Chapter 2

Fair Spread for Credit Portfolio

2.1 Spread as Average of PL and DL

Premium Accrual Analytical expression for spread computation with discrete-time accruals becomes complicated for 2nd-to-default and above. Remember that while multiple defaults are possible, only one is protected from.

- Continue with 2nd-to-default example and assuming that 1st default occurs before Year 2 end $\tau_1 < 2$, and 2nd default occurs in Year 3 $\tau_2 < 3$, then accrual can look like

$$s = \frac{(1-R)Z(0,\tau_2) \times \frac{1}{5}}{[Z(0,t_1)\Delta t_1 + Z(0,\tau_1)\delta t_{k=1}] \times \frac{5}{5} + [Z(0,t_2)(t_2-\tau_1) + Z(0,\tau_2)\delta t_{k=2}] \times \frac{4}{5}}$$

This and workshop implementations assume there is a default. If there is no default, keep accruing the premium until the 5Y end (expiry). This will only be one simulated value of PL.

Implied curve fitting The generalised coding solution is: create a running variable that accumulates PL at each time step $dt = 0.01$. For that you will need discount factors for each time step (a fitted discounting curve). This approximates continuous-time accrual – if we calculate the PL to default time exactly then, we do not need to adjust DL for accrued premium.

The alternative to near-continuous accruals are quarterly, semi-annual, annual in arrears (end of the year), or upfront payment of the premium. Each of these choices will modify PL (the denominator), often leading to substantial simplification. **Code any suitable choice.**

Averaging PL and DL **Remember** to calculate the averages of DL and PL across simulations separately. Then calculate the spread once as

$$s = \frac{< DL >}{< PL_{\not\$} >}$$

You are likely to find out that *a few early defaults* (e.g., from one day to 6M) are likely to bring PL average down and over-estimate the spread (introduce a bias). Therefore, you might impose a restriction that the first default can only occur (be settled) after first quarter, $\min \tau_k = 0.25$ or implement premium accrual for the whole year (paid in arrears). Here, you can also try variance reduction and importance sampling techniques, such as giving higher weight to simulations that have defaults in later years. Usual MC weights each simulation as $\frac{1}{T}$.

It is not advised to mix data for 1st, 2nd-, 3rd-to-default pricing for clarity reason.

These recommendations aim to improve the convergence of fair spread.

However, it has been reported up to $600,000$ rounds of Monte-Carlo were required for the spread to stabilise. Apart from code bugs, several inputs and factors impact computational stability and MC covergence in copula methods: **a.** averaging spread (rather than DL, PL) typically over-estimates the spread, **b.** a few early defaults introduce bias, **c.** using low spreads (i.e., 10-30 *bps*) is on the border of perturbations/computational instability, and **d.** correlation matrix is problematic (i.e., correlations are very high or homogeneous) and has no determinant.

2.2 Calculating Exact Default Time τ_i from simulated u_i

Exact default time $\tau = t_{m-1} + \delta t$ is estimated by first, identifying the year by an iterative procedure (comparing $\log(1-u)$ to the running sum of hazard rate) and second, calculating the year fraction δt.

Extending derivation for $\tau \overset{D}{=} F^{-1}(u)$ as the inverse of Exponential Distribution **CDF**

$$\tau \overset{D}{=} -\frac{\log(1-u)}{\lambda_\tau} \quad \Rightarrow \quad \log(1-u) = -\lambda_\tau \tau = -\int_0^\tau \lambda_s ds = \log P(0,\tau)$$

If $1-u = P(0,\tau)$ is the exact probability of survival, and default occurs as $t_{m-1} \leq \tau \leq t_m$

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

because survival probability is cumulative and decreasing over time (see CDS Bootstrapping). Re-specified in terms of intensities $\log P(0,t_m) = -\sum_{j=1}^{m} \lambda_j \Delta t_j$, the inequality becomes:

$$-\sum_{j=1}^{m} \lambda_j \Delta t_j \;\leq\; \log(1-u) \;\leq\; -\sum_{j=1}^{m-1} \lambda_j \Delta t_j \tag{2.1}$$

First, the inequality is used in the iterative procedure to determine the year of default.

We can continue to re-arrange around u

$$\begin{aligned} P(0,t_m) \;\leq\; 1-u \;\leq\; P(0,t_{m-1}) \\ -P(0,t_{m-1}) \;\leq\; u-1 \;\leq\; -P(0,t_m) \\ 1-P(0,t_{m-1}) \;\leq\; u \;\leq\; 1-P(0,t_m) \\ \mathrm{PD}_{m-1} \;\leq\; u \;\leq\; \mathrm{PD}_m \end{aligned}$$

Since we compare the threshold u to the cumulative probability of default, then if u is large (more than cumulative PD for year 5) it has a meaning of default occurring after year 5.

Second, calculating the year fraction also relies on $1-u = P(0,\tau)$, where $\tau = t_{m-1} + \delta t$

$$\delta t = -\frac{1}{\lambda_m} \log\left(\frac{1-u}{P(0,t_{m-1})}\right) = -\frac{1}{\lambda_m} \log\left(\frac{P(0,t_{m-1}+\delta t)}{P(0,t_{m-1})}\right) \tag{2.2}$$

This essentially refers to calculation of hazard rate as a log-ratio of survival probabilities.

Copula Fitting. Technical Note by Dr Richard Diamond

Copula estimation and sampling from operates with pseudo-samples U rather than original data X.

For Basket CDS project, $\mathbf{X}$ consists of five columns of historical data (say, equity returns or ΔPD used for estimation of correlation matrix $\mathbf{\Sigma}_{5\times5}$. Once chosen how to convert the historical data into uniform the **sampling from copula** algorithm implementation is a straightforward by-step. Choices are **converting to Normal variable by *differencing* $\Delta\mathbf{X} \to \mathbf{Z}$** or **applying probability integral transform and *kernel smoothing* $\mathbf{X} \to \mathbf{U} \to \mathbf{Z}$**.

For Gaussian copula, we estimate linear correlation ρ on Normally distributed $\mathbf{Z}$ so $\mathbf{\Sigma} = \rho(\mathbf{Z})$, for t copula we estimate correlation on the ranks of $\mathbf{X}$ so $\mathbf{\Sigma}_S = \rho(\mathbf{U})$ for Spearman's rho, while separate formula $\mathbf{\Sigma}_\tau = \rho_\tau(\mathbf{X})$ is defined for Kendall's tau. To convert into linear correlation $\rho = 2\sin\left(\frac{\pi}{6}\rho_S\right)$ and $\rho = \sin\left(\frac{\pi}{2}\rho_\tau\right)$ elementwise. This converted matrix is not guaranteed to be positive definite as required for Cholesky – so the nearest correlation matrix is obtained.

Historical sample data $\mathbf{X}^{\text{Hist}}$ (five columns credit spreads/default probabilities/hazard rates) is converted to pseudo-samples $\mathbf{U}^{\text{Hist}} = \widehat{F}(\mathbf{X}^{\text{Hist}})$. That is achieved by special transformation of data by its own *Empirical CDF* and involves *kernel density estimation* in order to guarantee uniformity.[1] Estimation done without making assumption about distribution of marginals is non-parametric and called Canonical Maximum Likelihood.[2]

Let's consider notation as we go from copula fitting (ie, calibration) to simulation,

- $\mathbf{Z} = \Phi^{-1}(\mathbf{U})$ obtained from pseudo-samples, so can be expressed as $\mathbf{Z}^{\text{Hist}}$. Use $\mathbf{\Sigma} = \rho(\mathbf{Z}^{\text{Hist}})$ The shortcut which avoids kernel smoothing is first, take differences $\mathbf{X} = \Delta\mathbf{X}^{\text{Hist}}$ and second, standartise $\mathbf{Z}_t^{(j)} = \frac{\mathbf{X}_t^{(j)} - \mu_j}{\sigma_j}$ for each row (observation) t of column j.
- For calculation of copula density, $\mathbf{U}_t \equiv \mathbf{U}_{t,1\times5}$ refers to **a row** of values for five reference names as observed at time t.
- $\mathbf{Z_{t+}}^{\text{Sim}}$ or simply $\mathbf{Z_{t+}}$ is a vector of simulated 1×5 Standard Normal random variables, and so $\mathbf{U_{t+}} = \Phi(\mathbf{Z_{t+}})$ for Gaussian or $\mathbf{U_{t+}} = T_\nu(\mathbf{Z_{t+}})$ for t copula.

  For the simulated 1×5 $\mathbf{U_{t+}}$, each value is converted to default time $u \to \tau$ using *its own* term structure of hazard rates $\tau \sim Exp(\hat{\lambda}_{1Y}, \ldots, \hat{\lambda}_{5Y})$.

Elliptical copulae might fail to fit dependence structure of empirical data (eg, higher density of tail observations, low density of the middle high-peaked observations). That is a model risk the copula method. A quick recipe is to **check bivariate scatters between the columns of U** – the scatter should have the familiar pattern of Elliptical copula density.

[1] There is no analytical formula for Empirical CDF function. It is obtained via a set of algorithms.

[2] Each column of $\mathbf{X}^{\text{Hist}}$ is 'a marginal' with its own univariate distribution that is usually bi-modal for raw credit spreads. Therefore, we have to work with *changes* in spreads $\Delta\mathbf{X}$ (daily or weekly).

2.3 Kernel Smoothing: An Introduction

The term refers to the estimation (fitting) of analytical probability density function $\widehat{f}()$ to the data. Most software-implemented kernel smoothers fit probability density function (PDF), from which additional steps have to be taken to obtain CDF $\widehat{F}$ – those are numerical integration over kernel PDF and interpolation. Altogether, the set of algorithms is known as Probability Integral Transform:

$$\mathbf{U} = \widehat{F}(\mathbf{X})$$

Performing MLE on pseudo-samples $\mathbf{U}$ instead of the original data $\mathbf{X}$ is a superior approach. For example, applying the familiar linear correlation formula on ranks $\mathbf{U}$ immediately delivers Spearman's rho, a rank correlation measure.

$$\mathbf{\Sigma}_S = \rho(\mathbf{U})$$

- $\mathbf{X} \to \mathbf{U} \to \mathbf{Z}$. Kernel smoothing on $\mathbf{X}^{\text{Hist}}$ by Empirical CDF algorithm (as implemented in Matlab/R/NAG functions), where implementation guarantees the uniformity of $\mathbf{U}^{\text{Hist}}$.
- $\mathbf{X} \to \mathbf{Z} \ldots \mathbf{U}$. Hidden assumption that original data $\mathbf{X}^{\text{Hist}}$ is converted to near-Normal, for example, by differencing $\mathbf{X} = \Delta\mathbf{X}^{\text{Hist}}$. Next steps are standardization $\mathbf{Z}_t^{(j)} = \frac{\mathbf{X}_t^{(j)} - \mu_j}{\sigma_j}$ and inferring pseudo-samples $\mathbf{U} = \Phi(\mathbf{Z})$. However, empirical pseudo-samples obtained this way (without kernel smoothing) might be insufficiently uniform.[3]

Where possible *use the ready implementation of kernel smoothing that gives Empirical CDF*, such as Matlab *ksdensity(), and check the uniformity of the output* $\mathbf{U}$ by plotting a histogram for each column (reference name).

When using kernel smoothing **the choice of bandwidth is very important**! Think of it as a bucket of observations for cumulative probability step (standard deviation on uniform scale). MATLAB's *ksdensity()* calibrates some optimal bandwidth, however you might be able to get better smoothing result in terms of uniformity of output $\mathbf{U}$ by interactively experimenting with the 'bw' setting from default down to circa 0.0001.

NAG kernel density estimation (PDF only) does require bandwidth as a ready input, so interactive experiment is necessary. Setting the bandwidth (window width) too high results in the data being represented as fully Normal and therefore, *oversmoothed* and highly correlated across names. Setting the bandwidth too low represents data very close to original (*undersmoothed*) and results in u_i that are zero or close, creating a problem with TINV calculation $T_\nu^{-1}(\mathbf{U})$.

- Each data column of $\mathbf{X}^{\text{Hist}}$ might require calibration of its own bandwidth setting. Here, for kernel PDF data are *changes* in credit spreads/default probabilities/hazard rates.
- Credit monitors rely on weekly changes and drop $1-3\%$ of extreme observations. [4]

[3] $\mathbf{Z} \ldots \mathbf{U}$ by $\mathbf{U} = \Phi(\mathbf{Z})$ is actually **the wrong way** but we are trying to see what $\mathbf{U}$ is implied.

[4] In utmost generality, one can look at changes between 5Y hazard rates averaged per period.

2.3.1 Kernel smoothing in various contexts

Let's begin with the idea of Empirical CDF $\widehat{F}()$. CDF is a function that assigns the lower weight on common observations and the higher weight on tail observations in order to transform a bell-shaped histogram to uniform histogram. Consider two simple cases of Empirical CDF.

- Case 1: bucket of one, each observation is converted as $u_i = \frac{1}{N}$. Then the uniformity is perfect but correlating the set of the same u_i with another alike set is not useful.
- Case 2: create buckets and calculate an average of each bucket. By the Central Limit Theorem the distribution of averages is guaranteed to be Normal.

Task: Fit the uniform density (ie, shown as a histogram) from the Normal data $\mathbf{X}$ (or close). The center of the density is not the target for conditioning.

Method: Implement state-conditioning using exponents for each tail. Exponential kernel gives flexible probabilities p_t, which will be applied to the original data as $u_i \propto p_i X_i$

$$p_t|z^* \propto e^{-\frac{|z_t - z^*|}{h}}$$

where bandwidth h decreasing as N_{obs} increases and $z^*_{1,2}$ 'cut' each of the tails.

Extra Smoothing: Given the data has already been converted to uniform $\mathbf{U}_t = \widehat{F}(\mathbf{X}_t)$, in this context kernel smoothing means fitting a continuous PDF from discrete observations $\mathbf{U}_t$.

$$\hat{f}(u,h) = \frac{1}{hN}\sum_{t=1}^{N} K\left(\frac{u - U_t}{h}\right)$$

u is a continuously smoothed variable, and bandwidth h decreases as N increases. The formula applies for any density. The choice of a PDF-like function for K is not important, for computational efficiency 'Gaussian kernel' (similar to exponential) is used

$$K(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2} \qquad \text{where} \quad \int_{-\infty}^{+\infty} K(t)dt = 1$$

For the uniform density the recommended kernel is $K = \dfrac{\Gamma(2)}{2\,\Gamma(1)\Gamma(1)}\, 1_{|u|<1}$. However, since the fraction is equals to $1/2$ it is easy to see that this expression is over-fitting and results in the same u_i (Case 1).

Histogram: is simply defined for any random x (as opposed to just u) as

$$\hat{f}(x,h) = \frac{1}{hN} N_j$$

where N_j is the number of observations falling in the interval $a + (j-1)h < x < a + jh$. For example, take sample of $N = 100$ and bandwidth of $h = 0.1$ ($N_s = 10$ observations per bucket expected if data is uniform) so, if $N_j = 8$ observations then bar height is 0.8, if $N_j = 15$ observations bar height is 1.5.

2.4 Copula Fitting: Rank vs. Linear Correlation

The model behind linear correlation is a linear regression, ie, the ability to draw a straight line to explain relationship between changes in two variables. Due to this shortcoming, the low linear correlation does not imply low dependence because the relationship can be strong but non-linear.

The idea behind rank correlation is rather simple, **each value can be converted to rank, then ranks are correlated**. Estimating from ranks gives scale-invariant measures; converting into ranks requires either formula or algorithm but the method itself is non-parametric.[5] Rank correlation measures are also invariant to any non-linear transformation in variable as long as it is monotonic. These properties are very suitable for copula fitting because **a.** we deal with credit spreads/PDs/hazard rates data and **b.** we desire robust estimation of dependence structure. Rank correlation matrix is unlikely to vary as we transform data, particularly to pseudo-samples $\mathbf{U}^{\text{Hist}}$.

- **Linear correlation** is a good measure of co-dependence for Normal variables. If the Normality assumption is violated, linear correlation can be very misleading. Linear correlation is not preserved under any general transformation. There is problem with non-elliptical distributions and non-linear transformation of a random variable (factor). For example, transforming from Normal to log-Normal variable by simply taking a log creates the problem of *attainable correlation* – if correlation measured between log-Normal variables, then its transformation into linear by something like $e^{\rho\cdots}$ will not give the full range $[-1, 1]$.

  The linear correlation of two Normal random variables is

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

- **Spearman's rho** is formally defined as a linear correlation of associated CDFs. The ranks of the data points are usually obtained by applying Empirical CDF to the data $\mathbf{U} = \widehat{F}(\mathbf{X})$ (up to a multiplicative factor).

$$\rho_S = \rho(U_1, U_2) = \frac{\mathbb{C}\text{ov}(F(X_1),\, F(X_2))}{\sqrt{\mathbb{V}\text{ar}(F(X_1))\mathbb{V}\text{ar}(F(X_2))}}$$

  Spearman's rho estimation: for N pairs of observations $(X_{1,i}, X_{2,i})$, generate ranking in the ascending order and calculate

$$\rho_S = 1 - \frac{6\sum D_i^2}{N(N^2-1)}, \qquad \text{where } D_i = (U_{1,i} - U_{2,i})$$

[5] Remember how applying CDF to the values is a special transformation [insert formula] that gives grades, and copula is referred to as a distribution of grades.

- **Kendall's tau** is another rank correlation measure, formally defined as the difference between probability of concordance and dis-concordance. It measures the degree to which the large values of one variable associate with the large values of another variable.

$$\begin{aligned}\rho_\tau &= \Pr\left[(X_1-\tilde{X}_1)(X_2-\tilde{X}_2)>0\right]-\Pr\left[(X_1-\tilde{X}_1)(X_2-\tilde{X}_2)<0\right]\\ &= \mathbb{E}\left[\text{sign}\left((X_1-\tilde{X}_1)(X_2-\tilde{X}_2)\right)\right]\end{aligned}$$

  A positive sign refers to the increasing pair (line goes upwards and downwards but its slope is positive), while a negative sign gives at least one decreasing case.

  Pairs (X_1, X_2) and 'an independent copy' $(\tilde{X}_1, \tilde{X}_2)$ are drawn from the identical joint distributions $F(X_1, X_2) \equiv F(\tilde{X}_1, \tilde{X}_2)$ are concordant if $X_1 > \tilde{X}_1$ and $X_2 > \tilde{X}_2$.

  Kendall's tau estimation: for N pairs of observations $(X_{1,i}, X_{2,i})$, we use a sign function $sign(x) = -1$ if $x < 0$ and $sign(x) = 1$ if $x > 0$.

$$A_{ij} = \text{sign}\left((X_{1,i} - X_{1,j})(X_{2,i} - X_{2,j})\right)$$

  and estimate using

$$\rho_\tau = \mathbb{E}[\mathbf{A}] = \frac{2}{N(N-1)}\sum_{i=1}^{N}\sum_{j>i}^{N} A_{ij} = \frac{N_c - N_d}{N(N-1)/2}$$

  where N_c is the number of concordant pairs and N_d is the number of discordant pairs.

The linear correlation is a required input to both, Gaussian and t copulae. If using a rank correlation measure the result has to be converted as $\rho = 2\sin\left(\frac{\pi}{6}\rho_S\right)$ and $\rho = \sin\left(\frac{\pi}{2}\rho_\tau\right)$.

Linear Correlation from Returns Data $\mathbf{X} \to \mathbf{Z} \ldots \mathbf{U}$ estimation can be streamlined by relying on a reasonable assumption of Normal distribution for asset returns. This works better over the long-term. So, for estimating default correlation matrix using equity returns it makes sense to use a 3-year historical time period (longer period for weekly data).

1. Convert price data into linear returns $\mathbf{R} = .\dfrac{X_{t+1} - X_t}{X_t}$ where X_t is equity price at time t.
2. Normalise returns $\mathbf{Z} = \dfrac{\mathbf{R}_t^j - \widehat{\mu}}{\widehat{\sigma}}$ where $\widehat{\mu}$ is sample mean and $\widehat{\sigma}$ is sample std. deviation.
3. Calculate linear correlation $\widehat{\mathbf{\Sigma}} = \rho(\mathbf{Z})$ (between each two columns of data).

The *implied* pseudo-samples can be obtained $\mathbf{U} = \Phi(\mathbf{Z})$ and checked for uniformity (by histogram). On the other hand, one can revert to the preferred CML method by applying kernel smoothing to returns $\mathbf{U} = \widehat{F}(\mathbf{R})$ and obtaining $\mathbf{Z} = \Phi^{-1}(\mathbf{U})$ – in this case, the kernel smoothing of Empirical CDF $\widehat{F}(\mathbf{R})$ is a conventional task of smoothing over the Normal-like density of $\mathbf{R}$.

Important Disclaimer. Imposing assumption about distribution of changes the procedure from *Canonical Maximum Likelihood* to *Inference Function for Margins*. CML gives MLE-optimal copula fit without parametric assumptions about marginal distributions.

2.4.1 Where rank correlation fails

Remember that correlating *probabilities of default* is the preferred method to obtain default correlation matrix (while estimation from equity returns is just an acceptable choice). If one starts examining time series of credit spreads/PD/hazard rates empirically (see DB Research data for Sovereign CDS_{5Y} and ready-bootstrapped PD_{5Y}) the quick findings will be as follows:

- credit spread/PD time series are non-stationary (ie, have unit root); and
- exhibit bimodal distribution (histogram)

Rank correlation measures were developed for any random variable but their application to the levels of non-stationary variables remains problematic. The important aspect is that **even rank correlation measures require co-monotonic (linear, exponential, etc) dependence.** Correlation measures fail when there is a break in co-monotonicity (eg, zig-zag or cross shapes). Combining two variables with a bimodal distribution each seems to create non-monotonic dependence.

2.5 t Copula and Degrees of Freedom ν (MLE Procedure Guide)

Degree of freedom ν is an additional parameter required for t copula that regulates tail dependence – that is, increased likelihood of large(small) values in simulated $\mathbf{U}_{t+}$ appear together.

In our flexible Canonical Maximum Likelihood approach, we first obtain the default correlation matrix and then use the same underlying data (pseudo-samples $\mathbf{U}^{Hist}$), in order to calculate the log-likelihood function (of ν) as sum of densities.[6]

$$\underset{\nu}{\text{argmax}} \left\{ \sum_{t=1}^{T} \log c_t \left(\mathbf{U}_t^{Hist}; \nu, \widehat{\mathbf{\Sigma}} \right) \right\}$$

To calculate a contribution to copula density (joint density) we input $\mathbf{U}_{t,1\times 5}$ values for five reference names as observed at time t. Then, we calculate density contribution at time $t+1$ and up to T, **a row at a time**.[7]

For each value of ν from 1 to 25, we repeat the calculation of total log-likelihood and produce a plot that must be similar to Figure 2.1 below.

[6] Inferring linear correlation explicitly first and then 'fixing' it is a restriction compatible with Canonical MLE.

[7] Unlike probability density, copula density $c()$ is calculated 'up to a multiple'. The result can be above 1 and scale depends on the original transformation into $\mathbf{U}$. Canonically, the sum of densities under log will be negative $O(-2000)$ to $O(-800)$.

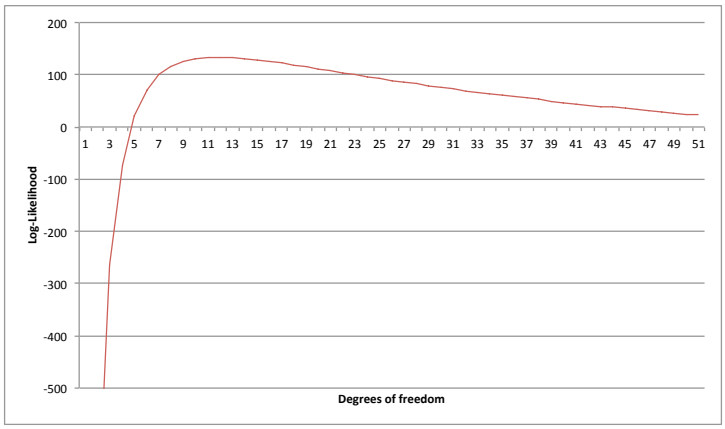

Figure 2.1: Log-likelihood for t copula as a function of degrees of freedom ν.

A shortcut to degrees of freedom estimation is to make an informed choice between 4 and 13 (circa 25/2). Do not hesitate using this shortcut at least initially. An explanation for the shortcut comes from the approach to degrees of freedom in a structural equation system.

1. Within a multivariate regression

$$d.f. = N_{total} - N_{var} - 1$$

For the 5×5 correlation matrix which is symmetric with ones on the diagonal,

$$d.f. = (5 \times 5 - 5)/2 - 1 = 9$$

2. If two variables are strongly correlated then we effectively have 4 independent variables,

$$d.f. = (4 \times 4 - 4)/2 - 1 = 5$$

d.f. parameter reduced to from 9 to 5 with increased correlation!

3. With 25-30 d.f. t distribution begins to closely approximate the Normal distribution.

Tail dependence index means that with large (small) values of U_1 we expect to observe the large (small) values of U_2 (in probability terms). This applies applies to t copula and is the reason for sampling from it.[8]

$$\lambda_l = \lambda_u = 2t_{\nu+1}\left(-\sqrt{\frac{(\nu+1)(1-\rho)}{1+\rho}}\right)$$

where $\lambda_l = \lambda_u$ is coefficient of tail dependence, ν is degrees of freedom parameter and ρ is linear correlation between U_1 and U_2.

[8]Correlated values sampled from Gaussian copula are asymptotically independent in both tails. Even with high correlation input, there is no tail dependence (increase in likelihood of co-movement) from a Gaussian copula.

Inverse CDF. t Distribution Working with the t copula requires the CDF $T_v(\mathbf{X})$ (when sampling from) and its inverse $T_v^{-1}(\boldsymbol{U'})$ (when fitting – calculating density $c()$ to obtain likelihood). Student's t CDF formula (2.52) is on page 14 of *Monte-Carlo Methods in Finance* textbook by Peter Jaekel.

The formula uses Beta function which is not available in VBA but things can be done using the *BetaDist()* as below. To understand link between *Beta function* and *BetaDist()* compare (2.52) to (2.38) and make necessary derivations.

This works for $x > 0$ (to understand the modification for $x \leq 0$ refer to the formulae)

```
'x = Z-score,  n = degrees of freedom
Public Function MyTDIST(x As Double, n As Single) As Double
MyTDIST = 1 - Application.BetaDist(n / (n + x ^ 2), n / 2, 1 / 2) / 2
End Function
```

To calculate the inverse of a Student's t CDF, the following code can be used

```
Public Function MyTINV(x As Double, n As Single) As Double
Dim tmp As Double
tmp = Application.BetaInv(2 * Application.Min(x, 1 - x), n / 2, 1 / 2)
tmp = 1 / tmp
MyTINV = (Sgn(x - 0.5) * (n * (tmp - 1)) ^ 0.5)
End Function
```

It is up to delegates to confirm validated code.

Summary Estimation of both, correlation and degrees of freedom, is known as **copula fitting**. As an illustration please check out how copula fitting is done in MATLAB by using only a few ready functions. `http://www.mathworks.co.uk/help/stats/copulafit.html`

2.6 Sensitivity Analysis (required)

Sensible hedging strategies for a credit portfolio (basket credit swap) rely on the notion of Default Basket Delta that measures sensitivity *w.r.t.* changes in credit spreads of reference names as

$$\Delta_i = \frac{\partial V_{Basket}}{\partial S_i} \Big/ \frac{\partial V_i}{\partial S_i}$$

or Change in Basket Spread / Change in Credit Spread (for a reference name).

If you can, come up with a different term structure of hazard rates to be used for pricing – that would be important aspect of testing the model risk. Sensitivity analysis *wrt* Recovery Rate and Discount Factor (when pricing) is optional. Generally one can explore the limitations of the Credit Triangle $S = \lambda(1 - R)$.

Correlation Sensitivity. Other key input for sensitivity analysis is default correlation matrix $\mathbf{\Sigma}$. *Imposing absolute levels of base default correlation*, ie, same for all names low 0.05 to high 0.95, is a simple stress-testing technique on the fair spread. With t copula, keep the estimated d.f. parameter fixed.

A smarter technique would be *to shift the values of correlation matrix by a percentage*, eg, 10% at a time. Fisher transform can be used to convert correlation to a variable such as $Z \in [-\infty, \infty]$ – this is useful in stress-testing

$$Z = \frac{1}{2} \log \frac{1+\rho}{1-\rho}$$

the inverse transform is

$$\rho = \frac{e^{2Z} - 1}{e^{2Z} + 1}.$$

The key model risk is about copula fitting from the empirical data, particularly if data are credit spreads/default probabilities. We have discussed the importance of checking bi-variate plots (visually) for both $\mathbf{X}^{\text{Hist}}$ and $\mathbf{U}^{\text{Hist}}$. This check is for tricky non-monotonic relationship (eg, zig-zag, cross) between correlated variables which can't be picked up by even rank measures.

Histograms for each column (reference name) of $\mathbf{X}^{\text{Hist}}$ as well as kernel-smoothed $\mathbf{U}^{\text{Hist}}$ will show the empirical marginal distributions and effectiveness of kernel smoothing – each histogram of $\mathbf{U}^{\text{Hist}}$ must have uniform bars of similar height. Remember that any histogram presents buckets of observations subject to bandwidth h.

Chapter 3

Monte Carlo Improvement

3.1 for Credit Spread Pricing

Please see the section on Averaging PL and DL. The number of simulations required for convergence (with naive Excel random numbers) can reach into $100,000$s, particularly for the higher kth-to-default. Check sensibility of your pricing result (is it as expected, is its variance reducing) even if it *appears* to converge after $10,000$s simulations.

The key reference for simulation design and sampling strategies, particularly when sampling from copula required, is a textbook on *Monte-Carlo Methods in Finance* by Peter Jaekel.

3.2 for HJM Simulations

With $k = 3 \ldots 5$ factors (volatility functions) used to simulate the curve, the dimensionality of Monte Carlo simulation is sufficiently low. Potentially, fractal patterns in random numbers across the depth of multiple dimensions can lead to pricing patterns that can be arbitraged.

There is no requirement to simulate the curve for up to 25 years into the 'future time' (with $\Delta t = 0.01$ that would be 2500 rows on the HJM Spreadsheet). You can start with a 5-year future period – that will give enough data for most caplet pricing illustrations.

However, for each simulated 'table' of forward rates one can observe that the same kind of simulated curve propagates into the future time. That raises an issue about importance of re-sampling when pricing by Monte Carlo.

[Insert 3D plots here]

3.3 Random Numbers Generation

There are following RN generation methods that you can adopt from ready implementations (eg, NAG Library) to give professional quality to the Monte Carlo within your project. Low discrepancy generators provide statistically dependent numbers with improved evenness in the multidimensional space.

1. Mersenne Twister (Pseudo RN)
2. Halton Numbers (low discrepancy, Quasi RN)
3. Sobol Numbers (low discrepancy, Quasi RN)

The convergence provided by low discrepancy RN generators is the order of $c(d)\frac{(\ln N)^d}{N}$ vs. $\frac{1}{\sqrt{N}}$ for quasi RN generators such as Excel's *RAND()* and Mersenne Twister. N refers to the number of simulations, d is number of dimensions (eg, $d = 3$ factors in HJM SDE and $d = 5$ reference names in Basket CDS), and $c(d)$ is some scaling function.

Example for HJM implementation: while Excel-generated random numbers might not give convergence even after $2,000$ simulations, Monte Carlo with proper low discrepancy RNs gives a good (low variance) estimate right after 200 simulations and satisfactory convergence after about 600 simulations. The second problem brought by this example is that **the use of Excel's pseudo random numbers over-estimated the price of a derivative.** Between Excel RAND() and Quasi RN, the variance of the estimator can vary threefold!

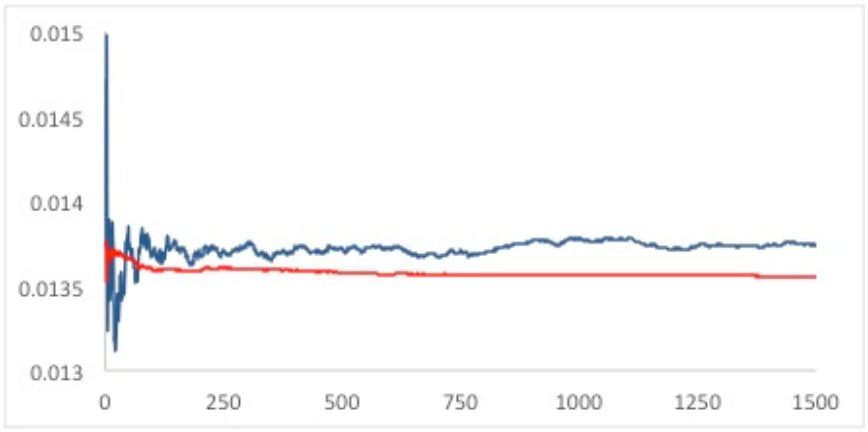

Figure 3.1: Dark blue line is a derivative price using *RAND()*, while red line result is for the same derivative obtained using low-discrepancy RNs.

3.4 Variance Reduction

Monte-Carlo simulation is evaluated by the speed and related to that, variance of the estimator. In addition to plotting a running average of the price for convergence, plot running standard deviation (after each 100 of simulations added).

The important trick to reduce variance is to identify which simulated inputs increase the standard deviation of the estimate. For example in Basket CDS projects very small u imply default times τ_i which are less than one quarter. Because each default time is standalone (inter-arrival times are conditionally independent) it is possible to remove those early defaults without breaking the continuity of Monte-Carlo. On the other hand, you will not be able to do the same thing when simulating a log-normal asset price (i.e., removing price levels at will).