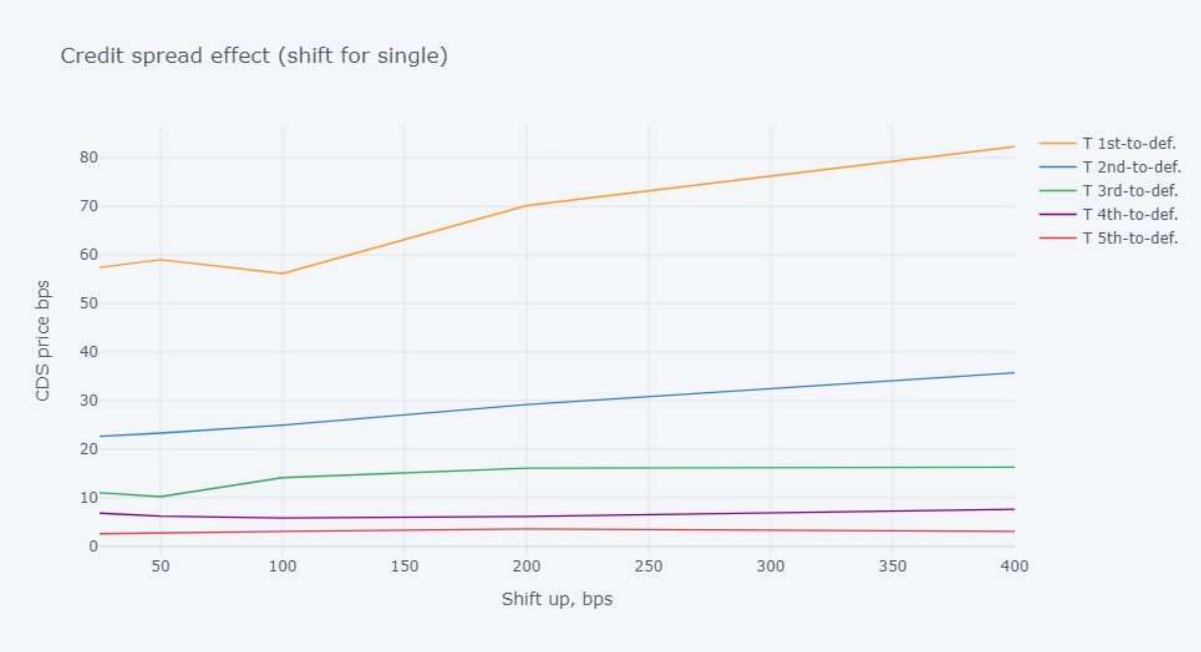


Figure 15. Credit spread effect of pricing of Basket CDS (Shift a single name, Gaussian copula)

Recovery rate effect

There is no surprise in changing recovery rate level for both copulas: the higher recovery the less price of the instruments. The equity tranche is the most sensitive, followed by the mezzanine trance and then senior tranches.

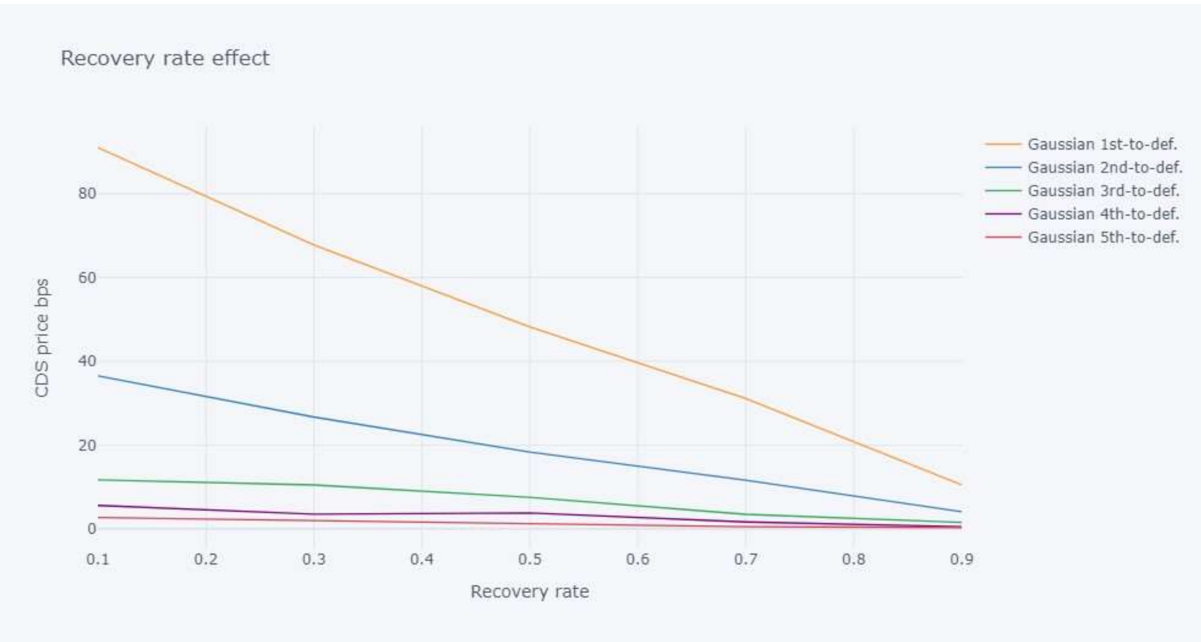


Figure 16. Recovery rate effect (Gaussian copula)

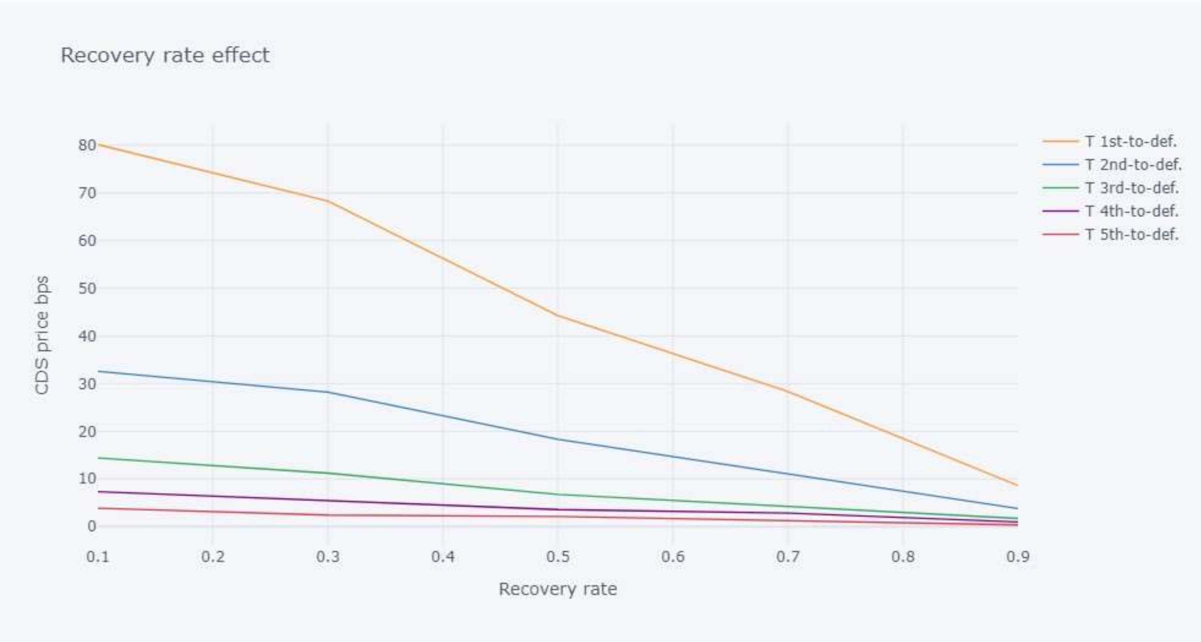


Figure 17. Recovery rate effect (Gaussian copula)

Variance reduction techniques

Low Discrepancy (Sobol) Sequences

Conventional Monte Carlo simulation uses sequences of pseudo-random numbers, but their convergence is often slow. To expedite convergence, low-discrepancy sequences, also known as quasi-random sequences, are utilized, as they efficiently and evenly cover the possibility space. Various forms of low-discrepancy sequences exist, including the Van der Corput, Halton, and Sobol's sequences.

In this paper, we apply 'Sobol_seq' library in python. The results of generating two standard normal sequences can be seen below. We acknowledge that algorithm works as it produces evenly distributed values as on the Figure 18.

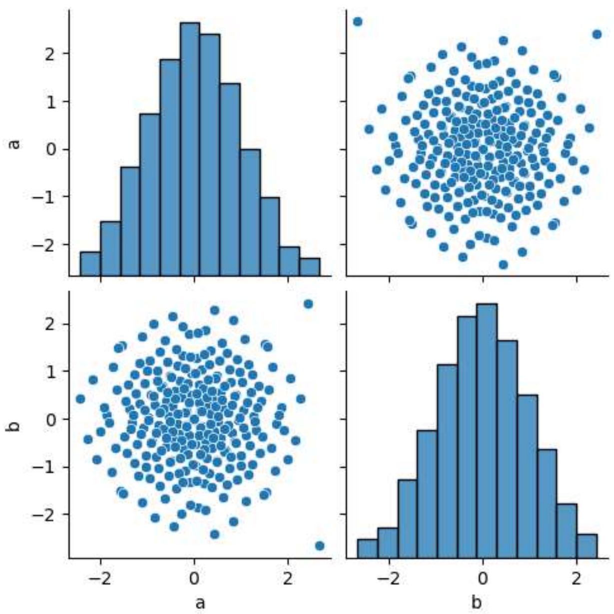


Figure 18. Scatter plot and histograms for sobol sequeces generated in python

The results of pricing with Sobol sequences are given in the table 5. The results are very close to the plain Monte Carlo approach, although our main interest is to see whether it reduces variance of the pricing.

in bps	Gaussian copula	T copula
1st to default	59.8	55.5
2nd to default	22.7	23.0
3rd to default	8.3	9.4
4th to default	3.7	5.3
5th to default	1.5	2.6

Table 5. Results of pricing Basket CDS by using Sobol sequences (50 000 iterations)

From the Figure 19 and Figure 20 we can see that use of Sobol sequences is not a 'game changer'. The Sobol approach gives a slightly better pricing in terms of the price variance, especially for number of iterations from 1 to 4 thousand, while more or less sufficient results for plain Monte Carlo estimates starts from 4 thousand iterations. Overall, we see that Sobol sequences are superior compared to the pseudo RNG, however to tackle rare-event events we have to focus on particular parts of multivariate distribution. The next level model could be use of importance sampling which is briefly covered below but not yet sufficiently implemented below.

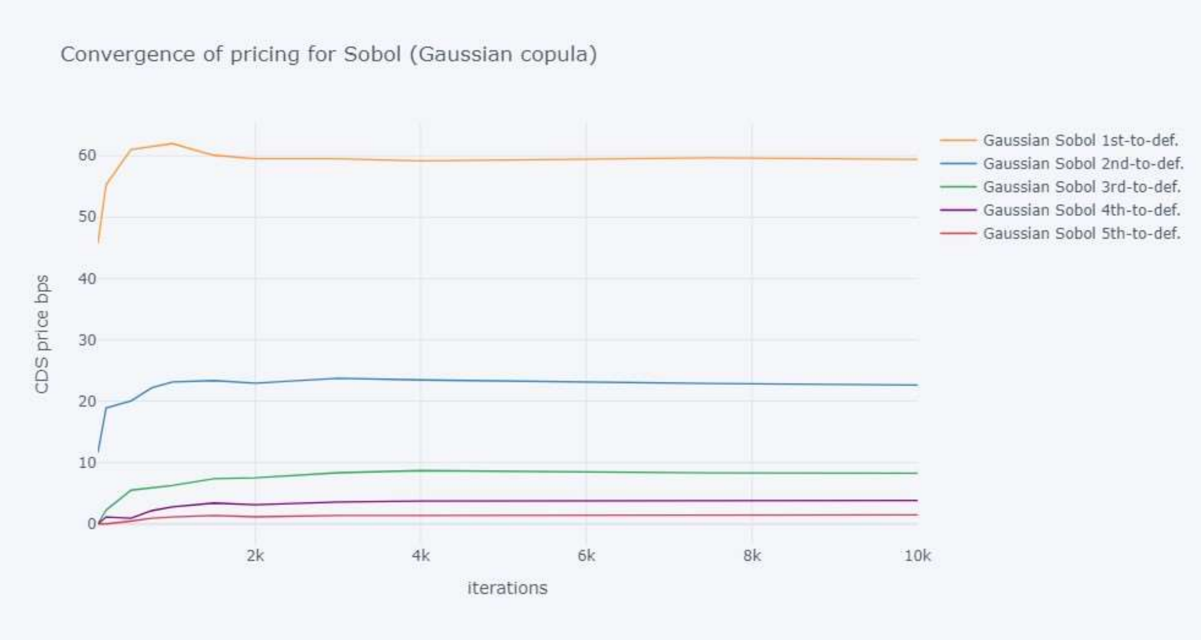


Figure 19. Convergence of pricing with Sobol sequences (Gaussian copula)

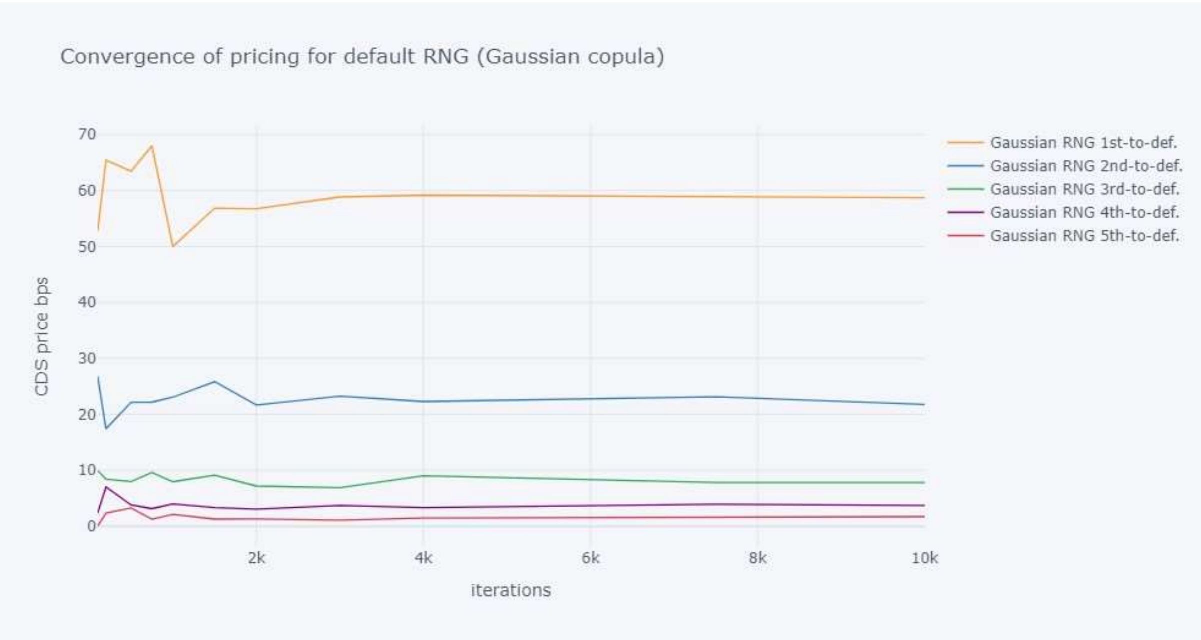


Figure 20. Convergence of pricing with default RNG (Gaussian copula)

Importance Sampling: Modified JK Algorithm

This paper endeavors to implement an importance sampling algorithm, originally developed by Joshi and Kainth and subsequently modified by Glasserman and Chen. The implementation done in Python adheres to the instructions provided by these authors. However, the prices computed by this method diverge significantly from the plain Monte Carlo approach. While the specific results are not presented in this paper, the *implementation details are available in the accompanying code. The Author would be more than grateful if a reviewer can advise on fixing the issues and completing of algorithm. In the meantime, the Author will try to finish on their own. To provide context and elucidate on the methodology, a brief description from the article by Glasserman and Chen (2008) is presented below.*

Here is an informal explanation of the JK method applicable to a first-to-default swap. The first asset is attributed a likelihood of 1/N to default by time T, irrespective of its actual default probability. If it does default, the importance sampling is deactivated, and the simulation progresses conventionally. Alternatively, should the first asset not default, the second asset is assigned a default probability of 1/(N-1). Subsequent assets have their default probabilities incremented in a sequence, 1/(N-2), 1/(N-3), and so forth until a default arises, at which point the importance sampling ceases. If the Nth asset is reached without any prior defaults, it is endowed with a default probability of 1/(N-(N-1)) = 1 to ensure at least one default occurs on every path. The approach for an nth-to-default swap is analogous. When the algorithm reaches the ith asset and j<n defaults have occurred at that point, the ith asset is assigned a default probability of (n-j)/(N-i+1), ensuring that at least n defaults arise on every path. Once n defaults have transpired, the method reverts to the original sampling procedure. Each step in this process is vital to the coherent application of the JK method for first-to-default and nth-to-default swap scenarios.

For the IS algorithm, define default indicator variables $Y_i = I(\tau_i \leqslant T), i = 1, \dots, N$. The conditional default probabilities (see calculation in Chen & Glasserman 2008) are then given by $p_1 = F_1(T)$ and

$$p_i = \mathbf{P}(Y_i = 1 \mid \tau_1, \dots, \tau_{i-1}) = F_i(T \mid \tau_1, \dots, \tau_{i-1}), \quad i = 2, \dots, N.$$

These will be replaced by new probabilities $\tilde{p}_i, i = 1, \dots, N$, with $\tilde{p}_1$ fixed and $\tilde{p}_i$ a function of $Y_1, \dots, Y_{i-1}$. The particular choice used by Joshi and Kainth (2004) can be expressed as

$$\tilde{p}_1 = n/N,\ \tilde{p}_i = \begin{cases} \left(n - \sum_{j=1}^{i-1} Y_j\right)/(N-i+1), & \sum_{j=1}^{i-1} Y_j < n, \\ p_i, & \sum_{j=1}^{i-1} Y_j \geqslant n. \end{cases}$$

Here, the sum over Y_j simply counts the number of assets defaulting by time T.

The JK procedure can be viewed as generating the pairs $(Y_1, \tau_1), \dots, (Y_N, \tau_N)$ recursively using independent $V_1, \dots, V_N$, uniformly distributed between zero and one. Each replication works as follows:

Sampling Procedure. For each $i = 1, \dots, N$,

Step 1. Generate V_i uniformly over (0,1).

Step 2. Set $Y_i = I(V_i \leqslant \tilde{p}_i)$,

$$U_i = \begin{cases} p_i V_i / \tilde{p}_i & \text{if } Y_i = 1, \\ p_i + [(1-p_i)(V_i - \tilde{p}_i)/(1-\tilde{p}_i)] & \text{if } Y_i = 0, \end{cases}$$

and

$\tau_i = F_i^{-1}(U_i \mid \tau_1, \dots, \tau_{i-1})$.

(In the case $i = 1$, set $\tau_1 = F_1^{-1}(U_1)$.)
Step 3. Calculate the weight:

$$L_i = \begin{cases} p_i/\tilde{p}_i & \text{if } Y_i = 1, \\ (1 - p_i)/(1 - \tilde{p}_i) & \text{if } Y_i = 0. \end{cases}$$

Once $\tau_1, \dots, \tau_N$ have been generated, evaluate $V(\tau_1, \dots, \tau_N)$ and return the weighted estimate $V(\tau_1, \dots, \tau_N)L$, with $L = L_1 L_2 \cdots L_N$ the weight for the path.

References

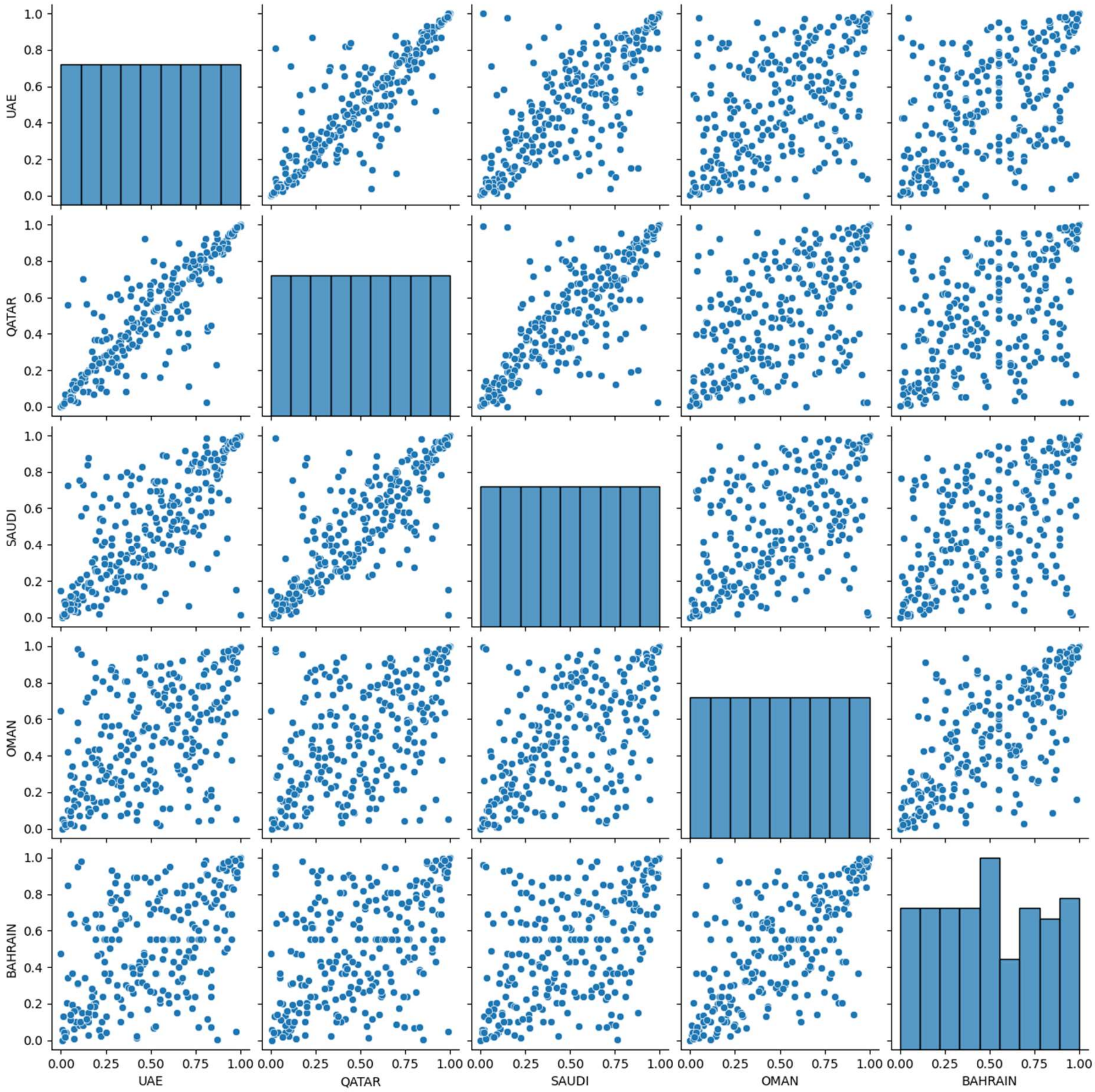
1. Sklar, A. (1973). Random variables, joint distribution functions, and copulas. *Kybernetika*, *9*(6), 449-460.
2. Joshi, M. S., & Kainth, D. (2004). Rapid and accurate development of prices and Greeks for n th to default credit swaps in the Li model. *Quantitative Finance*, *4*(3), 266.
3. Chen, Z., & Glasserman, P. (2008). Fast pricing of basket default swaps. *Operations Research*, *56*(2), 286-303.
4. Glasserman, P. (2004). *Monte Carlo methods in financial engineering* (Vol. 53, pp. xiv+-596). New York: springer.
5. Jäckel, P. (2002). *Monte Carlo methods in finance* (Vol. 5). John Wiley & Sons.
6. Sehra, A. (2014). Pricing Kth to Default Basket CDS.
7. Joe, S., & Kuo, F. Y. (2008). Notes on generating Sobol sequences. *ACM Transactions on Mathematical Software (TOMS)*, *29*(1), 49-57.
8. CQF lectures, workshops, tutorials
9. CQF Python labs 16 & 17
10. CQF additional notes provided by Richard Diamond
11. Copulas: Generate Correlated Samples: https://in.mathworks.com/help/stats/copulas-generate-correlated-samples.html
12. Using Extreme Value Theory and Copulas to Evaluate Market Risk https://in.mathworks.com/help/econ/using-extreme-value-theory-and-copulas-to-evaluate-market-risk.html

Annex

Annex 1. CDS term structure

COUNTRY	BBG ID	CURRENCY	TERM	TICKER	CONTRIBUTOR	BID(bps)	ASK(bps)
QATAR	307687	USD	6mo	CT990879	CMAN	10.21	18.13
QATAR	307687	USD	1yr	CQTA1U1	CMAN	11.69	18.54
QATAR	307687	USD	2yr	CQTA1U2	CMAN	18.62	25.37
QATAR	307687	USD	3yr	CQTA1U3	CMAN	26.58	32.99
QATAR	307687	USD	4yr	CQTA1U4	CMAN	34.05	39.86
QATAR	307687	USD	5yr	CQTA1U5	CBIN	43.278	46.851
QATAR	307687	USD	7yr	CQTA1U7	CMAN	59.88	67.19
QATAR	307687	USD	10yr	CQTR1U10	CMAN	73.55	83.37
SAUDI	307693	USD	6mo	CT990871	CMAN	12.97	18.89
SAUDI	307693	USD	1yr	CT965291	CMAN	18.69	24.13
SAUDI	307693	USD	2yr	CT965295	CMAN	26.11	30.08
SAUDI	307693	USD	3yr	CT965299	CMAN	34.17	38.26
SAUDI	307693	USD	4yr	CT965303	CMAN	40.98	44.93
SAUDI	307693	USD	5yr	CT965307	CBIN	50.146	53.48
SAUDI	307693	USD	7yr	CT965315	CMAN	62.27	68.23
SAUDI	307693	USD	10yr	CT965327	CMAN	77.6	83.5
BAHRAIN	171957	USD	6mo	CT990867	CMAN	43.95	77.71
BAHRAIN	171957	USD	1yr	CT965095	CMAN	67.7	98.46
BAHRAIN	171957	USD	2yr	CT393409	CMAN	116.29	135.31
BAHRAIN	171957	USD	3yr	CT965099	CMAN	146.23	161.69
BAHRAIN	171957	USD	4yr	CT965103	CMAN	173.91	182.43
BAHRAIN	171957	USD	5yr	CT393413	CBIN	198.282	211.821
BAHRAIN	171957	USD	7yr	CT965111	CMAN	229.46	243.98
BAHRAIN	171957	USD	10yr	CT393417	CMAN	248.6	268.18
UAE	12257737	USD	6mo	CX967360	CMAN	10.6	15.5
UAE	12257737	USD	1yr	CX855687	CMAN	11.49	15.9
UAE	12257737	USD	2yr	CX855691	CMAN	18.52	22.61
UAE	12257737	USD	3yr	CX855695	CMAN	24.59	28.81
UAE	12257737	USD	4yr	CX855703	CMAN	30.93	35.7
UAE	12257737	USD	5yr	CX855707	CBIN	39.653	43.89
UAE	12257737	USD	7yr	CX855751	CMAN	55.35	60.53
UAE	12257737	USD	10yr	CX855763	CMAN	67.53	74.99
OMAN	209923	USD	6mo	CT991527	CMAN	25.82	47.22
OMAN	209923	USD	1yr	CT991531	CMAN	32.06	61.82
OMAN	209923	USD	2yr	CT991535	CMAN	50.29	67.09
OMAN	209923	USD	3yr	CT991539	CMAN	66.93	82.47
OMAN	209923	USD	4yr	CT991543	CMAN	93.3	101.52
OMAN	209923	USD	5yr	CT991547	CBIN	112.681	122.81
OMAN	209923	USD	7yr	CT991551	CMAN	143.97	157.94
OMAN	209923	USD	10yr	CT991555	CMAN	167.82	187.94

Annex 2. Uniforms from Quantile Transformer from Python SciPy



Annex 3. Price sensitivity analysis

Annex 3.1. Correlation matrix analysis

Multiplier	Gaussian 1st-to-def.	Gaussian 2nd-to-def.	Gaussian 3rd-to-def.	Gaussian 4th-to-def.	Gaussian 5th-to-def	T 1st-to-def.	T 2nd-to-def.	T 3rd-to-def.	T 4th-to-def.	T 5th-to-def.
-	90.51	12.44	0.57	-	-	76.13	16.67	3.11	0.36	0.12
0.30	80.15	15.47	2.32	0.40	-	72.64	18.35	5.45	1.36	0.36
0.60	74.19	18.51	4.17	1.15	0.12	65.02	19.42	7.40	3.12	0.57
0.90	66.59	20.67	6.77	3.05	0.81	55.42	23.80	9.03	5.33	2.75
1.20	49.30	26.46	9.19	5.82	3.45	47.11	24.61	10.03	6.71	4.31
1.50	46.64	24.22	12.24	7.37	5.19	43.94	25.20	9.85	8.40	5.93

Annex 3.2. Credit spread effect (Shift all constituents)

+bps shift	Gaussian 1st-to-def.	Gaussian 2nd-to-def.	Gaussian 3rd-to-def.	Gaussian 4th-to-def.	Gaussian 5th-to-def.	T 1st-to-def.	T 2nd-to-def.	T 3rd-to-def.	T 4th-to-def.	T 5th-to-def.
25.00	68.40	104.60	109.83	103.95	83.38	60.54	95.28	107.96	104.91	91.82
50.00	75.31	102.51	113.62	107.26	86.23	68.56	103.49	115.33	109.86	91.00
100.00	87.45	113.91	116.74	110.42	92.39	79.52	113.58	120.62	114.52	94.61
200.00	115.97	138.77	128.43	120.82	96.19	109.66	128.45	133.24	122.62	98.37
400.00	180.16	170.06	150.87	144.16	120.65	158.10	168.53	150.86	147.96	118.11

Annex 3.3. Credit spread effect (Shift all constituents)

+bps shift	Gaussian 1st-to-def.	Gaussian 2nd-to-def.	Gaussian 3rd-to-def.	Gaussian 4th-to-def.	Gaussian 5th-to-def.	T 1st-to-def.	T 2nd-to-def.	T 3rd-to-def.	T 4th-to-def.	T 5th-to-def.
25.00	60.41	23.37	8.59	4.33	1.48	57.38	22.63	11.04	6.81	2.58
50.00	60.60	24.25	8.37	3.59	1.29	58.99	23.30	10.22	6.21	2.69
100.00	65.20	27.42	9.46	4.90	1.22	56.15	24.95	14.13	5.81	3.06
200.00	73.93	28.93	12.82	5.57	1.82	70.13	29.16	16.07	6.17	3.56
400.00	89.71	36.45	14.59	5.89	2.11	82.26	35.71	16.28	7.63	3.04

Annex 3.4. Recovery rate effect

Recovery rate	Gaussian 1st-to-def.	Gaussian 2nd-to-def.	Gaussian 3rd-to-def.	Gaussian 4th-to-def.	Gaussian 5th-to-def.	T 1st-to-def.	T 2nd-to-def.	T 3rd-to-def.	T 4th-to-def.	T 5th-to-def.
0.10	91.07	36.54	11.65	5.55	2.67	80.14	32.56	14.39	7.32	3.88
0.30	67.79	26.67	10.45	3.49	1.83	68.28	28.21	11.22	5.48	2.42
0.50	48.19	18.29	7.48	3.76	1.18	44.25	18.33	6.73	3.57	2.15
0.70	31.09	11.56	3.43	1.60	0.47	28.33	11.10	4.30	2.82	1.24
0.90	10.47	4.03	1.46	0.47	0.28	8.62	3.82	1.71	0.99	0.36