Correlation and Aggregate Loss Distributions With An Emphasis On The Iman-Conover Method

Stephen J. Mildenhall
CAS Working Party on Correlation

November 27, 2005

Abstract

Motivation. The CAS Research Working Party on Correlation and Dependencies Among All Risk Sources has been charged to "lay the theoretical and experimental foundation for quantifying variability when data is limited, estimating the nature and magnitude of dependence relationships, and generating aggregate distributions that integrate these disparate risk sources."

Method. The Iman-Conover method represents a straight forward yet powerful approach to working with dependent random variables. We explain the theory behind the method and give a detailed step-by-step algorithm to implement it. We discuss various extensions to the method, and give detailed examples showing how it can be used to solve real world actuarial problems. We also summarize pertinent facts from the theory of univariate and multivariate aggregate loss distributions, with a focus on the use of moment generating functions. Finally we explain how Vitale's Theorem provides a sound theoretical foundation to the Iman-Conover method.

Availability. The software used to generate the paper's examples is available at http://www.mynl.com/wp.

Keywords. Dependency, correlation, aggregate loss distributions, fast Fourier transform.

Chapter 1

INTRODUCTION

The Iman Conover (IC) Method is a practical, down-to-earth approach to dealing with dependent random variables. It should be part of every actuary's toolkit.

When two variables X and Y are positively correlated there is a tendency for large values of X to be associated with large values of Y. Knowing how the large values are associated would make it possible to work in reverse: by ordering samples from X and Y so that large-large matches and small-small matches are more likely would result in a bivariate sample with positive correlation. The Iman-Conover (IC) method gives a way of determining reasonable associations, and hence inducing correlation between samples of variables. It is ideally suited to simulation work where marginal distributions are sampled independently but must be combined to achieve some desired level of correlation. The IC method is used by the popular @Risk software package to induce correlation.

Before describing the IC method, we begin with a review of measures of correlation and association in Chapter 2. Then, in Chapter 3 we describe several useful techniques for working with univariate and multivariate aggregate loss distributions. These include formulae to compute moments of aggregates using moment generating functions, a discussion of mixed Poisson counting distributions, approximating univariate aggregates using the shifted gamma and shifted lognormal distributions, Fast Fourier transform methods, and computing correlation coefficients related to multivariate aggregates in a variety of situations.

Next we turn to a description of the IC method itself, which can simplistically be described as follows. Given two samples of n values from known marginal distributions X and Y and a desired correlation between them, first determine a sample from some reference bivariate distribution that has exactly the desired linear correlation. Then re-order the samples from X and Y to have the same rank order as the reference distribution. The output will be a sample from a bivariate distribution with the correct marginal distributions and with rank correlation coefficient equal to that of a bivariate distribution which, in turn, has exactly the desired correlation coefficient. Since linear correlation and rank correlation are typically close, the output has approximately the desired correlation structure. What makes the IC method work so effectively is the existence of easy algorithms to determine samples from reference distributions with prescribed correlation structures. Obviously the method can then be extended to work with samples from multivariate distributions in any dimension.

In their original paper, Iman and Conover [21] point out that their method has several desirable properties.

1. It is very simple to understand and implement.

2. It is distribution free; it may be used with equal facility on all types of input distributions.

3. It can be applied to any sampling scheme for which correlated input variables could logically be considered. That is, the output sample contains the same values as the input, only with a different pairing. Hence in Latin hyper cube sampling, the integrity of the intervals is maintained.

4. The marginal distributions remain intact.

The actual IC method involves some enhancements over such a naive implementation, and we give full details in Chapter 4. We give a step-by-step example to explain how the method works in practice in Section 4.3. In Section 4.4 we show how the basic IC method can be extended, and illustrate the impact these extensions have on the types of multivariate distributions the method produces. Section 4.5 compares the IC method with the normal copula method described in Wang [37].

Chapter 5 gives a detailed practical example which computes the bivariate distribution of ceded and retained losses in a book with an excess of loss reinsurance structure. Such a bivariate distribution is necessary to compute the net underwriting result if the reinsurance contains any variable feature like a sliding scale commission, profit commission or annual aggregate deductible.

Chapter 6 discusses the theoretical underpinnings of the IC method in a more technical manner. It can be ignored with impunity by readers more interested in practice than theory.

Appendix A discusses practical computational issues and describes some freely available software which can be used to implement the IC method in Excel.

Some sections are marked with an asterisk. These I regard as interesting, but not "core". The remaining, un-starred sections, contain core facts which I think every actuary working with correlation and aggregate losses should know.

When modeling correlation the reader should remember that the model must follow an understanding of reality. Model building, especially modeling correlation, must start with an understanding of the underlying processes. Graphical representations are often useful to aid understanding and help communicate results. It may be necessary to build pre-models to understand the underlying processes and use these to parameterize quicker, more computationally efficient, implementations. The IC method is ideal here: having understood the drivers of correlation and estimated an appropriate correlation matrix and suitable multivariate structure, the IC method can be used to produce correlated samples with blistering speed. However, the reader should not succumb to the temptation to estimate a

200×200 correlation matrix using data and expect to get realistic result from it. It will be more noise than signal.

In conclusion, we recommend the IC method as being fast, simple to understand, and efficient. We also recommend the use of a shifted gamma or shifted lognormal distribution to model univariate aggregate losses.

Throughout the paper bold face roman variables represent vectors (lower case) and matrices (upper case). Sections in a smaller typeface are optional, more technical discussions. Sections marked with an asterisk* contain non-core material.

Acknowledgement. I would like to thank Christopher Monsour, Chuck Thayer, Leigh Halliwell, Roger Hayne, Phil Heckman, and Kevin Shang for their helpful comments and suggestions.

Chapter 2

CORRELATION AND ASSOCIATION

Before discussing specific measures of correlation and association, recall that two random variables X and Y are independent if

$$\Pr(X \in A, Y \in B) = \Pr(X \in A)\Pr(Y \in B) \tag{2.1}$$

for all suitable sets A and B. It is possible to prove that X and Y are independent if and only if

$$\mathrm{E}[f(X)g(Y)] = \mathrm{E}[f(X)]\mathrm{E}[g(Y)] \tag{2.2}$$

for all functions f and g.

See Wang [37] and Press et al. [31] for more information on the definitions and terms described here.

2.1 Correlation, Rank Correlation and Kendall's Tau

There are three common measures of association (more commonly called simply correlation) between two random variables X and Y: linear or Pearson correlation, rank correlation and Kendall's Tau.

The linear or Pearson correlation coefficient, usually just called correlation, is defined as

$$\rho(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sigma(X)\sigma(Y)} = \frac{\mathrm{E}[(X-\mathrm{E}(X))(Y-\mathrm{E}(Y))]}{\sigma(X)\sigma(Y)} \tag{2.3}$$

where $\sigma(X)$ is the standard deviation of X. By the Cauchy-Schwarz inequality the correlation coefficient always lies in the range $[-1,1]$. The correlation coefficient is sometimes called the Pearson correlation coefficient or linear correlation coefficient. Perfect correlation, when $\rho = \pm 1$, occurs if and only if $Y = aX + b$ for constants $a > 0$ (resp. $a < 0$) and b. The correlation coefficient is a natural measure of association when X and Y come from a bivariate normal distribution because it is enough to completely specify the dependence between the marginals. Needless to say, such pleasant results do not hold in general! For a multidimensional distribution the correlation matrix has i,jth element equal to the correlation coefficient of the i and jth marginals.

A related measure is the covariance coefficient defined as

$$\omega(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\mathrm{E}(X)\mathrm{E}(Y)}. \tag{2.4}$$

By (2.2) independent variables are uncorrelated. However, the converse is not true. The classic counter-examples of uncorrelated but dependent variables include

- X a standard normal and $Y = X^2$,
- (X,Y) uniformly distributed over a circle of radius one centered at the origin, and
- (X,Y) distributed with a bivariate t-distribution with zero correlation.
- Let $X, X_1, X_2, \ldots$ be identically distributed random variables with mean zero and let N be a counting distribution. Then $A = X_1 + \cdots X_N$ and N are uncorrelated but not independent. If X and X_i have a non-zero mean then $\mathrm{Cov}(A,N) = \mathrm{E}(X)\mathrm{Var}(N)$.

The correlation coefficient of a bivariate sample (X_i, Y_i), $i = 1, \ldots, n$, is defined as

$$\rho = \frac{\sum_i (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_i (X_i - \bar{X})^2 \sum_i (Y_i - \bar{Y})^2}} \tag{2.5}$$

where $\bar{X} = n^{-1} \sum_i X_i$ and similarly for $\bar{Y}$.

Let X be an $n \times r$ matrix representing an n-fold sample of $1 \times r$ vectors. Suppose that the means of the columns of X are all zero (subtract the means if necessary). Then the variance-covariance matrix of X is simply $n^{-1}X'X$, where X' denotes matrix transpose.

The second measure of association we consider is rank correlation. Given a sample $X_1, \ldots, X_n$ of observations from a random variable X the rank order statistics $X_{(1)}, \ldots, X_{(n)}$ are a permutation of the original observations ordered so that $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$. Call j the rank of $X_{(j)}$. The rank correlation, or Spearman rank correlation, of a sample is defined as the correlation of the ranks of the sample. Rank correlation lies in the range $[-1, 1]$ because it is a correlation. It is invariant under strictly monotonic transforms of X and Y, so for example the rank correlation of a sample (X, Y) is the same as the transformed samples $(\log(X), \log(Y))$ or $(\exp(X), \exp(Y))$. Rank correlation is a nonparametric measure of association because it is invariant under transformation. For continuous random variables rank correlation can also be computed as

$$12\mathrm{E}[(F_X(X) - 0.5)(F_Y(Y) - 0.5)] \tag{2.6}$$

where F_X (resp. F_Y) is the distribution function of X (resp. Y).

The third common measure of association is called Kendall's tau. Kendall's tau looks at concordances and discordances between pairs of data points (x_i, y_i) and (x_j, y_j). A pair of observation-pairs is concordant if $(x_i - x_j, y_i - y_j)$ lies in the upper right hand or lower left hand quadrants of the plane, and discordant otherwise. Take all $n(n-1)/2$ distinct pairs of data from the sample and count the number of concordances c and discordances d, except that if the ranks of the x's are the same the pair is called an extra y pair and if the ranks of the y's are the same the pair is an extra x pair. If the ranks of both x and y are the same the pair

does not count at all. Let e_x and e_y be the number of extra x and y pairs. Kendall's tau is defined in Press et al. [31] as

$$\tau = \frac{c-d}{\sqrt{c+d+e_y}\sqrt{c+d+e_x}}. \tag{2.7}$$

Kendall's tau can also be computed as

$$\tau(X,Y) = 4\int_0^1\int_0^1 F_{X,Y}(x,y)d^2F_{X,Y}(x,y) - 1 \tag{2.8}$$

provided singularities are handled appropriately, see Wang [37]. The Kendall's tau of a sample (X_i, Y_i), $i = 1, \ldots, n$ can be computed as

$$\tau = \frac{2}{n(n-1)}\sum_{i<j}\text{sign}((X_i - X_j)(Y_i - Y_j)) \tag{2.9}$$

where $\text{sign}(z)$ is 1, 0 or -1 when z is positive, zero or negative.

The statistics of Kendall's tau are covered in more detail by Conover, [6]. Conover points out that if the null hypothesis that (X, Y) are independent is true, the distribution of tau approaches the normal quite rapidly. Hence the normal approximation for tau is better than the one for Spearman's rho under the null hypothesis. He also points out that tau has a natural interpretation in terms of the probability that an observation is concordant versus discordant.

Equation (2.9) is precisely consistent with the definition in Equation (2.7) only when there are no ties. In the no-ties case, (2.9) is the form that Kendall proposed in his 1938 paper. When there are ties, (2.9) ignores ties in either X or Y, but it counts every pair of observations in the total used in the denominator.

Equation (2.7) accounts explicitly for ties without distorting the answer unduly, and it always provides an answer regardless of the number of ties in the sample. Conover's method fails when every pair results in a tie in the rank of the Xs. On the other hand, if the ranks of all the Xs are tied, X should not really be termed a "variable", much less a "random variable"!

Conover's alternative to (2.9) is to use a different method to account for ties. If the Y ranks are tied, he adds $1/2$ to both c and d. If the X ranks are tied, the

comparison is dropped entirely, adding nothing to c or to d. Otherwise, c gets incremented for positive signs and d counts the negatives in the sum. Conover's final statistic is

$$\tau = \frac{c-d}{c+d}. \tag{2.10}$$

Conover's statistic adopts a form of Kendall's tau that was introduced by Goodman and Kruskal [14]. Equation (2.10), which is also called the gamma coefficient, can attain the values $+1$ and -1 even in the presence of ties in the sample data.

There are several relationships between these measures of correlation, particularly if the sample comes from a multivariate normal distribution. For example if (X, Y) are bivariate normal with correlation ρ then

$$\tau(\Phi(X), \Phi(Y)) = \frac{2}{\pi} \arcsin(\rho) \tag{2.11}$$

and the rank correlation

$$\text{rankCorr}(\Phi(X), \Phi(Y)) = \frac{6}{\pi} \arcsin(\rho/2). \tag{2.12}$$

Similar results hold for samples from any elliptically contoured distribution, see Fang and Zhang [11], Embrechts, Lindskog and McNeil [9] and Embrechts, McNeil and Straumann [10].

2.2 Comonotonicity*

Two random variables X and Y are comonotonic if there exists a third variable Z and non-decreasing functions f and g such that $X = f(Z)$ and $Y = g(Z)$. For example, if X and Y are two different excess layers on the same risk then they are comonotonic. A stock and an option on it have comonotonic payouts. Comontonicity represents a high level of association between two values, but it need not result in a high level of linear correlation.

Some authors propose that risk measures r should be sub-additive, $r(X+Y) \leq r(X) + r(Y)$, with the tag-line "mergers cannot cause risk". Coherent measures require sub-additivity, see Artzner et al. [2]. Others authors propose additivity for comonotonic risks $r(X + Y) = r(X) + r(Y)$ if X and Y are comonotonic, see Wang [36].

2.3 Measures for Non-Normal Variables*

Linear correlation is the perfect measure of association for normally distributed random variables. It does not deal so effectively with non-normal variables. However, any continuous random variable X with distribution function F can be transformed into a normal variable Y via

$$Y = \Phi^{-1}(F(X)). \tag{2.13}$$

It therefore makes sense to transform non-normal variables using (2.13) and then to compute correlations between the transformed variables. If X is already a normal variable then (2.13) simply normalizes X to mean 0 and standard deviation 1.

Normalizing transformations are related to the IC method and the normal copula method as we will explain with Theorem 2 below. The normalizing transformation has been used in the literature by Wang [38] and Sornette et al. [32] amongst others.

Chapter 3

GENERAL PROPERTIES OF AGGREGATE LOSS DISTRIBUTIONS

Here we gather together some useful techniques for working with aggregate distributions, modeling correlation, parameter uncertainty, and so forth. Many of the techniques we introduce here will be used in the extended example, given in the Chapter 5. We introduce the negative multinomial distribution and we provide an introduction to Fast Fourier Transform (FFT) methods in both one and two dimensions. We begin with a discussion of moment generating functions and mixed Poisson frequency distributions.

We will use the following notation. The variance of a random variable X is $\mathrm{Var}(X) = \mathrm{E}(X^2) - \mathrm{E}(X)^2$. The standard deviation is $\sigma(X) = \sqrt{\mathrm{Var}(X)}$. The coefficient of variation (CV) of X is $\mathrm{CV}(X) = \sigma(X)/\mathrm{E}(X)$. The skewness of X is $\mathrm{E}[(X - \mathrm{E}(X))^3]/\sigma(X)^3$.

3.1 Moment Generating Functions

The moment generating function of a random variable X is defined as

$$M_X(\zeta) = \mathrm{E}(\exp(\zeta X)). \tag{3.1}$$

The moment generating function is related to the characteristic function of X which is defined as $\phi_X(\zeta) = \mathrm{E}(\exp(i\zeta X)) = M_X(i\zeta)$. ϕ is guaranteed to converge for all real ζ and so is preferred in certain situations.

Moment generating functions get their name from the fundamental property that

$$\left.\frac{\partial^n M_X}{\partial \zeta^n}\right|_{\zeta=0} = \mathrm{E}(X^n) \tag{3.2}$$

for all positive integers n provided the differential exists.

Let F be the distribution function of X. Feller [12, Section XVII.2a] shows that if F has expectation μ then ϕ, the characteristic function of F, has a derivative ϕ' and $\phi'(0) = i\mu$. However the converse is false. Exactly what does hold is spelt out in the next theorem.

Theorem 1 *(Pitman) The following are equivalent.*

1. $\phi'(0) = i\mu$.
2. *As* $t \to \infty$, $t(1 - F(t) + F(-t)) \to 0$ *and*
$$\int_t^{-t} x dF(x) \to \mu. \tag{3.3}$$
$F(-t) := \lim F(s)$ *as* $s \uparrow t$.
3. *The average* $(X_1 + \cdots + X_n)/n$ *tends in probability to* μ*, that is* $\Pr(|(X_1 + \cdots + X_n)/n - \mu| > \epsilon) \to 0$ *as* $n \to \infty$.

Note that the condition for the limit in (3.3) to exist is weaker than the requirement that $\mathrm{E}(X)$ exists if X is supported on the whole real line. For the expectation to exist requires $\int_{-\infty}^{\infty} x dF(x)$ exists which means $\lim_{t\to-\infty} \lim_{s\to\infty} \int_t^s x dF(x)$.

The moment generating function of a bivariate distribution (X_1, X_2) is defined as

$$M_{X_1,X_2}(\zeta_1, \zeta_2) = \mathrm{E}(\exp(\zeta_1 X_1 + \zeta_2 X_2)). \tag{3.4}$$

It has the property that

$$\left.\frac{\partial^{m+n} M_{X_1,X_2}}{\partial \zeta_1^m \partial \zeta_2^n}\right|_{(0,0)} = \mathrm{E}(X_1^m X_2^n) \tag{3.5}$$

for all positive integers n, m.

The MGF of a normal variable with mean μ and standard deviation σ is $M(\zeta) = \exp(\mu\zeta + \sigma^2\zeta^2/2)$. The MGF of a Poisson variable with mean n is

$$M(\zeta) = \exp(n(e^\zeta - 1)), \tag{3.6}$$

a fact we will use repeatedly below.

See Feller [12] and Billingsley [3] for more information on moment generating functions, characteristic functions and modes of convergence.

3.2 Mixed Poisson Frequency Distributions

Here we consider some basic facts about mixed Poisson frequency distributions. Let G be a non-negative mixing distribution with $\mathrm{E}(G) = 1$ and $\mathrm{Var}(G) = c$. The variable c is called the contagion. Let N be a claim count random variable where the conditional distribution of $N|G = g$ is Poisson with mean gn for some non-negative real n. We will call N a G-mixed Poisson random variable.

By (3.6), the MGF of a G-mixed Poisson is

$$M_N(\zeta) = \mathrm{E}(e^{\zeta N}) = \mathrm{E}(\mathrm{E}(e^{\zeta N}|G)) = \mathrm{E}(e^{nG(e^{\zeta}-1)}) = M_G(n(e^{\zeta}-1)) \tag{3.7}$$

since $M_G(\zeta) := \mathrm{E}(e^{\zeta G})$. Thus

$$\mathrm{E}(N) = M_N'(0) = nM_G'(0) = n, \tag{3.8}$$

because $\mathrm{E}(G) = M_G'(0) = 1$, and

$$\mathrm{E}(N^2) = M_N''(0) = n^2 M_G''(0) + nM_G'(0) = n^2(1+c) + n. \tag{3.9}$$

Hence

$$\mathrm{Var}(N) = n(1+cn). \tag{3.10}$$

Finally

$$\mathrm{E}(N^3) = M_N^{(3)}(0) = n^3 M_G^{(3)}(0) + 3n^2 M_G''(0) + nM_G'(0) \tag{3.11}$$

from which it is easy to compute the skewness.

We can also assume G has mean n and work directly with G rather than nG, $\mathrm{E}(G) = 1$. We will call both forms mixing distributions.

Table 3.1: Parameterizations of the Gamma Distribution

Model	Density	MGF	Mean	Var
(a) α, β	$\frac{x^{\alpha-1}e^{-x/\beta}}{\beta^{\alpha}\Gamma(\alpha)}$	$(1-\beta t)^{-\alpha}$	$\alpha\beta$	$\alpha\beta^2$
(b) α, β	$\frac{x^{\alpha-1}\beta^{\alpha}e^{-x\beta}}{\Gamma(\alpha)}$	$(1-t/\beta)^{-\alpha}$	α/β	α/β^2
(c) α, θ	$\frac{x^{\alpha-1}e^{-x/\theta}}{\theta^{\alpha}\Gamma(\alpha)}$	$(1-t\theta)^{-\alpha}$	$\alpha\theta$	$\alpha\theta^2$

3.3 Gamma and Negative Binomial Variables

Recall that a negative binomial is a gamma-mixed Poisson: if $N|G$ is distributed as a Poisson with mean G, and G has a gamma distribution, then the unconditional distribution of N is a negative binomial. Both the gamma and negative binomial occur in the literature with many different parameterizations. The main ones are shown in the Tables 3.1 and 3.2.

In Table 3.1 model (a) is used by Microsoft Excel, Wang [37] and Johnson et al. [22, Chapter 17]. Model (b) is used by Bowers et al. [4]. Model (c) is used by Klugman, Panjer and Willmot in the Loss Models text [25]. Obviously model (c) is just model (a) with a change of notation.

In Table 3.2 model (a) is used by Wang and Loss Models, (b) by Johnson et al. [24, Chapter 5] and (c) by Bowers et al. [4] and Excel. In model (c) the parameter r need not be an integer because the binomial coefficient can be computed as

$$\binom{r+x-1}{x} = \frac{\Gamma(r+x)}{\Gamma(r)x!},$$

an expression which is valid for all r. The cumulative distribution function of the negative binomial can be computed using the cumulative distribution of the beta

distribution. Using the model (c) parameterization, if N is negative binomial p, r then

$$\Pr(N \le k) = \text{BETADIST}(p; r, k+1) := \frac{1}{B(r, k+1)} \int_0^p u^{r-1}(1-u)^k du$$

where B is the complete beta function. See Johnson, Kotz and Kemp [24, Eqn. 5.31] for a derivation. BETADIST is the Excel beta cumulative distribution function.

The name negative binomial comes from an analogy with the binomial. A binomial variable has parameters n and p, mean np and variance npq, where $p + q = 1$. It is a sum of n independent Bernoulli variables B where $\Pr(B = 1) = p$ and $\Pr(B = 0) = q = 1 - p$. The MGF for a binomial is $(q + pe^{\zeta})^n$ and the probabilities are derived from the binomial expansion of the MGF. By analogy the negative binomial can be defined in terms of the negative binomial expansion of $(Q - Pe^{\zeta})^{-k}$ where $Q = 1 + P$, $P > 0$ and $k > 0$.

For the actuary there are two distinct ways of looking at the negative binomial which give very different results and it is important to understand these two views. First there is the contagion view, where the mixing distribution G has mean n and variance c producing a negative binomial with mean n and variance $n(1 + cn)$. (In fact G is a gamma with model (a) parameters $\alpha = r$ and $\beta = 1/r$.) The word contagion was used by Heckman and Meyers [17] and is supposed to indicate a "contagion" of claim propensity driven by common shock uncertainty, such as claim inflation, economic activity, or weather. Here the variance grows with the square of n and the coefficient of variation tends to $\sqrt{c} > 0$ as $n \to \infty$. Secondly, one can consider an over-dispersed family of Poisson variables with mean n and variance vn for some $v > 1$. We call v the variance multiplier. Now the coefficient of variation tends to 0 as $n \to \infty$. The notion of over-dispersion and its application in modeling is discussed in Clark and Thayer [5] and Verrall [34].

Table 3.2: Parameterizations of the Negative Binomial Distribution

Model	**Density**	**MGF**	**Mean**	**Var**
(a) α, β	$\binom{\alpha+x-1}{x}\left(\frac{\beta}{1+\beta}\right)^x\left(\frac{1}{1+\beta}\right)^\alpha$	$(1-\beta(e^t-1))^{-\alpha}$	$\alpha\beta$	$\alpha\beta^2$
(b) P, k	$\binom{k+x-1}{x}\left(\frac{P}{Q}\right)^x\left(\frac{Q-P}{Q}\right)^k$	$(Q-Pe^t)^{-k}$	kP	kPQ
(c) p, $r>0$	$\binom{r+x-1}{x}p^r q^x$	$\frac{p^r}{(1-qe^s)^r}$	rq/p	rq/p^2

$Q = P+1$, $q = 1-p$, $0<p<1$ and $r>0$, and $P = 1/(\beta+1)$.

Table 3.3: Fitting the Negative Binomial Distribution

Model	**Parameters**	**Variance Multiplier**		**Contagion**	
		Scale	**Shape**	**Scale**	**Shape**
(a)	r, β	$r = m/(v-1)$	$\beta = v-1$	$r = 1/c$	$\beta = cn$
(b)	k, P	$k = m/(v-1)$	$P = v-1$	$k = 1/c$	$P = cn$
(c)	r, p	$r = m/(v-1)$	$p = 1/v$	$r = 1/c$	$p = 1/(1+cn)$

3.4 Aggregate Distributions

Let $A = X_1 + \cdots + X_N$ be an aggregate distribution, where N is a G-mixed Poisson and X_i are iid severity random variables. Then, proceeding by analogy with 3.7, we have

$$\begin{aligned} M_A(\zeta) &= \mathrm{E}(\exp(\zeta(X_1 + \cdots X_N))) \\ &= \mathrm{E}\mathrm{E}(\exp(\zeta(X_1 + \cdots X_N))|N) \\ &= \mathrm{E}(\mathrm{E}(\exp(\zeta X_1)^N)) \\ &= \mathrm{E}(\mathrm{E}(M_X(\zeta)^N|G)) \\ &= \mathrm{E}(\exp(nG(M_X(\zeta) - 1))) \\ &= M_G(n(M_X(\zeta) - 1)) \end{aligned} \tag{3.12}$$

Thus

$$\mathrm{E}(A) = M_A'(0) = nM_G'(0)M_X'(0) = n\mathrm{E}(X) \tag{3.13}$$

and

$$\begin{aligned} \mathrm{E}(A^2) &= M_A''(0) \\ &= n^2 M_G''(0)M_X'(0)^2 + nM_G'(0)M_X''(0) \\ &= n^2\mathrm{E}(G^2)\mathrm{E}(X)^2 + n\mathrm{E}(X^2). \end{aligned} \tag{3.14}$$

Hence, using the fact that $\mathrm{E}(G^2) = 1 + c$, we get

$$\begin{aligned} \mathrm{Var}(A) &= n^2\mathrm{E}(G^2)\mathrm{E}(X)^2 + n\mathrm{E}(X^2) - n^2\mathrm{E}(X)^2 \\ &= n^2 c\mathrm{E}(X)^2 + n\mathrm{E}(X^2) \\ &= (\mathrm{Var}(N) - \mathrm{E}(N))\mathrm{E}(X)^2 + \mathrm{E}(N)\mathrm{E}(X^2) \\ &= \mathrm{Var}(N)\mathrm{E}(X)^2 + \mathrm{E}(N)\mathrm{Var}(X). \end{aligned} \tag{3.15}$$

Continuing along the same vein we get

$$\begin{aligned} \mathrm{E}(A^3) =& \mathrm{E}(N)\mathrm{E}(X^3) + \mathrm{E}(N^3)\mathrm{E}(X)^3 + 3\mathrm{E}(N^2)\mathrm{E}(X)\mathrm{E}(X^2) \\ &- 3\mathrm{E}(N)\mathrm{E}(X)\mathrm{E}(X^2) - 3\mathrm{E}(N^2)\mathrm{E}(X)^3 + 2\mathrm{E}(N)\mathrm{E}(X)^3. \end{aligned} \tag{3.16}$$

and so we can compute the skewness of A—remember that $\mathrm{E}[(A - \mathrm{E}(A))^3] = \mathrm{E}(A^3) - 3\mathrm{E}(A^2)\mathrm{E}(A) + 2\mathrm{E}(A)^3$. Further moments can be computed using derivatives of the moment generating function.

Having computed the mean, CV and skewness of the aggregate using these equations we can use the method of moments to fit a shifted lognormal or shifted gamma distribution. We turn next to a description of these handy distributions.

3.5 Shifted Gamma and Lognormal Distributions

The shifted gamma and shifted lognormal distributions are versatile three parameter distributions whose method of moments parameters can be conveniently computed by closed formula. The examples below show that they also provide a very good approximation to aggregate loss distributions. The shifted gamma approximation to an aggregate is discussed in Bowers et al. [4]. Properties of the shifted gamma and lognormal distributions, including the method of moments fit parameters, are also shown in Daykin et al. [7, Chapter 3].

Let L have a lognormal distribution. Then $S = s \pm L$ is a shifted lognormal, where s is a real number. Since s can be positive or negative and since L can equal $s + L$ or $s - L$, the shifted lognormal can model distributions which are positively or negatively skewed, as well as distributions supported on the negative reals. The key facts about the shifted lognormal are shown in Table 3.4. The variable η is a solution to the cubic equation

$$\eta^3 + 3\eta - \gamma = 0$$

where γ is the skewness.

Let G have a gamma distribution. Then $T = s \pm G$ is a shifted gamma distribution, where s is a real number. Table 3.1 shows some common parametric forms for the gamma distribution. The key facts about the shifted gamma distribution are also shown in Table 3.4.

The exponential is a special case of the gamma where $\alpha = 1$. The χ^2 is a special case where $\alpha = k/2$ and $\beta = 2$ in the Excel parameterization. The Pareto is a mixture of exponentials where the mixing distribution is gamma.

Table 3.4: Shifted Gamma and Lognormal Distributions

Item	**Shifted Gamma**	**Shifted Lognormal**
Parameters	s, α, θ	s, μ, σ
Mean m	$s + \alpha\theta$	$s + \exp(\mu + \sigma^2/2)$
Variance	$\alpha\theta^2$	$m^2 \exp(\sigma^2 - 1)$
CV, ν	$\sqrt{\alpha}\beta/\gamma m$	$\exp((\sigma^2 - 1)/2)$
Skewness, γ	$2/\sqrt{\alpha}$	$\gamma = \nu(\nu^2 + 3)$
Method of Moments Parameters		
η	n/a	$\eta = u - 1/u$ where $u^3 = \sqrt{\gamma^2 + 4}/2 + \gamma/2$
Shift variable, s	$m - \alpha\beta$	$m(1 - \nu\eta)$
α or σ	$4/\gamma^2$	$\sqrt{\ln(1 + \eta^2)}$
β or μ	$m\nu\gamma/2$	$\ln(m - s) - \sigma^2/2$

3.6 Excess Frequency Distributions

Given a ground-up claim count distribution N, what is the distribution of the number of claims exceeding a certain threshold? We assume that severities are independent and identically distributed and that the probability of exceeding the threshold is q. Define an indicator variable I which takes value 0 if the claim is below the threshold and the value 1 if it exceeds the threshold. Thus $\Pr(I = 0) = p = 1 - q$ and $\Pr(I = 1) = q$. Let M_N be the moment generating function of N and N' is the number of claims in excess of the threshold. By definition we can express N' as an aggregate

$$N' = I_1 + \cdots + I_N. \tag{3.17}$$

Thus the moment generating function of N' is

$$\begin{aligned} M_{N'}(\zeta) &= M_N(\log(M_I(\zeta))) \\ &= M_N(\log(p + qe^{\zeta})) \end{aligned} \tag{3.18}$$

Using indicator variables I is called p-thinning by Grandell [15].

Here are some examples.

Let N be Poisson with mean n. Then

$$M_{N'}(\zeta) = \exp(n(p + qe^{\zeta} - 1)) = \exp(qn(e^{\zeta} - 1)) \tag{3.19}$$

so N' is also Poisson with mean qn—the simplest possible result.

Next let N be a G-mixed Poisson. Thus

$$\begin{aligned} M_{N'}(\zeta) &= M_N(\log(p + qe^{\zeta})) \\ &= M_G(n(p + qe^{\zeta} - 1)) \\ &= M_G(nq(e^{\zeta} - 1)). \end{aligned} \tag{3.20}$$

Hence N' is also a G-mixed Poisson with lower underlying claim count nq in place of n.

In particular, if N has a negative binomial with parameters P and c (mean cP, $Q = 1 + P$, moment generating function $M_N(\zeta) = (Q - Pe^{\zeta})^{-1/c}$), then N' has parameters qP and c. If N has a Poisson-inverse Gaussian distribution with parameters μ and β, so

$$M_N(\zeta) = \exp\left(-\mu(\sqrt{1 + 2\beta(e^{\zeta} - 1)} - 1)\right),$$

then N is also Poisson inverse Gaussian with parameters μq and βq.

In all cases the variance of N' is lower than the variance of N and N' is closer to Poisson than N in the sense that the variance to mean ratio has decreased. For the general G-mixed Poisson the ratio of variance to mean decreases from $1 + cn$ to $1 + cqn$. As $q \to 0$ the variance to mean ratio approaches 1 and N' approaches a Poisson distribution. The fact that N' becomes Poisson is called the law of small numbers.

3.7 Negative Multinomial Distribution and Related Frequency Distributions*

When we consider mixed Poisson distributions we often regard G as carrying inter-risk correlation, or more evocatively "contagion", information about weather, the state of the economy and inflation, gas prices etc. Hence if we have two related frequency variables N_1 and N_2 we should expect to use the same G and produce a bivariate mixed Poisson where, conditional on $G = g$, N_i has a Poisson distribution with mean $n_i g$ and N_1 and N_2 are conditionally independent. The MGF of such a distribution will be

$$\begin{aligned} M(\zeta_1, \zeta_2) &= \mathrm{E}(e^{\zeta_1 N_1 + \zeta_2 N_2}) \\ &= \mathrm{E}(\mathrm{E}(e^{\zeta_1 N_1 + \zeta_2 N_2}|G)) \\ &= \mathrm{E}_G(\mathrm{E}(e^{\zeta_1 N_1}|G)\mathrm{E}(e^{\zeta_2 N_2}|G)) \\ &= \mathrm{E}_G(\exp(G(n_1(e^{\zeta_1}-1)+n_2(e^{\zeta_2}-1)))) \\ &= M_G(n_1(e^{\zeta_1}-1)+n_2(e^{\zeta_2}-1)). \end{aligned} \tag{3.21}$$

For example, if G is a gamma random variable with MGF

$$M_G(\zeta) = (1-\beta\zeta)^{-k} \tag{3.22}$$

(mean $k\beta$, variance $k\beta^2$) we get a bivariate frequency distribution with MGF

$$\begin{aligned} M(\zeta_1, \zeta_2) &= [1-\beta(n_1(e^{\zeta_1}-1)+n_2(e^{\zeta_2}-1))]^{-k} \\ &= [1+\beta\sum_i n_i - \beta\sum_i n_i e^{\zeta_i}]^{-k} \\ &= (Q - \sum_i P_i e^{\zeta_i})^{-k} \end{aligned} \tag{3.23}$$

where $P_i = \beta n_i$, $P = \sum_i P_i$ and $Q = 1 + P$. Equation (3.23) is the moment generating function for a negative multinomial distribution, as defined in Johnson, Kotz and Kemp [23]. The negative multinomial distribution has positively correlated marginals as expected given its construction with a common contagion G.

The form of the moment generating function for negative multinomial distribution can be generalized allowing us to construct multivariate frequency distributions $(N_1, \ldots, N_t)$ where

1. Each N_i is a negative binomial.

2. The sum $N_1 + \cdots + N_t$ under the multivariate distribution is also negative binomial. (In general, the sum of independent negative binomials will not be negative binomial.)

3. The N_i are correlated.

We will call such multivariate frequencies, with common mixing distributions, G-mixed multivariate Poisson distributions.

3.7.1 Evolution of Claims Over Time*

Here is an application of the NMN distribution. If A is an aggregate distribution representing ultimate losses we may want to determine a decomposition $A = \sum_t D_t$ into a sum of losses paid at time t for $t = 1, \ldots, T$.

If $A = X_1 + \cdots + X_N$ has a compound Poisson distribution then such a decomposition is easy to arrange. Let π_t be the expected proportion of ultimate losses paid at time t, so $\sum_{t=1}^{t=T} \pi_t = 1$. By definition we mean

$$\mathrm{E}(D_t) = \pi_t \mathrm{E}(A). \tag{3.24}$$

(Equation (3.24) is a different assumption to

$$\mathrm{E}(D_t) = \pi_t \mathrm{E}(A|\text{information available at } t-1) = \pi_t A_{t-1},$$

which is closer to the problem actually faced by the reserving actuary. Our π_t's are prior estimates assumed known at time 0. These types of differences have interesting implications for actuarial methods and they are explored further in Mack [28].) Now we seek a decomposition

$$A = D_1 + D_2 + \cdots + D_T \tag{3.25}$$

but we know only (3.24). The simplest approach is to assume that severity X is independent of time and that $\pi_t n$ of the total n claims are paid at time t. If we

further assume that the number of claims paid at time t is also Poisson, then the moment generating function of $D_1 + \cdots + D_T$ is given by

$$\begin{aligned} M_{D_1+\cdots+D_T}(\zeta) &= \prod_t \exp(\pi_t n(M_X(\zeta) - 1)) \\ &= \exp(n(\sum_t \pi_t M_X(\zeta) - 1)) \\ &= \exp(n(M_X(\zeta) - 1)) \\ &= M_A(\zeta). \end{aligned} \tag{3.26}$$

Thus we have a very simple decomposition for (3.25): the individual D_t are independent compound Poisson variables with expected claim count $\pi_t n$ and severity distribution X.

Moving one step further, it is often observed in practice that average severity increases with t so the assumption that X is fixed for all t is unrealistic. It may be better to assume that losses which close at time t are samples of a random variable X_t. As above, we assume that the expected number of such losses is $\pi'_t n$ where n is the expected ultimate number of claims, and π'_t adjusts the original π_t for the difference in average severity $\mathrm{E}(X)$ vs. $\mathrm{E}(X_t)$. Now

$$\begin{aligned} M_{D_1+\cdots+D_T}(\zeta) &= \prod_t \exp(\pi'_t n(M_{X_t}(\zeta) - 1)) \\ &= \exp(n(\sum_t \pi'_t M_{X_t}(\zeta) - 1)) \\ &= \exp(n(M_{X'}(\zeta) - 1)) \\ &= M_A(\zeta) \end{aligned} \tag{3.27}$$

where X' is a mixture of the X_t with weights π'_t. Equation (3.27) is a standard result in actuarial science, see Bowers et al. [4].

If we try to replicate the compound Poisson argument using a negative binomial distribution for N we will clearly fail. However if X is defined as a mixture of X_t with weights π_t, as before, then we can write

$$M_{D_1,\ldots,D_T}(\zeta_1, \ldots, \zeta_T) = (Q - \sum_t P\pi_t M_{X_t}(\zeta_t))^{-k} \tag{3.28}$$

and so

$$M_A(\zeta) = M_{D_1,\ldots,D_T}(\zeta,\ldots,\zeta) = (Q - \sum_t P_t M_{X_t}(\zeta))^{-k} = (Q - PM_X(\zeta))^{-k} \tag{3.29}$$

where $P_t := \pi_t P$. Equation (3.28) is the MGF for a negative multinomial distribution, as defined in the previous section and Johnson, Kotz and Kemp [23]. As we have seen the negative multinomial distribution has positively correlated marginals, in line with our prior notions of liability dynamics. It therefore provides a good model for the decomposition of ultimate losses into losses paid each period.

3.7.2 Related Multivariate Frequency Distributions*

We can use the same trick with other mixing distributions than the gamma. The Poisson inverse Gaussian (PIG) distribution is an inverse Gaussian mixture of Poissons, just as the negative binomial distribution is a gamma mixture. The MGF is

$$M(\zeta) = \exp(-\tau(\sqrt{1+\beta(e^\zeta - 1)} - 1)). \tag{3.30}$$

The mean is $\tau\beta$ and the variance is $\tau\beta(1+\beta)$. We can define a multivariate PIG (MPIG) by

$$M(\zeta_1,\ldots,\zeta_T) = \exp(-\tau(\sqrt{1+\sum \beta_i(e^{\zeta_i} - 1)} - 1)). \tag{3.31}$$

Sichel's distribution is an generalized-inverse Gaussian mixture of Poissons. The MGF is

$$M(\zeta) = \frac{K_\gamma(\omega\sqrt{1-2\beta(e^\zeta - 1)})}{K_\gamma(\omega)(1-2\beta(e^\zeta - 1))^{\gamma/2}}. \tag{3.32}$$

The mean and variance are given in Johnson, Kotz and Kemp [24, page 456]. Clearly we can apply the same techniques to get another multivariate frequency distribution.

The Poisson-Pascal distribution is a Poisson-stopped sum of negative binomials. It has moment generating function

$$M(\zeta) = \exp(\theta((1 - P(e^{\zeta} - 1))^{-k} - 1)) \tag{3.33}$$

and so will also yield another multivariate family. The mean and variance are given by

$$\mu = \theta k P \tag{3.34}$$

$$\mu_2 = \theta k P(Q + kP). \tag{3.35}$$

3.7.3 Excess count interpretation of G-mixed multivariate Poisson distributions*

The reader has probably realized that a G-mixed multivariate Poisson seems closely related to a single G-mixed Poisson and a series of indicator variables, combining results from the previous sub-sections with Section 3.6. Let N be G-mixed Poisson with parameter n and $\mathrm{Var}(G) = c$. Let (N_1, N_2) be G-mixed bivariate Poisson with parameters n_1 and n_2 and the same G, so the MGF of (N_1, N_2) is

$$M_1(\zeta_1, \zeta_2) = M_G(n_1(e^{\zeta_1} - 1) + n_2(e^{\zeta_2} - 1)). \tag{3.36}$$

Finally let (I, J) be a bivariate distribution supported on $\{0, 1\} \times \{0, 1\}$ with

$$\begin{aligned}
\Pr(I = 0, J = 0) &= p_{00} \\
\Pr(I = 1, J = 0) &= p_{10} \\
\Pr(I = 0, J = 1) &= p_{01} \\
\Pr(I = 1, J = 1) &= p_{11}
\end{aligned}$$

and $\sum p_{ij} = 1$.

We can define a new bivariate distribution from (I, J) and N as

$$(M_1, M_2) = (I_1, J_1) + \cdots + (I_N, J_N). \tag{3.37}$$

The MGF of (M_1, M_2) is

$$M_2(\zeta_1, \zeta_2) = M_G(n(p_{11}e^{\zeta_1+\zeta_2} + p_{10}e^{\zeta_1} + p_{01}e^{\zeta_2} + p_{00}). \tag{3.38}$$

Thus, if $p_{11} = 0$ we see the single-frequency sum of the bivariate (M_1, M_2) is actually a G-mixed bivariate Poisson. If $p_{00} = 0$ then $n = n_1 + n_2$, otherwise $(1 - p_{00})n = n_1 + n_2$ and there are some extraneous "zero" claims. However, if $p_{11} \neq 0$ then the single frequency sum is not a G-mixed bivariate Poisson.

Here is an interesting interpretation and application of (I, J). We can regard I as an indicator of whether a claim has been reported at time t and J and indicator of whether the claim is closed. Then

$\Pr(I = 0, J = 0) =$ meaningless

$\Pr(I = 1, J = 0) =$ reported claim which closes without payment

$\Pr(I = 0, J = 1) =$ claim not yet reported which closes with payment

$\Pr(I = 1, J = 1) =$ claim reported and closed with payment.

Combining with a distribution N of ultimate claims we can use (3.37) to produce $(M_1, M_2) = (I_1 + \cdots + I_N, J_1 + \cdots + J_N)$—a bivariate distribution of (claims reported at time t, ultimate number of claims)! Note the value $(0, 0)$ is a meaningless annoyance (it scales n) and we assume $p_{00} = 0$. The three other parameters can easily be estimated using standard actuarial methods.

Given such a bivariate and a known number of claims reported we can produce a posterior distribution of ultimate claims. Furthermore, in all these techniques we can extend the simple count indicators (I, J) to be the distribution of case incurred losses and ultimate losses. Then we would get a bivariate distribution of case incurred to date and ultimate losses. I believe there is a lot of useful information that could be wrought from these methods and that they deserve further study. They naturally give confidence intervals on reserve ranges, for example.

We end with a numerical example illustrating the theory we have developed and showing another possible application. Rather than interpreting p_{ij} as reported

and ultimate claims we could interpret them as claims from line A and line B, where there is some expectation these claim would be correlated. For example A could be auto liability and B workers compensation for a trucking insured. Let $c = 0.02$ be the common contagion and $n = 250$. Then let

$$\begin{aligned}
\Pr(I = 0, J = 0) &= 0 \\
\Pr(I = 1, J = 0) &= 0.45 \\
\Pr(I = 0, J = 1) &= 0.05 \\
\Pr(I = 1, J = 1) &= 0.50.
\end{aligned}$$

We interpret I as indicating a workers compensation claim and J as indicating an auto liability claim. The distribution says that when there is an auto liability claim ($J = 1$) there is almost always an injury to the driver, resulting in a workers compensation claim ($I = 1$). However, there are many situations where the driver is injured but there is no liability claim—such as back injuries. Overall we expect $250(0.45 + 0.50) = 237.5$ workers compensation claims and $250(0.05 + 0.5) = 137.5$ auto liability claims and 250 occurrences.

We will consider the single-frequency bivariate distribution and the negative multinomial. We have seen that the negative multinomial distribution will be slightly different because $p_{11} \neq 0$. The appropriate parameters are $n_1 = 250(p_{10} + p_{11}) = 237.5$ and $n_1 = 250(p_{01} + p_{11}) = 137.5$. Figure 3.1 shows the negative multinomial bivariate (top plot) and the single-frequency bivariate aggregate of (I, J) (bottom plot). Because of the correlation between I and J, $p_{11} = 0.5$, the lower plot shows more correlation in aggregates and the conditional distributions have less dispersion. Figure 3.2 shows the two marginal distributions, which are negative binomial $c = 0.02$ and mean 237.5 and 137.5 respectively, the sum of these two variables assuming they are independent (labelled "independent sum"), the sum assuming the negative multinomial joint distribution ("NMN Sum") which is identical to a negative binomial with $c = 0.02$ and mean $350 = 237.5 + 137.5$, the total number of claims from both lines, and finally, the sum with dependent (I, J) ("bivariate sum"). The last sum is not the same as the negative binomial sum; it has a different MGF.

Figure 3.2 also shows the difference between the sum of two independent negative binomials with means n_1 and n_2 and contagion c and a negative binomial with mean $n_1 + n_2$ and contagion c. The difference is clearly very material in the tails and is an object lesson to modelers who subdivide their book into homogeneous parts but then add up those parts assuming independence. Such an approach is *wrong* and must be avoided.

As the contagion c increases the effects of G-mixing dominate and the difference between the two bivariate distributions decreases, and conversely as c decreases to zero the effect is magnified. The value $c = 0.02$ was selected to balance these two effects.

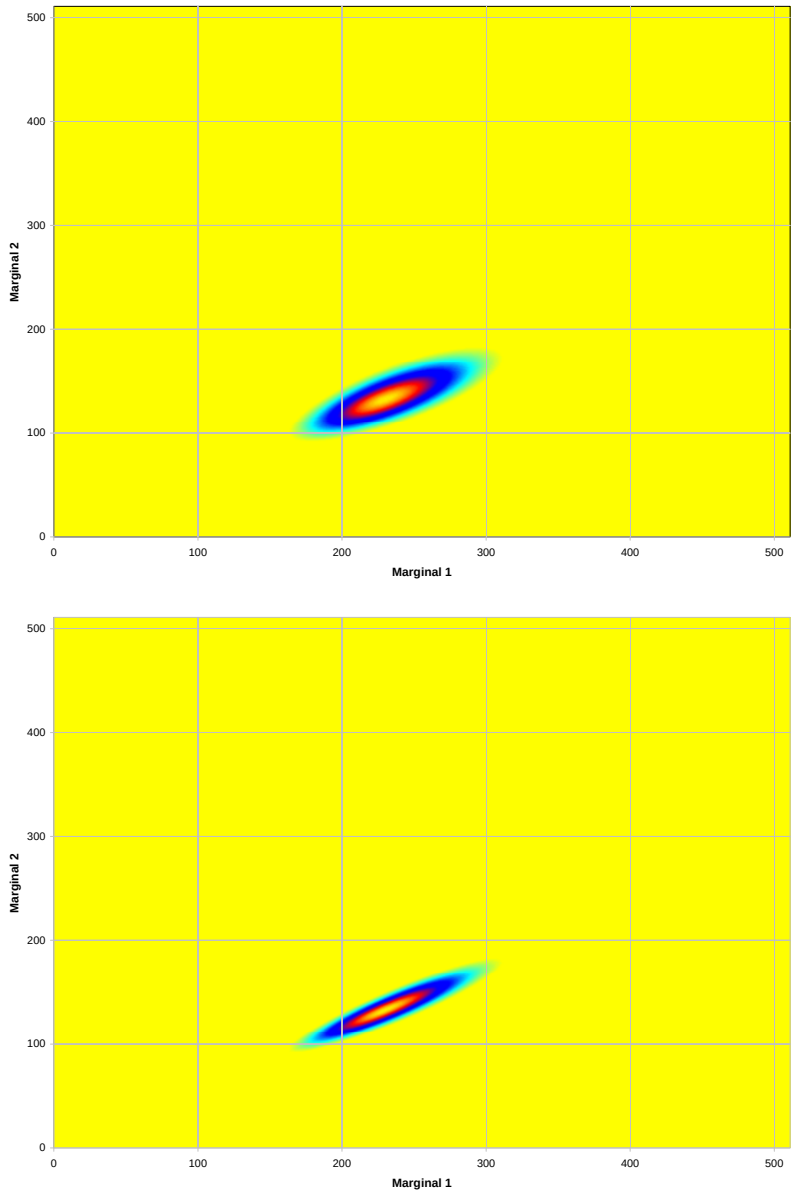

Figure 3.1: Comparison of negative multinomial (top) and single frequency bivariate claim count (bottom) bivariate distributions.

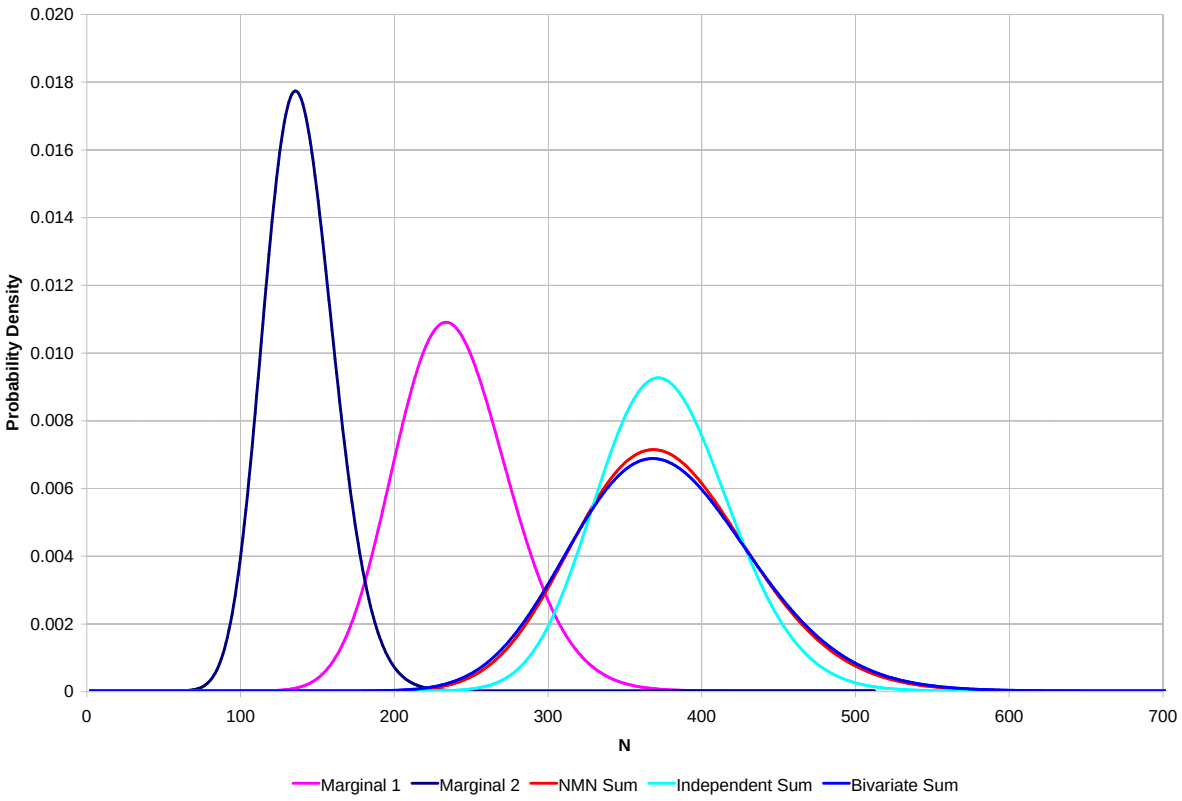

Figure 3.2: Comparison of negative multinomial and single frequency bivariate claim count marginal and total distributions.

3.8 Fast Fourier Transforms

The FFT method is a miraculous technique for computing aggregate distributions. It is especially effective when the expected claim count is relatively small and the underlying severity distribution is bounded. These assumptions are true for many excess of loss reinsurance treaties, for example. Thus the FFT is very useful when quoting excess layers with annual aggregate deductibles or other variable features. The FFT provides a discrete approximation to the moment generating function.

To use the FFT method, first "bucket" (or quantize) the severity distribution into a density vector $\mathbf{x} = (x_1, \ldots, x_m)$ whose length m is a power of two $m = 2^n$. Here

$$x_i = \Pr((i - 1/2)b < X < (i + 1/2)b) \tag{3.39}$$

$$x_1 = \Pr(X < b/2), \quad x_m = \Pr(X > (m - 1/2)b) \tag{3.40}$$

for some fixed b. We call b the bucket size. Note $\sum_i x_i = 1$ by construction. The FFT of the $m \times 1$ vector $\mathbf{x}$ is another $m \times 1$ vector $\hat{\mathbf{x}}$ whose jth component is

$$\sum_{k=0}^{2^n-1} x_k \exp(2\pi i j k / 2^n). \tag{3.41}$$

The coefficients of $\hat{\mathbf{x}}$ are complex numbers. It is also possible to express $\hat{\mathbf{x}} = \mathbf{F}\mathbf{x}$ where $\mathbf{F}$ is an appropriate matrix of complex roots of unity, so there is nothing inherently mysterious about a FFT. The trick is that there exists a very efficient algorithm for computing (3.41). Rather than taking time proportional to m^2, as one would expect, it can be computed in time proportional to $m \log(m)$. The difference between $m \log(m)$ and m^2 time is the difference between practically possible and practically impossible.

You can use the inverse FFT to recover $\mathbf{x}$ from its transform $\hat{\mathbf{x}}$. The inverse FFT is computed using the same equation (3.41) as the FFT except there is a minus sign in the exponent and the result is divided by 2^n. Because the equation is essentially the same, the inversion process can also be computed in $m \log(m)$ time.

The next step is magic in actuarial science. Remember that if N is a G-mixed Poisson and $A = X_1 + \cdots + X_N$ is an aggregate distribution then

$$M_A(\zeta) = M_G(n(M_X(\zeta) - 1)). \tag{3.42}$$

Using FFTs you can replace the *function* M_X with the discrete approximation *vector* $\hat{\mathbf{x}}$ and compute

$$\hat{\mathbf{a}} = M_G(n(\hat{\mathbf{x}} - 1)) \tag{3.43}$$

component-by-component to get an approximation vector to the function M_A. You can then use the inverse FFT to recover an discrete approximation **a** of A from $\hat{\mathbf{a}}$! See Wang [37] for more details.

Similar tricks are possible in two dimensions—see Press et al. [31] and Homer and Clark [20] for a discussion.

The FFT allows us to use the following very simple method to qualitatively approximate the density of an aggregate of dependent marginals $X_1, \ldots, X_n$ given a correlation matrix Σ. First use the FFT method to compute the sum S' of the X_i as though they were independent. Let $\mathrm{Var}(S') = \sigma'^2$ and let σ^2 be the variance of the sum of the X_i implied by Σ. Next use the FFT to add a further "noise" random variable N to S' with mean zero and variance $\sigma^2 - \sigma'^2$. Two obvious choices for the distribution of N are normal or shifted lognormal. Then $S' + N$ has the same mean and variance as the sum of the dependent variables X_i. The range of possible choices for N highlights once again that knowing the marginals and correlation structure is not enough to determine the whole multivariate distribution. It is an interesting question whether all possible choices of N correspond to actual multivariate structures for the X_i and conversely whether all multivariate structures correspond to an N. (It is easy to use MGFs to deconvolve N from the true sum using Fourier methods; the question is whether the resulting "distribution" is non-negative.)

Heckman and Meyers [17] used Fourier transforms to compute aggregate distributions by numerically integrating the characteristic function. Direct inversion of the Fourier transform is also possible using FFTs. The application of FFTs is not completely straight forward because of certain aspects of the approximations involved. The details are very clearly explained in Menn and Rachev [29]. Their method allows the use of FFTs to determine densities for distributions which have analytic MGFs but not densities—notably the class of stable distributions.

3.9 Correlated Aggregate Distributions

Here we extend some of the ideas in Section 3.7.3 from plain frequency distributions to aggregate distributions. Begin with bivariate aggregate distributions. There are two different situations which commonly arise. First we could model a bivariate severity distribution and a univariate count distribution:

$$(A, B) = (X_1, Y_1) + \cdots + (X_N, Y_N). \tag{3.44}$$

Equation (3.44) arises naturally as the distribution of losses and allocated expense, ceded and retained losses, reported and ultimate claims, and in many other situations. Secondly we could model

$$(A, B) = (X_1 + \cdots + X_M, Y_1 + \cdots + Y_N) \tag{3.45}$$

where X_i and Y_j are independent severities and (M, N) is a bivariate frequency distribution. (3.45) could be used to model losses in a clash policy.

We will use the following notation. $A = X_1 + \cdots + X_M$ and $B = Y_1 + \cdots + Y_N$ are two aggregate distributions, with X_i iid and Y_j iid, but neither X and Y nor M and N necessarily independent. Let $\mathrm{E}(X) = x$ and $\mathrm{E}(Y) = y$, $\mathrm{Var}(X) = v_x$ and $\mathrm{Var}(Y) = v_y$. Let $\mathrm{E}(M) = m$, $\mathrm{E}(N) = n$, c be the contagion of M and d that of N. Hence $\mathrm{Var}(M) = m(1 + cm)$ and $\mathrm{Var}(N) = n(1 + dn)$.

Will now calculate the correlation coefficient between A and B in four situations.

3.9.1 Correlated Severities, Single Frequency

Assume that the bivariate severity distribution (X, Y) has moment generating function $M_{(X,Y)}(\zeta, \tau)$. Also assume that the claim count distribution N is a G-mixed Poisson. Then, just as for univariate aggregate distributions, the MGF of the bivariate aggregate (A, B) is

$$M_{(A,B)}(\zeta, \tau) = M_G(n(M_{(X,Y)}(\zeta, \tau) - 1)). \tag{3.46}$$

Therefore, since $\mathrm{E}(G) = 1$ and $\mathrm{E}(G^2) = 1 + c$,

$$\begin{aligned} E(AB) &= \frac{\partial^2 M_{(A,B)}}{\partial\zeta\partial\tau}\Big|_{(0,0)} \\ &= M_G''(0)n^2\frac{\partial M_{(X,Y)}}{\partial\zeta}\frac{\partial M_{(X,Y)}}{\partial\zeta} + M_G'(0)n\frac{\partial^2 M_{(X,Y)}}{\partial\zeta\partial\tau} \\ &= (1+c)n^2xy + n\mathrm{E}(XY) \\ &= (1+c)n^2xy + n\mathrm{Cov}(X,Y) + nxy. \end{aligned} \tag{3.47}$$

The value of $\mathrm{Cov}(X, Y)$ will depend on the particular bivarate severity distribution.

For example, suppose that Z represents ground up losses, X represents a retention to a and Y losses excess of a (per ground up claim), so $Z = X + Y$. Then (X, Y) is a bivariate severity distribution. Since Y is zero when $Z \le a$ we have $\mathrm{Cov}(X, Y) = (a - x)y$.

3.9.2 Bivariate Frequency

The second method for generating correlated aggregate distributions is to use a bivariate frequency distribution. So, suppose (M, N) has a G-mixed bivariate Poisson distribution. The variance of A is given by Equation (3.15). To compute the covariance of A and B write the bivariate MGF of (A, B) as

$$M_{(A,B)}(\zeta,\eta) = M(\zeta,\eta) = M_G(m(M_X(\zeta)-1)+n(M_Y(\eta)-1)) = M_G(\psi(\zeta,\eta)) \tag{3.48}$$

where the last equality defines ψ. Then, evaluating at the partial derivatives at zero, we get

$$\begin{aligned} \mathrm{E}(AB) &= \frac{\partial^2 M}{\partial\zeta\partial\eta} \\ &= \frac{\partial^2 M_G}{\partial t^2}\frac{\partial\psi}{\partial\zeta}\frac{\partial\psi}{\partial\eta} + \frac{\partial M_G}{\partial t}\frac{\partial^2\psi}{\partial\zeta\partial\eta} \\ &= (1+c)mxny. \end{aligned} \tag{3.49}$$

Hence

$$\mathrm{Cov}(A,B) = \mathrm{E}(AB) - \mathrm{E}(A)\mathrm{E}(B) = cmnxy. \tag{3.50}$$

3.9.3 Parameter Uncertainty

It is common for actuaries to work with point estimates as though they are certain. In reality there is a range around any point estimate. We now work through one possible implication of such parameter uncertainty. We will model $\mathrm{E}(A) = R$ and $\mathrm{E}(B) = S$ with R and S correlated random variables, and A and B conditionally independent given R and S. We will assume for simplicity that the severities X and Y are fixed and that the uncertainty all comes from claim counts. The reader can extend the model to varying severities as an exercise. R and S pick up uncertainty in items like the trend factor, tail factors and other economic variables, as well as the natural correlation induced through actuarial methods such as the Bornheutter-Ferguson.

Suppose $\mathrm{E}(R) = r$, $\mathrm{E}(S) = s$, $\mathrm{Var}(R) = v_r$, $\mathrm{Var}(S) = v_s$ and let ρ be the correlation coefficient between R and S.

By (3.15) the conditional distribution of $A|R$ is a mixed compound Poisson distribution with expected claim count R/x and contagion c. Therefore the conditional variance is

$$\begin{aligned} \mathrm{Var}(A|R) &= \mathrm{E}(M|R)\mathrm{Var}(X) + \mathrm{Var}(M|R)\mathrm{E}(X)^2 \\ &= R/x v_x + R/x(1 + cR/x)x^2 \\ &= xR(1 + v_x/x^2) + cR^2, \end{aligned} \tag{3.51}$$

and the unconditional variance of A is

$$\begin{aligned} \mathrm{Var}(A) &= \mathrm{E}(\mathrm{Var}(A|R)) + \mathrm{Var}(\mathrm{E}(A|R)) \\ &= \mathrm{E}(xR(v_x/x^2 + 1) + cR^2) + \mathrm{Var}(R) \\ &= xr(v_x/x^2 + 1) + c(v_r + r^2) + v_r. \end{aligned} \tag{3.52}$$

Next, because A and B are conditionally independent given R and S,

$$\begin{aligned} \mathrm{Cov}(A, B) &= \mathrm{E}(\mathrm{Cov}(A, B|R, S)) + \mathrm{Cov}(\mathrm{E}(A|R), \mathrm{E}(B|S)) \\ &= \mathrm{Cov}(R, S). \end{aligned} \tag{3.53}$$

Note Equation (3.53) is only true if we assume $A \neq B$.

3.9.4 Parameter Uncertainty and Bivariate Frequency

Finally, suppose $\mathrm{E}(A) = R$, $\mathrm{E}(B) = S$ with R and S correlated parameters *and* conditional on (R, S) suppose that (M, N) has a G-mixed bivariate Poisson distribution. By (3.50) $\mathrm{Cov}(A, B|R, S) = cRS$. The unconditional variances are as given in (3.52). The covariance term is

$$\begin{aligned}
\mathrm{Cov}(A, B) &= \mathrm{E}(\mathrm{Cov}(A, B|R, S)) + \mathrm{Cov}(\mathrm{E}(A|R), \mathrm{E}(B|S)) \\
&= c\mathrm{E}(RS) + \mathrm{Cov}(R, S) \\
&= (1 + c)\mathrm{Cov}(R, S) + crs \\
&= \rho\sqrt{v_r v_s}(1 + c) + crs. \qquad (3.54)
\end{aligned}$$

3.10 Severity is Irrelevant

In some cases the actual form of the severity distribution is essentially irrelevant to the shape of the aggregate distribution. Consider an aggregate with a G-mixed Poisson frequency distribution. If the expected claim count n is large and if the severity is tame (roughly tame means "has a variance"; any severity from a policy with a limit is tame; unlimited workers compensation may not be tame) then particulars of the severity distribution diversify away in the aggregate. Moreover the variability from the Poisson claim count component also diversifies away and the shape of the aggregate distribution converges to the shape of the frequency mixing distribution G. Another way of saying the same thing is that the normalized distribution of aggregate losses (aggregate losses divided by expected aggregate losses) converges in distribution to G.

We can prove these assertions using moment generating functions. Let X_n be a sequence of random variables with distribution functions F_n and let X another random variable with distribution F. If $F_n(x) \to F(x)$ as $n \to \infty$ for every point of continuity of F then we say F_n converges weakly to F and that X_n converges in distribution to F.

Convergence in distribution is a relatively weak form of convergence. A stronger form is convergence in probability, which means for all $\epsilon > 0$ $\Pr(|X_n - X| > \epsilon) \to 0$ as $n \to \infty$. If X_n

converges to X in probability then X_n also converges to X in distribution. The converse is false. For example, let $X_n = Y$ and X be binomial 0/1 random variables with $\Pr(Y = 1) = \Pr(X = 1) = 1/2$. Then X_n converges to X in distribution. However, since $\Pr(|X - Y| = 1) = 1/2$, X_n does not converge to X in probability.

It is a fact that X_n converges to X if the MGFs M_n of X_n converge to the MFG of M of X for all t: $M_n(t) \to M(t)$ as $n \to \infty$. See Feller [12] for more details. We can now prove the following theorem.

Proposition 1 *Let N be a G-mixed Poisson distribution with mean n, G with mean 1 and variance c, and let X be an independent severity with mean x and variance $x(1+\gamma^2)$. Let $A = X_1 + \cdots + X_N$ and $a = nx$. Then A/a converges in distribution to G, so*

$$\Pr(A/a < \alpha) \to \Pr(G < \alpha) \tag{3.55}$$

as $n \to \infty$. Hence

$$\sigma(A/a) = \sqrt{c + \frac{x(1+\gamma^2)}{a}} \to \sqrt{c}. \tag{3.56}$$

Proof: By (3.12)

$$M_A(\zeta) = M_G(n(M_X(\zeta) - 1)) \tag{3.57}$$

and so using Taylor's expansion we can write

$$\begin{aligned}
\lim_{n\to\infty} M_{A/a}(\zeta) &= \lim_{n\to\infty} M_A(\zeta/a) \\
&= \lim_{n\to\infty} M_G(n(M_X(\zeta/nx) - 1)) \\
&= \lim_{n\to\infty} M_G(n(M_X'(0)\zeta/nx + R(\zeta/nx))) \\
&= \lim_{n\to\infty} M_G(\zeta + nR(\zeta/nx))) \\
&= M_G(\zeta)
\end{aligned}$$

for some remainder function $R(t) = O(t^2)$. Note that the assumptions on the mean and variance of X guarantee $M_X'(0) = x = \mathrm{E}(X)$ and that the remainder term in Taylor's expansion actually is $O(t^2)$. The second part is trivial.

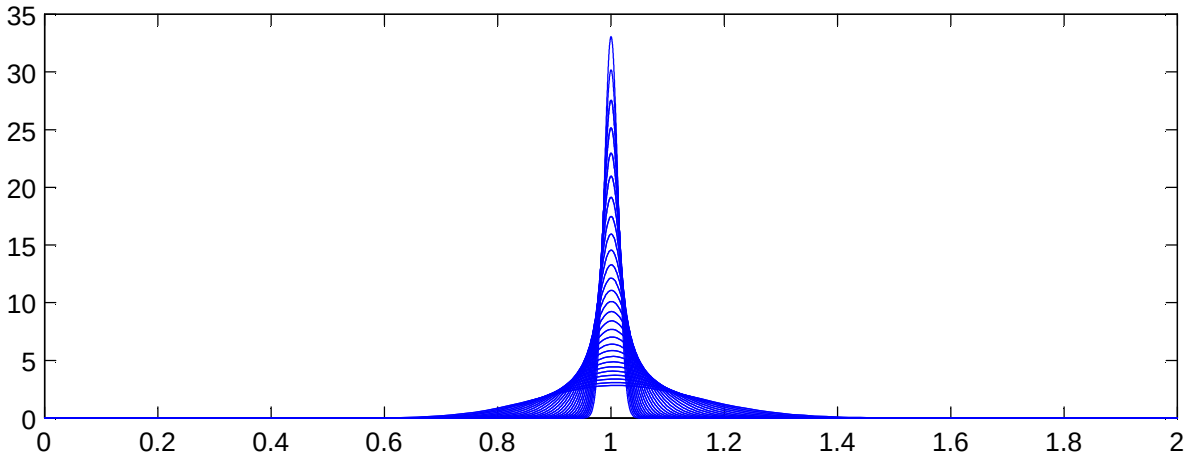

Figure 3.3: Theoretical distribution of scaled aggregate losses with no parameter or structure uncertainty and Poisson frequency.

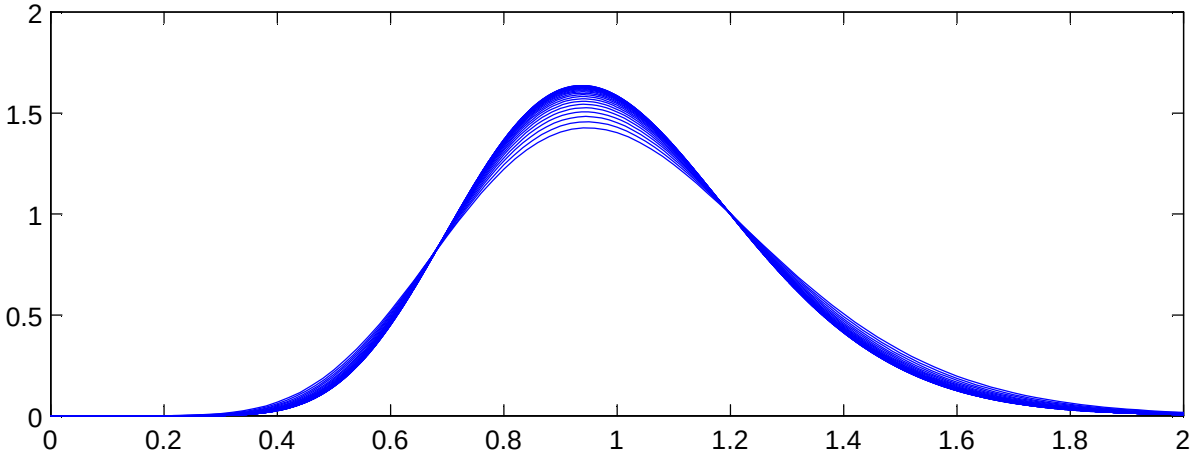

Figure 3.4: Theoretical distribution envelope of scaled aggregate losses with a gamma mixed Poisson frequency with mixing variance $c = 0.0625$.

The proposition implies that if the frequency distribution is actually a Poisson, so the mixing distribution G is $G = 1$ with probability 1, then the loss ratio distribution of a very large book will tend to the distribution concentrated at the expected, hence the expression that "with no parameter risk the process risk completely diversifies away."

Figures 3.3 and 3.4 illustrate the proposition, showing how aggregates change shape as expected counts increase.

In Figure 3.3 $G = 1$ and the claim count is Poisson. Here the scaled distributions get more and more concentrated about the expected value (scaled to 1.0).

In Figure 3.4 G has a gamma distribution with variance 0.0625 (asymptotic CV of 25%). Now the scaled aggregate distributions converge to G.

It is also interesting to compute the correlation between A and G. We have

$$\begin{aligned} \mathrm{Cov}(A,G) &= \mathrm{E}(AG) - \mathrm{E}(A)\mathrm{E}(G) \\ &= \mathrm{EE}(AG|G) - nx \\ &= \mathrm{E}(nxG^2) - nx \\ &= nxc, \end{aligned} \tag{3.58}$$

and therefore

$$\mathrm{Corr}(A,G) = nxc/\sqrt{nx\gamma + n(1+cn)}\sqrt{c} \to 1 \tag{3.59}$$

as $n \to \infty$.

The proposition shows that in some situations severity is irrelevant to large books of business. However, it is easy to think of examples where severity is very important, even for large books of business. For example, severity becomes important in excess of loss reinsurance when it is not clear whether a loss distribution effectively exposes an excess layer. There, the difference in severity curves can amount to the difference between substantial loss exposure and none. The proposition does *not* say that any uncertainty surrounding the severity distribution diversifies away; it is only true when the severity distribution is known with certainty. As is often the case with risk management metrics, great care needs to be taken when applying general statements to particular situations!

Chapter 4

THE IMAN-CONOVER METHOD

Here is the basic idea of the Iman-Conover method. Given samples of n values from two known marginal distributions X and Y and a desired correlation ρ between them, re-order the samples to have the same rank order as a reference distribution, of size $n \times 2$, with linear correlation ρ. Since linear correlation and rank correlation are typically close, the re-ordered output will have approximately the desired correlation structure. What makes the IC method work so effectively is the existence of easy algorithms to determine samples from reference distributions with prescribed linear correlation structures.

Section 4.1 explains the Choleski trick for generating multivariate reference distributions with given correlation structure. Section 4.2 gives a formal algorithmic description of the IC method.

4.1 Theoretical Derivation

Suppose that $\mathbf{M}$ is an n element sample from an r dimensional multivariate distribution, so $\mathbf{M}$ is an $n \times r$ matrix. Assume that the columns of $\mathbf{M}$ are uncorrelated, have mean zero, and standard deviation one. Let $\mathbf{M}'$ denote the transpose of $\mathbf{M}$. These assumptions imply that the correlation matrix of the sample $\mathbf{M}$ can be computed as $n^{-1}\mathbf{M}'\mathbf{M}$, and because the columns are independent, $n^{-1}\mathbf{M}'\mathbf{M} = \mathbf{I}$. (There is no need to scale the covariance matrix by the row and column standard

deviations because they are all one. In general $n^{-1}\mathbf{M}'\mathbf{M}$ is the covariance matrix of **M**.)

Let **S** be a correlation matrix, i.e. **S** is a positive semi-definite symmetric matrix with 1's on the diagonal and all elements ≤ 1 in absolute value. In order to rule out linearly dependent variables assume **S** is positive definite. These assumptions ensure **S** has a Choleski decomposition

$$\mathbf{S} = \mathbf{C}'\mathbf{C} \tag{4.1}$$

for some upper triangular matrix **C**, see Golub [13] or Press et al. [31]. Set $\mathbf{T} = \mathbf{MC}$. The columns of **T** still have mean zero, because they are linear combinations of the columns of **M** which have zero mean by assumption. It is less obvious, but still true, that the columns of **T** still have standard deviation one. To see why, remember that the covariance matrix of **T** is

$$n^{-1}\mathbf{T}'\mathbf{T} = n^{-1}\mathbf{C}'\mathbf{M}'\mathbf{M}\mathbf{C} = \mathbf{C}'\mathbf{C} = \mathbf{S}, \tag{4.2}$$

since $n^{-1}\mathbf{M}'\mathbf{M} = \mathbf{I}$ is the identity by assumption. Now **S** is actually the correlation matrix too because the diagonal is scaled to one, so the covariance and correlation matrices coincide. The process of converting **M**, which is easy to simulate, into **T**, which has the desired correlation structure **S**, is the theoretical basis of the IC method.

It is important to note that estimates of correlation matrices, depending on how they are constructed, need not have the mathematical properties of a correlation matrix. Therefore, when trying to use an estimate of a correlation matrix in an algorithm, such as the Iman-Conover, which actually requires a proper correlation matrix as input, it may be necessary to check the input matrix does have the correct mathematical properties.

Next we discuss how to make $n \times r$ matrices **M**, with independent, mean zero columns. The basic idea is to take n numbers $a_1, \ldots, a_n$ with $\sum_i a_i = 0$ and $n^{-1}\sum_i a_i^2 = 1$, use them to form one $n \times 1$ column of **M**, and then to copy it r times. Finally randomly permute the entries in each column to make them independent as columns of random variables. Iman and Conover call the a_i "scores".

They discuss several possible definitions for the scores, including scaled versions of $a_i = i$ (ranks) and a_i uniformly distributed. They note that the shape of the output multivariate distribution depends on the scores. All of the examples in their paper use normal scores. We will discuss normal scores here, and consider alternatives in Section 4.4.1.

Given that the scores will be based on normal random variables, we can either simulate n random standard normal variables and then shift and re-scale to ensure mean zero and standard deviation one, or we can use a stratified sample from the standard normal, $a_i = \Phi^{-1}(i/(n+1))$. By construction, the stratified sample has mean zero which is an advantage. Also, by symmetry, using the stratified sample halves the number of calls to Φ^{-1}. For these two reasons we prefer it in the algorithm below.

The correlation matrix of **M**, constructed by randomly permuting the scores in each column, will only be approximately equal to **I** because of random simulation error. In order to correct for the slight error which could be introduced Iman and Conover use another adjustment in their algorithm. Let $\mathbf{E} = n^{-1}\mathbf{M}'\mathbf{M}$ be the actual correlation matrix of **M** and let $\mathbf{E} = \mathbf{F}'\mathbf{F}$ be the Choleski decomposition of **E**, and define $\mathbf{T} = \mathbf{M}\mathbf{F}^{-1}\mathbf{C}$. The columns of **T** have mean zero, and the covariance matrix of **T** is

$$\begin{aligned} n^{-1}\mathbf{T}'\mathbf{T} &= n^{-1}\mathbf{C}'\mathbf{F}'^{-1}\mathbf{M}'\mathbf{M}\mathbf{F}^{-1}\mathbf{C} \\ &= \mathbf{C}'\mathbf{F}'^{-1}\mathbf{E}\mathbf{F}^{-1}\mathbf{C} \\ &= \mathbf{C}'\mathbf{F}'^{-1}\mathbf{F}'\mathbf{F}\mathbf{F}^{-1}\mathbf{C} \\ &= \mathbf{C}'\mathbf{C} \\ &= \mathbf{S}, \end{aligned} \tag{4.3}$$

and hence **T** has correlation matrix exactly equal to **S**, as desired. If **E** is singular then the column shuffle needs to be repeated.

Now the reference distribution **T** with exact correlation structure **S** is in hand, all that remains to complete the IC method is to re-order the each column of the input distribution **X** to have the same rank order as the corresponding column of **T**.

4.2 Algorithm

Here is a more algorithmic description of the IC method. The description uses normal scores and the Choleski method to determine the reference distribution. As we discussed above, it is possible to make other choices in place of these and they are discussed in Section 4.4. We will actually present two versions of the core algorithm. The first, called "Simple Algorithm" deals with the various matrix operations at a high level. The second "Detailed Algorithm" takes a more sophisticated approach to the matrix operations, including referencing appropriate Lapack routines [1]. Lapack is a standard set of linear algebra functions. Software vendors provide very high performance implementations of Lapack, many of which are used in CPU benchmarks. Several free Windows implementations are available on the web. The software described in the Appendix uses the Intel Performance http://www.intel.com/software/products/perflib/. The reader should study the simple algorithm first to understand what is going in the IC method. In order to code a high performance implementation you should follow the steps outlined in the detailed algorithm. Both algorithms have the same inputs and outputs.

Inputs: An $n \times r$ matrix $\mathbf{X}$ consisting of n samples from each of r marginal distributions, and a desired correlation matrix $\mathbf{S}$.

The IC method does not address how the columns of $\mathbf{X}$ are determined. It is presumed that the reader has sampled from the appropriate distributions in some intelligent manner. The matrix $\mathbf{S}$ must be a correlation matrix for linearly independent random variables, so it must be symmetric and positive definite. If $\mathbf{S}$ is not symmetric positive semi-definite the algorithm will fail at the Choleski decomposition step. The output is a matrix $\mathbf{T}$ each of whose columns is a permutation of the corresponding column of $\mathbf{X}$ and whose approximate correlation matrix is $\mathbf{S}$.

Simple Algorithm:

1. Make one column of scores $a_i = \Phi^{-1}(i/(n+1))$ for $i = 1, \ldots, n$ and rescale to have standard deviation one.

2. Copy the scores r times to make the score matrix **M**.

3. Randomly permute the entries in each column of **M**.

4. Compute the correlation matrix $\mathbf{E} = n^{-1}\mathbf{M}'\mathbf{M}$ of **M**.

5. Compute the Choleski decomposition $\mathbf{E} = \mathbf{F}'\mathbf{F}$ of **E**.

6. Compute the Choleski decomposition $\mathbf{S} = \mathbf{C}'\mathbf{C}$ of the desired correlation matrix **S**.

7. Compute $\mathbf{T} = \mathbf{M}\mathbf{F}^{-1}\mathbf{C}$. The matrix **T** has exactly the desired correlation structure by Equation (4.3).

8. Let **Y** be the input matrix **X** with each column reordered to have exactly the same rank ordering as the corresponding column of **T**.

Detailed Algorithm:

1. Compute the Choleski decomposition of **S**, $\mathbf{S} = \mathbf{C}'\mathbf{C}$, with **C** upper triangular. If the Choleski algorithm fails then **S** is not a valid correlation matrix. Flag an error and exit. Checking **S** is a correlation matrix in Step 1 avoids performing wasted calculations and allows the routine to exit as quickly as possible. Also check that all the diagonal entries of **S** are 1 so **S** has full rank. Again flag an error and exit if not. The Lapack routine DPOTRF can use be used to compute the Choleski decomposition. In the absence of Lapack, $\mathbf{C} = (c_{ij})$ can be computed recursively using

$$c_{ij} = \frac{s_{ij} - \sum_{k=1}^{j-1} c_{ik}c_{jk}}{\sqrt{1 - \sum_{k=1}^{j-1} c_{jk}^2}} \tag{4.4}$$

for $1 \le i \le j \le n$—since all the diagonal elements of S equal one. The empty sum $\sum_0^0 = 0$ and for $j > i$ the denominator of (4.4) equals c_{ii} and the elements of **C** should be calculated from left to right, top to bottom. See Wang [37, p. 889] or Herzog [19].

2. Let $m = \lfloor n/2 \rfloor$ be the largest integer less than or equal to $n/2$ and $v_i = \Phi^{-1}(i/(2m+1))$ for $i = 1, \ldots, m$.

3. If n is odd set

$$\mathbf{v} = (v_m, v_{m-1}, \ldots, v_1, 0, -v_1, \ldots, -v_m)$$

   and if n is even set

$$\mathbf{v} = (v_m, v_{m-1}, \ldots, v_1, -v_1, \ldots, -v_m).$$

   Here we have chosen to use normal scores. Other distributions could be used in place of the normal, as discussed in Section 4.4.1. Also note that by taking advantage of the symmetry of the normal distribution halves the number of calls to Φ^{-1} which is relatively computationally expensive. If multiple calls will be made to the IC algorithm then store **v** for use in future calls.

4. Form the $n \times r$ score matrix **M** from r copies of the scores vector **v**.

5. Compute $m_{xx} = n^{-1} \sum_i v_i^2$, the variance of **v**. Note that $\sum_i v_i = 0$ by construction.

6. Randomly shuffle columns $2, \ldots, r$ of the score matrix **M**. Knuth [26, pp.139–41] gives the following algorithm for a random shuffle, which we have implemented it in Visual Basic.

```
'' vtemp[0 to n-1] is the array being shuffled.
'' vtemp[j] is the end, you work backwards up the
''    array shuffling each element.
'' Rnd() returns a uniform random variable
'' between zero and one.

dim j as long, vtemp[0 to n-1] as double
dim temp as double, u as double
''
''  populate vtemp
''
```

```
j=n-1
do while j > 0
  u = Rnd()
  k = CLng(j * u)
  temp = vtemp[j]
  vtemp[j] = vtemp[k]
  vtemp[k] = temp
  j=j-1
loop
```

7. Compute the correlation matrix **E** of the shuffled score matrix **M**. Each column of **M** has mean zero, by construction, and variance m_{xx}. The correlation matrix is obtained by dividing each element of $\mathbf{M}'\mathbf{M}$ by m_{xx}. The matrix product can be computed using the Lapack routine DGEMM. If **E** is singular repeat step 6.

8. Determine Choleski decomposition $\mathbf{E} = \mathbf{F}'\mathbf{F}$ of **E** using the Lapack routine DPOTRF. Because **E** is a correlation matrix it must be symmetric and positive definite and so is guaranteed to have a Choleski root.

9. Compute $\mathbf{F}^{-1}\mathbf{C}$ using the Lapack routine DTRTRS to solve the linear equation $\mathbf{FA} = \mathbf{C}$ for **A**. Solving the linear equation avoids a time consuming matrix inversion and multiplication. The routine DTRTRS is optimized for upper triangular input matrices.

10. Compute the correlated scores $\mathbf{T} = \mathbf{M}\mathbf{F}^{-1}\mathbf{C} = \mathbf{MA}$ using DGEMM. The matrix **T** has exactly the desired correlation structure.

11. Compute the ranks of the elements of **T**. Ranks are computed by indexing the columns of **T** as described in Section 8.4 of [31]. Let $r(k)$ denote the index of the kth ranked element of **T**. See Appendix B for VBA code to perform indexing.

12. Let **Y** be the $n \times r$ matrix with ith column equal to the ith column of the input matrix **X** given the same rank order as **T**. The re-ordering is performed using the ranks computed in the previous step. First sort the input columns into ascending order if they are not already sorted and then set $\mathbf{Y}_{i,k} = \mathbf{X}_{i,r(k)}$.

Outputs: The output of the algorithm is a matrix $\mathbf{Y}$ each of whose columns is a permutation of the corresponding column of the input matrix $\mathbf{X}$. The rank correlation matrix of $\mathbf{Y}$ is identical to that of a multivariate distribution with correlation matrix $\mathbf{S}$.

4.3 Simple Example of Iman-Conover

Having explained the IC method, we now give a simple example to explicitly show all the details. The example will work with $n = 20$ samples and $r = 4$ different marginals. The marginals are samples from four lognormal distributions, with parameters $\mu = 12, 11, 10, 10$ and $\sigma = 0.15, 0.25, 0.35, 0.25$. The input matrix is

$$\mathbf{X} = \begin{pmatrix} 123,567 & 44,770 & 15,934 & 13,273 \\ 126,109 & 45,191 & 16,839 & 15,406 \\ 138,713 & 47,453 & 17,233 & 16,706 \\ 139,016 & 47,941 & 17,265 & 16,891 \\ 152,213 & 49,345 & 17,620 & 18,821 \\ 153,224 & 49,420 & 17,859 & 19,569 \\ 153,407 & 50,686 & 20,804 & 20,166 \\ 155,716 & 52,931 & 21,110 & 20,796 \\ 155,780 & 54,010 & 22,728 & 20,968 \\ 161,678 & 57,346 & 24,072 & 21,178 \\ 161,805 & 57,685 & 25,198 & 23,236 \\ 167,447 & 57,698 & 25,393 & 23,375 \\ 170,737 & 58,380 & 30,357 & 24,019 \\ 171,592 & 60,948 & 30,779 & 24,785 \\ 178,881 & 66,972 & 32,634 & 25,000 \\ 181,678 & 68,053 & 33,117 & 26,754 \\ 184,381 & 70,592 & 35,248 & 27,079 \\ 206,940 & 72,243 & 36,656 & 30,136 \\ 217,092 & 86,685 & 38,483 & 30,757 \\ 240,935 & 87,138 & 39,483 & 35,108 \end{pmatrix}. \tag{4.5}$$

Note that the marginals are all sorted in ascending order. The algorithm does not actually require pre-sorting the marginals but it simplifies the last step.

The desired target correlation matrix is

$$\mathbf{S} = \begin{pmatrix} 1.000 & 0.800 & 0.400 & 0.000 \\ 0.800 & 1.000 & 0.300 & -0.200 \\ 0.400 & 0.300 & 1.000 & 0.100 \\ 0.000 & -0.200 & 0.100 & 1.000 \end{pmatrix}. \tag{4.6}$$

The Choleski decomposition of **S** is

$$\mathbf{C} = \begin{pmatrix} 1.000 & 0.800 & 0.400 & 0.000 \\ 0.000 & 0.600 & -0.033 & -0.333 \\ 0.000 & 0.000 & 0.916 & 0.097 \\ 0.000 & 0.000 & 0.000 & 0.938 \end{pmatrix}. \tag{4.7}$$

Now we make the score matrix. The basic scores are $\Phi^{-1}(i/21)$, for $i = 1, \ldots, 20$. We scale these by 0.868674836252965 to get a vector **v** with standard deviation one. Then we combine four **v**'s and shuffle randomly to get

$$\mathbf{M} = \begin{pmatrix} -1.92062 & 1.22896 & -1.00860 & -0.49584 \\ -1.50709 & -1.50709 & -1.50709 & 0.82015 \\ -1.22896 & 1.92062 & 0.82015 & -0.65151 \\ -1.00860 & -0.20723 & 1.00860 & -1.00860 \\ -0.82015 & 0.82015 & 0.34878 & 1.92062 \\ -0.65151 & -1.22896 & -0.65151 & 0.20723 \\ -0.49584 & -0.65151 & 1.22896 & -0.34878 \\ -0.34878 & -0.49584 & -0.49584 & -0.06874 \\ -0.20723 & -1.00860 & 0.20723 & 0.65151 \\ -0.06874 & 0.49584 & 0.06874 & -1.22896 \\ 0.06874 & -0.34878 & -1.22896 & 0.49584 \\ 0.20723 & 0.34878 & 0.65151 & 0.34878 \\ 0.34878 & -0.06874 & -0.20723 & 1.22896 \\ 0.49584 & -1.92062 & -0.82015 & -0.20723 \\ 0.65151 & 0.20723 & 1.92062 & -1.92062 \\ 0.82015 & 1.00860 & 1.50709 & 1.50709 \\ 1.00860 & -0.82015 & -1.92062 & 1.00860 \\ 1.22896 & 1.50709 & 0.49584 & -1.50709 \\ 1.50709 & 0.06874 & -0.06874 & 0.06874 \\ 1.92062 & 0.65151 & -0.34878 & -0.82015 \end{pmatrix}. \tag{4.8}$$

As described in Section 4.1, **M** is approximately independent. In fact **M** has covariance matrix

$$\mathbf{E} = \begin{pmatrix} 1.0000 & 0.0486 & 0.0898 & -0.0960 \\ 0.0486 & 1.0000 & 0.4504 & -0.2408 \\ 0.0898 & 0.4504 & 1.0000 & -0.3192 \\ -0.0960 & -0.2408 & -0.3192 & 1.0000 \end{pmatrix} \tag{4.9}$$

and **E** has Choleski decomposition

$$\mathbf{F} = \begin{pmatrix} 1.0000 & 0.0486 & 0.0898 & -0.0960 \\ 0.0000 & 0.9988 & 0.4466 & -0.2364 \\ 0.0000 & 0.0000 & 0.8902 & -0.2303 \\ 0.0000 & 0.0000 & 0.0000 & 0.9391 \end{pmatrix}. \tag{4.10}$$

Thus $\mathbf{T} = \mathbf{M}\mathbf{F}^{-1}\mathbf{C}$ is given by

$$\mathbf{T} = \begin{pmatrix} -1.92062 & -0.74213 & -2.28105 & -1.33232 \\ -1.50709 & -2.06697 & -1.30678 & 0.54577 \\ -1.22896 & 0.20646 & -0.51141 & -0.94465 \\ -1.00860 & -0.90190 & 0.80546 & -0.65873 \\ -0.82015 & -0.13949 & -0.31782 & 1.76960 \\ -0.65151 & -1.24043 & -0.27999 & 0.23988 \\ -0.49584 & -0.77356 & 1.42145 & 0.23611 \\ -0.34878 & -0.56670 & -0.38117 & -0.14744 \\ -0.20723 & -0.76560 & 0.64214 & 0.97494 \\ -0.06874 & 0.24487 & -0.19673 & -1.33695 \\ 0.06874 & -0.15653 & -1.06954 & 0.14015 \\ 0.20723 & 0.36925 & 0.56694 & 0.51206 \\ 0.34878 & 0.22754 & -0.06362 & 1.19551 \\ 0.49584 & -0.77154 & 0.26828 & 0.03168 \\ 0.65151 & 0.62666 & 2.08987 & -1.21744 \\ 0.82015 & 1.23804 & 1.32493 & 1.85680 \\ 1.00860 & 0.28474 & -1.23688 & 0.59246 \\ 1.22896 & 1.85260 & 0.17411 & -1.62428 \\ 1.50709 & 1.20294 & 0.39517 & 0.13931 \\ 1.92062 & 1.87175 & -0.04335 & -0.97245 \end{pmatrix}. \tag{4.11}$$

An easy calculation will verify that **T** has correlation matrix **S**, as required.

To complete the IC method we must re-order each column of **X** to have the same rank order as **T**. The first column does not change because it is already in ascending order. In the second column, the first element of **Y** must be the 14th element of **X**, the second the 20th, third 10th and so on. The ranks of the other elements are

$$\begin{pmatrix} 14 & 20 & 10 & 18 & 11 & 19 & 17 & 13 & 15 & 8 & 12 & 6 & 9 & 16 & 5 & 3 & 7 & 2 & 4 & 1 \\ 20 & 19 & 16 & 4 & 14 & 13 & 2 & 15 & 5 & 12 & 17 & 6 & 11 & 8 & 1 & 3 & 18 & 9 & 7 & 10 \\ 18 & 6 & 15 & 14 & 2 & 8 & 9 & 13 & 4 & 19 & 10 & 7 & 3 & 12 & 17 & 1 & 5 & 20 & 11 & 16 \end{pmatrix}'$$

and the resulting re-ordering of **X** is

$$\mathbf{T} = \begin{pmatrix} 123,567 & 50,686 & 15,934 & 16,706 \\ 126,109 & 44,770 & 16,839 & 25,000 \\ 138,713 & 57,685 & 17,620 & 19,569 \\ 139,016 & 47,453 & 35,248 & 20,166 \\ 152,213 & 57,346 & 20,804 & 30,757 \\ 153,224 & 45,191 & 21,110 & 24,019 \\ 153,407 & 47,941 & 38,483 & 23,375 \\ 155,716 & 52,931 & 17,859 & 20,796 \\ 155,780 & 49,420 & 33,117 & 27,079 \\ 161,678 & 58,380 & 22,728 & 15,406 \\ 161,805 & 54,010 & 17,265 & 23,236 \\ 167,447 & 66,972 & 32,634 & 24,785 \\ 170,737 & 57,698 & 24,072 & 30,136 \\ 171,592 & 49,345 & 30,357 & 20,968 \\ 178,881 & 68,053 & 39,483 & 16,891 \\ 181,678 & 72,243 & 36,656 & 35,108 \\ 184,381 & 60,948 & 17,233 & 26,754 \\ 206,940 & 86,685 & 25,393 & 13,273 \\ 217,092 & 70,592 & 30,779 & 21,178 \\ 240,935 & 87,138 & 25,198 & 18,821 \end{pmatrix}. \tag{4.12}$$

The rank correlation matrix of **Y** is exactly **S**. The actual linear correlation is only approximately equal to **S**. The achieved value is

$$\begin{pmatrix} 1.00 & 0.85 & 0.26 & -0.11 \\ 0.85 & 1.00 & 0.19 & -0.20 \\ 0.26 & 0.19 & 1.00 & 0.10 \\ -0.11 & -0.20 & 0.10 & 1.00 \end{pmatrix}, \tag{4.13}$$

a fairly creditable performance given the input correlation matrix and the very small number of samples $n = 20$. When used with larger sized samples the IC method typically produces a very close approximation to the required correlation matrix, especially when the marginal distributions are reasonably symmetric.

4.4 Extensions of Iman-Conover

Following through the explanation of the IC method shows that it relies on a choice of multivariate reference distribution. A straightforward method to compute a reference is to use the Choleski decomposition method Equation (4.2) applied to certain independent scores. The example in Section 4.3 used normal

scores. However nothing prevents us from using other distributions for the scores provided they are suitably normalized to have mean zero and standard deviation one. We explore the impact of different choices of score distribution on the resulting multivariate distribution in Section 4.4.1.

Another approach to IC is to use a completely different multivariate distribution as reference. There are several other families of multivariate distributions, including the elliptically contoured distribution family (which includes the normal and t as a special cases) and multivariate Laplace distribution, which are easy to simulate from. We explore the impact of changing the reference distribution in Section 4.4.2. Note that changing scores is actually an example of changing the reference distribution; however, for the examples we consider the exact form of the new reference is unknown.

4.4.1 Alternative Scores

The choice of score distribution has a profound effect on the multivariate distribution output by the IC method. Recall that the algorithm described in Section 4.2 used normally distributed scores. We now show the impact of using exponentially and uniformly distributed scores.

Figure 4.1 shows three bivariate distributions with identical marginal distributions (shown in the lower right hand plot), the same correlation coefficient of 0.643 ± 0.003 but using normal scores (top left), exponential scores (top rigtht) and uniform scores (lower left). The input correlation to the IC method was 0.65 in all three cases and there are 1000 pairs in each plot. Here the IC method produced bivariate distributions with actual correlation coefficient extremely close to the requested value.

The normal scores produce the most natural looking bivariate distribution, with approximately elliptical contours. The bivariate distributions with uniform or exponential scores look unnatural, but it is important to remember that if all you know about the bivariate distribution are the marginals and correlation coefficient all three outcomes are possible.

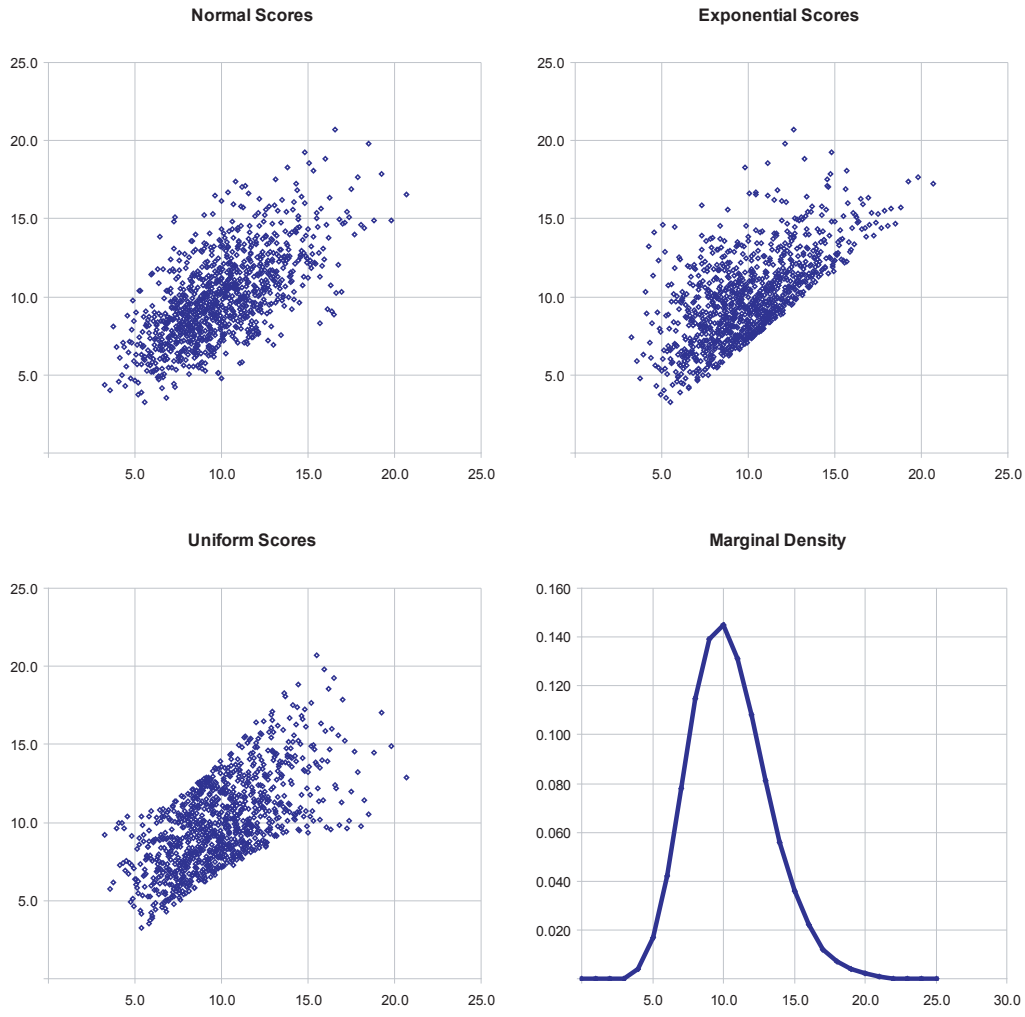

Figure 4.1: Bivariate distributions with normal, uniform and exponential scores.

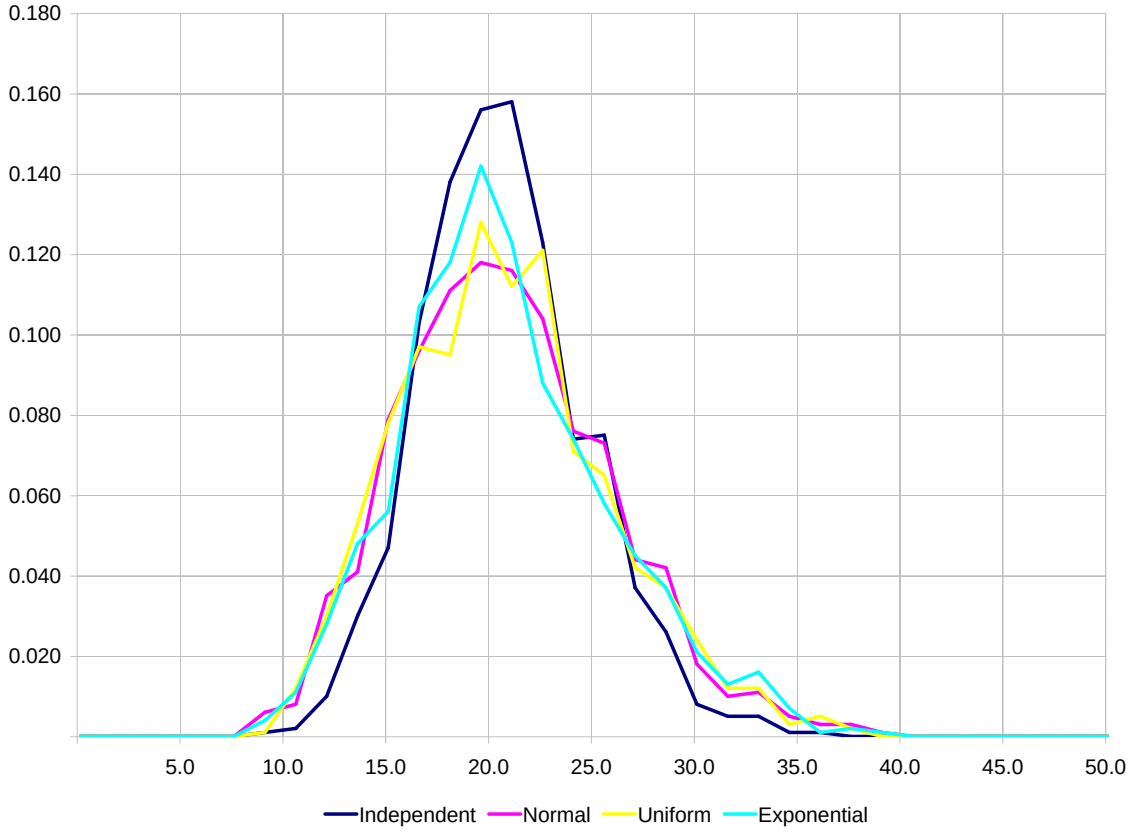

Figure 4.2: Sum of marginals from bivariate distributions made with different score distributions.

Figure 4.2 shows the distribution of the sum of the two marginals for each of the three bivariate distributions in Figure 4.1 and for independent marginals. The sum with exponential scores has a higher kurtosis (is more peaked) than with normal scores. As expected all three dependent sums have visibly thicker tails than the independent sum.

Iman and Conover considered various different score distributions in their paper. They preferred normal scores as giving more natural looking, elliptical contours. Certainly, the contours produced using exponential or uniform scores appear unnatural. If nothing else they provide a sobering reminder that knowing the marginal distributions and correlation coefficient of a bivariate distribution does not come close to fully specifying it!

4.4.2 Multivariate Reference Distributions

The IC method needs some reference multivariate distribution to determine an appropriate rank ordering for the input marginals. So far we have discussed using the Choleski decomposition trick in order to determine a multivariate normal reference distribution. However, any distribution can be used as reference provided it has the desired correlation structure. Multivariate distributions that are closely related by formula to the multivariate normal, such as elliptically contoured distributions and asymmetric Laplace distributions, can be simulated using the Choleski trick.

Elliptically contoured distributions are a family which extends the normal. For a more detailed discussion see Fang and Zhang [11]. The multivariate t-distribution and symmetric Laplace distributions are in the elliptically contoured family. Elliptically contoured distributions must have characteristic equations of the form

$$\Phi(\mathbf{t}) = \exp(i\mathbf{t}'\mathbf{m})\phi(\mathbf{t}'\mathbf{S}\mathbf{t}) \tag{4.14}$$

for some $\phi : \mathbf{R} \to \mathbf{R}$, where $\mathbf{m}$ is an $r \times 1$ vector of means and $\mathbf{S}$ is a $r \times r$ covariance matrix (nonnegative definite and symmetric). In one dimension the

elliptically contoured distributions coincide with the symmetric distributions. The covariance is **S**, if it is defined.

If **S** has rank r then an elliptically contoured distribution **x** has a stochastic representation

$$\mathbf{x} = \mathbf{m} + R\mathbf{T}'\mathbf{u}^{(r)} \tag{4.15}$$

where **T** is the Choleski decomposition of **S**, so $\mathbf{S} = \mathbf{T}'\mathbf{T}$, $\mathbf{u}^{(r)}$ is a uniform distribution on the sphere in $\mathbf{R}^r$, and R is a scale factor independent of $\mathbf{u}^{(r)}$. The idea here should be clear: pick a direction on the sphere, adjust by **T**, scale by a distance R and finally translate by the means **m**. A uniform distribution on a sphere can be created as $\mathbf{x}/\|\mathbf{x}\|$ where **x** has a multivariate normal distribution with identity covariance matrix. (By definition, $\|\mathbf{x}\|^2 = \sum_i x_i^2$ has a χ_r^2 distribution.) Uniform vectors $\mathbf{u}^{(r)}$ can also be created by applying a random orthogonal matrix to a fixed vector $(1, 0, \ldots, 0)$ on the sphere. Diaconis [8] describes a method for producing random orthogonal matrices.

The t-copula with ν degrees of freedom has a stochastic representation

$$\mathbf{x} = \mathbf{m} + \frac{\sqrt{\nu}}{\sqrt{S}}\mathbf{z} \tag{4.16}$$

where $S \sim \chi_\nu^2$ and **z** is multivariate normal with means zero and covariance matrix **S**. Thus one can easily simulate from the multivariate t by first simulating multivariate normals and then simulating an independent S and multiplying.

The multivariate Laplace distribution is discussed in Kotz, Kozubowski and Podgorski [27]. It comes in two flavors: symmetric and asymmetric. The symmetric distribution is also an elliptically contoured distribution. It has characteristic function of the form

$$\Phi(\mathbf{t}) = \frac{1}{1 + \mathbf{t}'\mathbf{S}\mathbf{t}/2} \tag{4.17}$$

where **S** is the covariance matrix. To simulate from (4.17) use the fact that $\sqrt{W}\mathbf{X}$ has a symmetric Laplace distribution if W is exponential and **X** a multivariate normal with covariance matrix **S**.

The multivariate asymmetric Laplace distribution has characteristic function

$$\Psi(\mathbf{t}) = \frac{1}{1 + \mathbf{t}'\mathbf{S}\mathbf{t}/2 - i\mathbf{m}'\mathbf{t}}. \tag{4.18}$$

To simulate from (4.18) use the fact that

$$\mathbf{m}W + \sqrt{W}\mathbf{X} \tag{4.19}$$

has a symmetric Laplace distribution if W is exponential and $\mathbf{X}$ a multivariate normal with covariance matrix $\mathbf{S}$ and means zero. The asymmetric Laplace is not an elliptically contoured distribution.

Figure 4.3 compares IC samples produced using a normal copula to those produced with a t-copula. In both cases the marginals are normally distributed with mean zero and unit standard deviation. The t-copula has $\nu = 2$ degrees of freedom. In both figures the marginals are uncorrelated, but in the right the marginals are not independent. The t-copula has pinched tails, similar to Venter's Heavy Right Tailed copula [33]

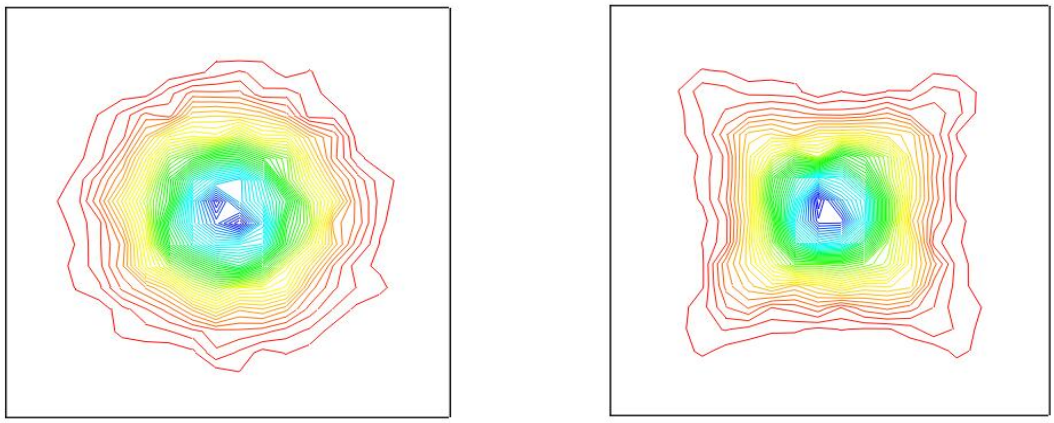

Figure 4.3: IC samples produced from the same marginal and correlation matrix using the normal and t copula reference distributions.

4.4.3 Algorithms for Extended Methods

In Section 4.4.2 we described how the IC method can be extended by using different reference multivariate distributions. It is easy to change the IC algorithm to incorporate different reference distributions for t-copulas and asymmetric Laplace distributions. Follow the detailed algorithm to step 10. Then use the stochastic representation (4.16) (resp. 4.19 for the Laplace): simulate from the scaling distribution for each row and multiply each component by the resulting number, resulting in an adjusted **T** matrix. Then complete steps 11 and 12 of the detailed algorithm.

4.5 Comparison With the Normal Copula Method

By the normal copula method we mean the following algorithm, described in Wang [37] and Herzog [19].

Inputs: A set of correlated risks $(X_1, \ldots, X_r)$ with marginal cumulative distribution functions F_i and Kendall's tau $\tau_{ij} = \tau(X_i, X_j)$ or rank correlation coefficients $r(X_i, X_j)$.

Algorithm:

1. Convert Kendall's tau or rank correlation coefficient to correlation using

$$\rho_{ij} = \sin(\pi\tau_{ij}/2) = 2\sin(\pi r_{ij}/6) \tag{4.20}$$

   and construct the Choleski decomposition $\mathbf{S} = \mathbf{C}'\mathbf{C}$ of $\mathbf{S} = (\rho_{ij})$.

2. Generate r standard normal variables $\mathbf{Y} = (Y_1, \ldots, Y_r)$.

3. Set $\mathbf{Z} = \mathbf{YC}$.

4. Set $u_i = \Phi(Z_i)$ for $i = 1, \ldots, r$.

5. Set $X_i = F_i^{-1}(u_i)$.

Outputs: The vectors $(X_1, \ldots, X_r)$ form a sample from a multivariate distribution with prescribed correlation structure and marginals F_i.

The Normal Copula method works because of the following theorem from Wang [37, Theorem 2].

Theorem 2 *Assume that $(Z_1, \ldots, Z_k)$ have a multivariate normal joint probability density function given by*

$$f(z_1, \ldots, z_k) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp(-\mathbf{z}'\Sigma^{-1}\mathbf{z}/2), \tag{4.21}$$

$\mathbf{z} = (z_1, \ldots, z_k)$, with correlation coefficients $\Sigma_{ij} = \rho_{ij} = \rho(Z_i, Z_j)$. Let $H(z_1, \ldots, z_k)$ be their joint cumulative distribution function. Then

$$C(u_1, \ldots, u_k) = H(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_k)) \tag{4.22}$$

defines a multivariate uniform cumulative distribution function called the normal copula.

For any set of given marginal cumulative distribution functions $F_1, \ldots, F_k$, the set of variables

$$X_1 = F_1^{-1}(\Phi(Z_1)), \ldots, X_k = F_1^{-1}(\Phi(Z_k)) \tag{4.23}$$

have a joint cumulative function

$$F_{X_1,\ldots,X_k}(x_1, \ldots, x_k) = H(\Phi^{-1}(F_x(u_1)), \ldots, \Phi^{-1}(F_k(u_k)) \tag{4.24}$$

with marginal cumulative distribution functions $F_1, \ldots, F_k$. The multivariate variables $(X_1, \ldots, X_k)$ have Kendall's tau

$$\tau(X_i, X_j) = \tau(Z_i, Z_j) = \frac{2}{\pi} \arcsin(\rho_{ij}) \tag{4.25}$$

and Spearman's rank correlation coefficients

$$\mathit{rkCorr}(X_i, X_j) = \mathit{rkCorr}(Z_i, Z_j) = \frac{6}{\pi} \arcsin(\rho_{ij}/2) \tag{4.26}$$

In the normal copula method we simulate from H and then invert using (4.23). In the IC method with normal scores we produce a sample from H such that $\Phi(z_i)$ are equally spaced between zero and one and then, rather than invert the distribution functions, we make the jth order statistic from the input sample correspond to $\Phi(z) = j/(n+1)$ where the input has n observations. Because the jth order statistic of a sample of n observations from a distribution F approximates $F^{-1}(j/(n+1))$ we see the normal copula and IC methods are doing essentially the same thing.

While the normal copula method and the IC method are confusingly similar there are some important differences to bear in mind. Comparing and contrasting the two methods should help clarify how the two algorithms are different.

1. Theorem 2 shows the normal copula method corresponds to the IC method when the latter is computed using normal scores and the Choleski trick.

2. The IC method works on a given sample of marginal distributions. The normal copula method generates the sample by inverting the distribution function of each marginal as part of the simulation process.

3. Though the use of scores the IC method relies on a stratified sample of normal variables. The normal copula method could use a similar method, or it could sample randomly from the base normals. Conversely a sample could be used in the IC method.

4. Only the IC method has an adjustment to ensure that the reference multivariate distribution has exactly the required correlation structure.

5. IC method samples have rank correlation exactly equal to a sample from a reference distribution with the correct linear correlation. Normal copula samples have approximately correct linear and rank correlations.

6. An IC method sample must be taken in its entirety to be used correctly. The number of output points is fixed by the number of input points, and the sample is computed in its entirety in one step. Some IC tools (@Risk, SCARE) produce output which is in a particular order. Thus, if you sample the nth observation from multiple simulations, or take the first n samples, you will not get a random sample from the desired distribution. However, if you select random rows from multiple simulations (or, equivalently, if you randomly permute the rows output prior to selecting the nth) then you will obtain the desired random sample. It is important to be aware of these issues before using canned software routines.

7. The normal copula method produces simulations one at a time, and at each iteration the resulting sample is a sample from the required multivariate distribution. That is, output from the algorithm can be partitioned and used in pieces.

In summary remember these differences can have material practical consequences and it is important not to misuse IC method samples.

Chapter 5

EXAMPLES

We now give an extended example which applies the IC method and some of the other methods introduced in Chapter 3. The example will compute the bivariate distribution of retained and ceded losses, where the reinsurance is an excess of loss cover. Such a bivariate distribution would be useful for a ceding company trying to determine its distribution of net underwriting results if the reinsurance included a variable feature such as a swing rate, sliding scale commission, annual aggregate deductible or profit commission.

The example will apply the following methods and techniques:

M1. Computing aggregate distributions using FFTs.

M2. Compare aggregate distributions computed using FFTs (essentially exact) with method of moments shifted lognormal and shifted gamma approximations.

M3. Computing the bivariate distribution of ceded and retained losses using two dimensional FFTs.

M4. Computing the bivariate distribution of ceded and retained losses using the IC method.

M5. Compare the FFT and IC method.

M6. Illustrate the effect on the IC method of using a t reference distribution.

The first two examples compute univariate marginal distributions, the fundamental inputs to the IC method. The next five examples compute multivariate distributions in various ways.

The reader should take away two lessons from these examples. First, the FFT method works incredibly well for small claim counts and thin tailed severity distributions. In particular, any severity distribution with an occurrence limit is thin tailed. Second, the shifted gamma and shifted lognormal approximations to an aggregate distribution are exceedingly good in all but the most extreme cases. Extreme cases include a very small claim count (say less than five) or a thick tailed severity distribution.

5.1 Example Parameters

The input parameters for the example are as follows. Severity is modeled using a lognormal variable with $\mu = 9.0$ and $\sigma = 2.0$. Underlying policies have a \$1M policy limit. The excess of loss reinsurance attaches at \$200,000 and has a limit of \$800,000; thus it pays all losses between \$200,000 and the policy limit, ground-up. The ground-up expected loss is \$25M.

The nth moments of the layer y excess a of for losses with density f, viz.

$$\mathrm{E}[\min(y, \max(X-a, 0))^n] = \int_a^{a+y} (x-a)^n f(x)dx + y^n \mathrm{Pr}(X > a+y), \quad (5.1)$$

can be computed using

$$\int_a^{a+y} (x-a)^n f(x)dx = \sum_{k=0}^{n} \binom{n}{k} (-a)^k \int_a^{a+y} x^{n-k} f(x)dx \quad (5.2)$$

reducing the problem to computing simpler integrals. For the lognormal, the integral $\lambda(n) := \int_a^{a+y} x^n f(x)dx$ equals

$$\lambda(n) = \exp(n\mu + n^2\sigma^2/2)\Phi((\log(a+y) - \mu - n\sigma^2)/\sigma) \quad (5.3)$$

if $a = 0$ and

$$\lambda(n) = \exp(n\mu+n^2\sigma^2/2)(\Phi((\log(a+y)-\mu-n\sigma^2)/\sigma)-\Phi((\log(a)-\mu-n\sigma^2)/\sigma)) \tag{5.4}$$

for $a > 0$. Then by the above formula for the lognormal we have

$$\mathrm{E}[\min(y, \max(X-a, 0))^n] = y^n(1-\Phi((\log(a+y)-\mu)/\sigma))+\sum_{k=0}^{k=n} \binom{n}{k} (-a)^k \lambda(n-k). \tag{5.5}$$

Using these formulae we find that the expected ground up loss is $47,439, the expected retained loss is $31,591 and the expected ceded loss is $15,848. The probability of attaching the reinsurance is 0.054463 and so the excess severity, conditional on an excess claim, is $290,985.

The expected claim count is $n = 526.00 = 25000000/47439$. We will model claims using a negative binomial with contagion parameter $c = 0.0625$ which by the discussion in Section 3.3 corresponds to an asymptotic CV of 25% for the aggregate distribution. The parameters of the negative binomial are $r = 1/c = 16$ and $p = 1/(1 + cn) = 0.0295$, using the Bowers et al. [4] parameterization. The moments of the negative binomial are

$$\begin{aligned} E(N) &= (1-p)r/p \\ E(N^2) &= (p-1)r((p-1)r-1)/p^2 \\ E(N^3) &= (1-p)r((p-1)r((p-1)r-3)-p+2)/p^3 \end{aligned}$$

(computed using symbolic differentiation of the moment generating function using a computer algebra program).

Now we can compute the moments of the gross aggregate distribution using (3.14) and (3.16). Writing $A = X_1 + \cdots + X_N$ the results are

$$\begin{aligned} \mathrm{E}(A) &= \mathrm{E}(N)\mathrm{E}(X) \\ \mathrm{E}(A^2) &= \mathrm{E}(N)\mathrm{E}(X^2) + \mathrm{E}(X)^2\mathrm{E}(N^2) - \mathrm{E}(N)\mathrm{E}(X)^2 \\ \mathrm{E}(A^3) &= \mathrm{E}(N)\mathrm{E}(X^3) + \mathrm{E}(X)^3\mathrm{E}(N^3) + 3\mathrm{E}(X)\mathrm{E}(N^2)\mathrm{E}(X^2) - \\ & \quad 3\mathrm{E}(N)\mathrm{E}(X)\mathrm{E}(X^2) - 3\mathrm{E}(X)^3\mathrm{E}(N^2) + 2\mathrm{E}(N)\mathrm{E}(X)^3. \end{aligned}$$

From these expressions we can compute the variance, standard deviation, CV and skewness of frequency, severity and aggregate losses using the definitions at the beginning of Chapter 3. The results are shown in Table 5.1.

Table 5.1: Frequency, Severity and Aggregate Distribution Statistics

Severity	**Ground-Up**	**Retained**	**Ceded**
$\mathrm{E}(X)$	47,439.0	31,591.0	290,985.3
$\mathrm{CV}(X)$	2.7217	1.6745	0.9513
$\mathrm{skew}(X)$	5.2380	2.2351	0.8375
Frequency			
$\mathrm{E}(X)$	527.0	527.0	28.7
$\mathrm{CV}(X)$	0.2538	0.2538	0.3120
$\mathrm{skew}(X)$	0.5001	0.5001	0.5123
Aggregate			
$\mathrm{E}(X)$	25,000,000.0	16,648,209.8	8,351,790.2
$\mathrm{CV}(X)$	0.2801	0.2640	0.3590
$\mathrm{skew}(X)$	0.5128	0.5018	0.5543

5.2 Univariate Methods—Computing Marginal Distributions

5.2.1 Fast Fourier Transform Methods

To compute the aggregate distribution using Fast Fourier Transforms (FFT) we first have to "bucket" the severity distributions. We will use 4,096 buckets (the number must be a power of two for the FFT to work at peak efficiency) and a bucket size $b = 12,500$. The largest loss that we can produce is therefore \$51.1875M which will be adequate for our example. The easiest way to bucket the severity is to compute the cumulative distribution function F at $b/2, 3b/2, \ldots$ and then take differences. The coefficients of bucketed distribution must sum to one. The distribution for ceded losses is actually the conditional distribution given an excess loss, $F(x)/(1 - F(a))$ where a is the attachment and F is the ground up severity distribution. The first few terms in the bucketed severities are shown in Table 5.2

There are slight errors introduced when you bucket the severity distribution, particularly for the retained losses. The mean of the retained severity is 1.9% lower than the actual; the CV is 2.8% higher and the skewness is 1.5% lower. The

Table 5.2: Bucketed Severity Distributions

Bucket Start	Bucket Mid-Point	Ground-Up	Retained	Ceded
0	6,250	0.448350	0.448350	0.030800
12,500	18,750	0.214215	0.214215	0.056797
25,000	31,250	0.087561	0.087561	0.051170
37,500	43,750	0.050294	0.050294	0.046328
50,000	56,250	0.033252	0.033252	0.042130
62,500	68,750	0.023819	0.023819	0.038467
75,000	81,250	0.017980	0.017980	0.035252
87,500	93,750	0.014089	0.014089	0.032415
100,000	106,250	0.011353	0.011353	0.029899
112,500	118,750	0.009351	0.009351	0.027658
125,000	131,250	0.007839	0.007839	0.025653
137,500	143,750	0.006667	0.006667	0.023853
150,000	156,250	0.005740	0.005740	0.022230
162,500	168,750	0.004993	0.004993	0.020762
175,000	181,250	0.004382	0.004382	0.019431
187,500	193,750	0.003876	0.003876	0.018219
200,000	206,250	0.003452	0.056238	0.017114
212,500	218,750	0.003093	0.000000	0.016103
225,000	231,250	0.002787	0.000000	0.015176
237,500	243,750	0.002523	0.000000	0.014323
250,000	256,250	0.002295	0.000000	0.013538
262,500	268,750	0.002095	0.000000	0.012814
275,000	281,250	0.001920	0.000000	0.012143
287,500	293,750	0.001765	0.000000	0.011522
300,000	306,250	0.001628	0.000000	0.010946
etc.				

Table 5.3: Shifted Gamma Approximations to the Aggregate Distributions

Parameter	Ground-Up	Retained	Ceded
s	-2.341E+06	-8.907E+05	-2.468E+06
α	15.19659	15.88163	13.02156
β	1.7780E+06	1.0842E+06	8.3084E+05

excess severity is virtually exact—because the bucket size is small relative to the features of the distribution. The ground-up severity is in between. The smaller the bucket size the lower these discretization errors will be, but on the other hand the less "space" available for the aggregate distribution. Selecting a bucket size which is an exact divisor of the limit will greatly help improve the accuracy of the discretized severity distribution. To determine if your bucket size is appropriate look at the moments of the FFT aggregate relative to the exact moments and plot a graph of the output density. It is usually obvious when the method has failed.

Next we have to take the Fast Fourier Transform of the three 4096×1 severity vectors. We will assume the reader has a computer routine available which will compute FFTs—see Appendix A for one freely available implementation. Then you apply the moment generating function of the frequency distribution (see Table 3.2) row-by-row to the transformed severity. Note that you will be working with complex numbers. Finally you apply the inverse FFT to get a vector of real numbers. Because of the form of the input you are guaranteed that the output will be real and will sum to 1.

5.2.2 Method of Moments and the Shifted Gamma and Lognormal Distributions

In Section 3.5 we introduced the shifted gamma and lognormal distributions and gave explicit expressions for their method-of-moments parameters in terms of mean, CV and skewness. In our example the gross, retained and ceded fits are shown in Table 5.3 for the shifted gamma, 5.4 for the shifted lognormal, and 5.5 for the lognormal.

Table 5.4: Shifted Lognormal Approximations to the Aggregate Distributions

Parameter	Ground-Up	Retained	Ceded
s	-1.636E+07	-9.872E+06	-8.057E+06
μ	17.52370	17.07988	16.59690
σ	0.16811	0.16463	0.18122

Table 5.5: Lognormal Fits to the Aggregate Distributions

Parameter	Ground-Up	Retained	Ceded
μ	16.98348	16.57454	15.87726
σ	0.27554	0.26015	0.34819

Figure 5.1 shows a comparison of the shifted gamma fits (denoted with an asterisk in the legend) with the FFTs. For each of the total, ground-up loss, retained loss and ceded or excess loss the fits appear essentially perfect. On a log-scale, Figure 5.2, we see that the fits are again essentially perfect except for disagreement for small losses. However, the disagreement actually shows an error in the FFT; probabilities for losses greater than the largest bucket size (approximately $50M) wrap around in the FFT and re-appear as small losses, thus the FFT picture is actually inaccurate. The wrapping phenomenon is an example of aliasing; it is the same effect that causes wagon wheels to appear to rotate backwards in Western movies. See Hamming [16] for more details. The shifted gamma approximation is recommended in Bowers et al. [4].

Figure 5.3 shows the shifted lognormal fit. Although not quite as good as the shifted gamma, the fit is still very close. A log scale (not shown) would show that the shifted lognormal is somewhat thicker in the extreme tail. The fact that the shifted gamma does a better job in the tail should not be a surprise since the negative binomial uses a gamma mixing distribution.

Finally, Figure 5.4 shows a comparison of the FFTs with a regular two parameter lognormal. The lognormal is too skewed (peaks too soon) and does not match the true shape of the aggregate well. Using a shifted gamma or shifted lognormal distribution gives a much more satisfactory fit to the true aggregate for very little extra work.

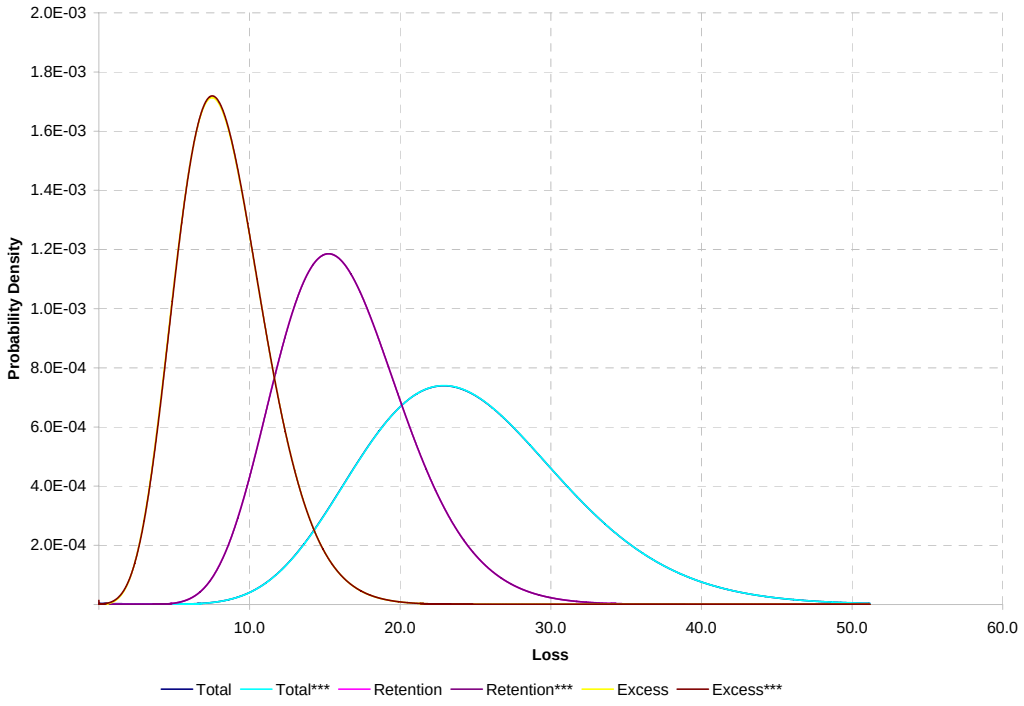

Figure 5.1: FFT vs. shifted gamma approximations for total, retained and ceded losses, illustrating that the gamma is an almost perfect fit.

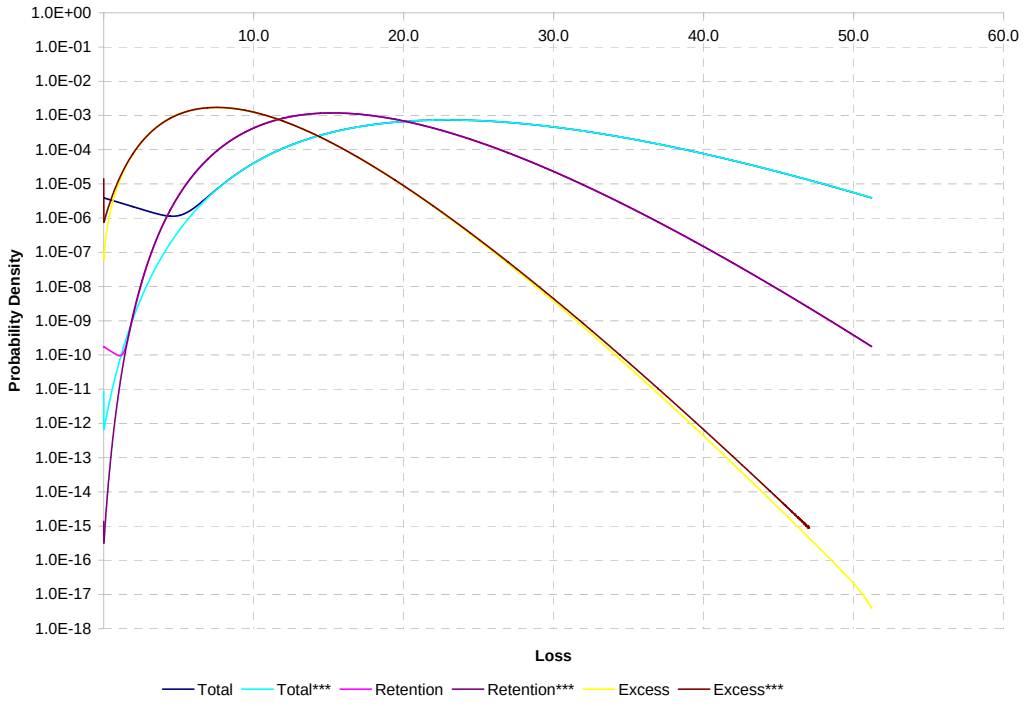

Figure 5.2: FFT vs. shifted gamma approximations for total, retained and ceded losses on a log density scale.

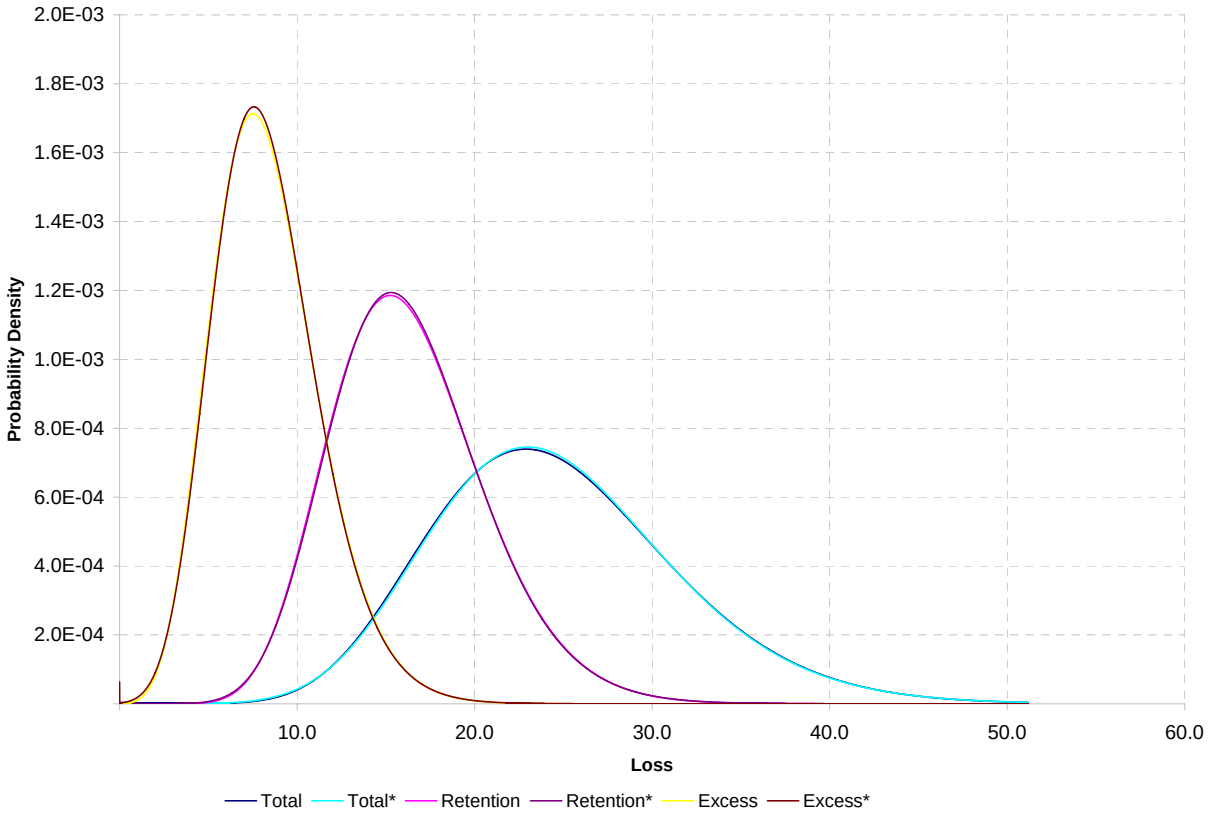

Figure 5.3: FFT vs. shifted lognormal approximations for total, retained and ceded losses.

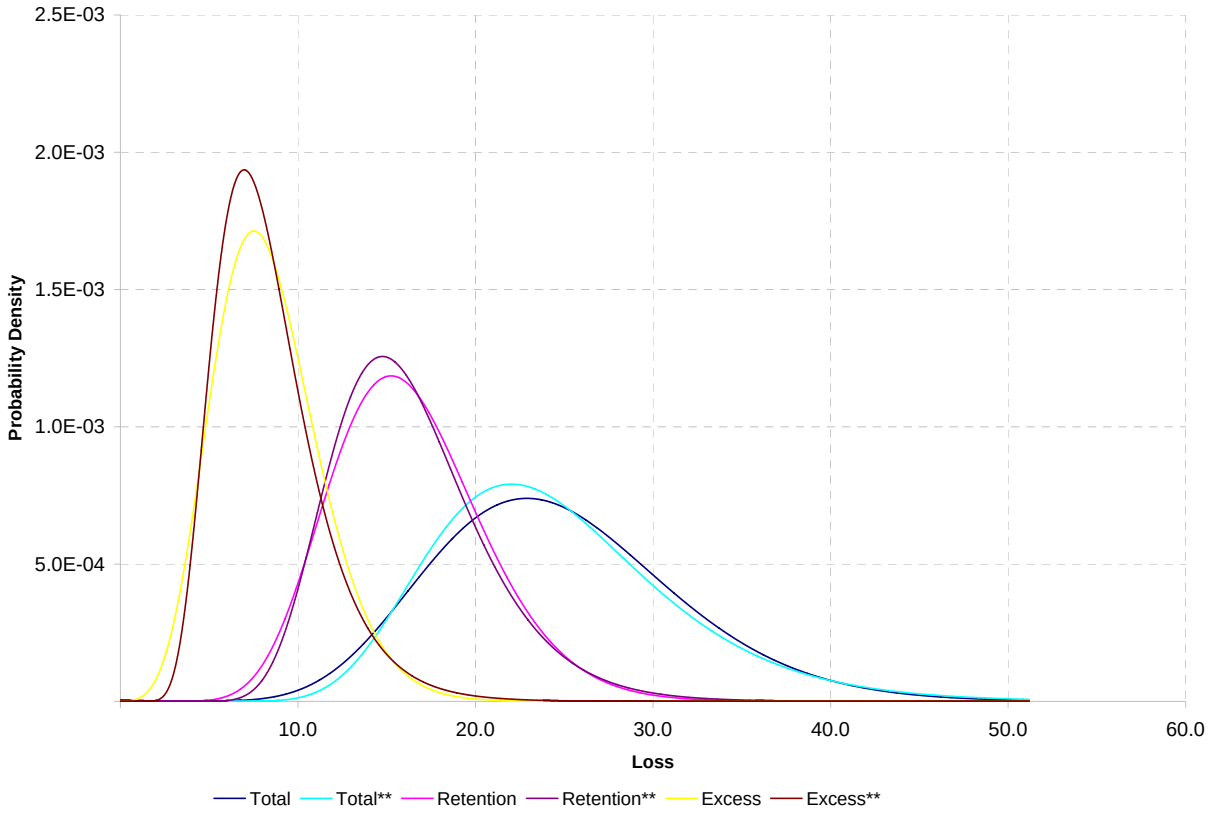

Figure 5.4: FFT vs. lognormal approximations for total, retained and ceded losses, illustrating that the lognormal is a poorer fit than the shifted lognormal.

5.3 Multivariate Methods and the IC Method

Now we have the marginal distributions we need we can apply the IC method to determine the bivariate distribution of retained and ceded losses.

5.3.1 Fast Fourier Transform Methods

In order to have a benchmark for the IC method we begin by computing the exact bivariate distribution of ceded and retained losses using two dimensional FFTs. The two dimensional FFT method is far more limited than the one dimensional version because it is impractical to use discretized distributions larger than 4096×4096—the size we will use here. One is caught by the need for a small bucket size to capture the shape of the ground-up severity and the need for enough buckets to capture the whole aggregate distribution.

The method for two dimensional FFTs is essentially the same as for one dimension: compute a discretized version of the input severity distribution, which will be a matrix rather than a column vector, apply the FFT, apply the MGF of the frequency distribution term-by-term, and then apply the inverse FFT. The resulting distribution is shown in Figure 5.5.

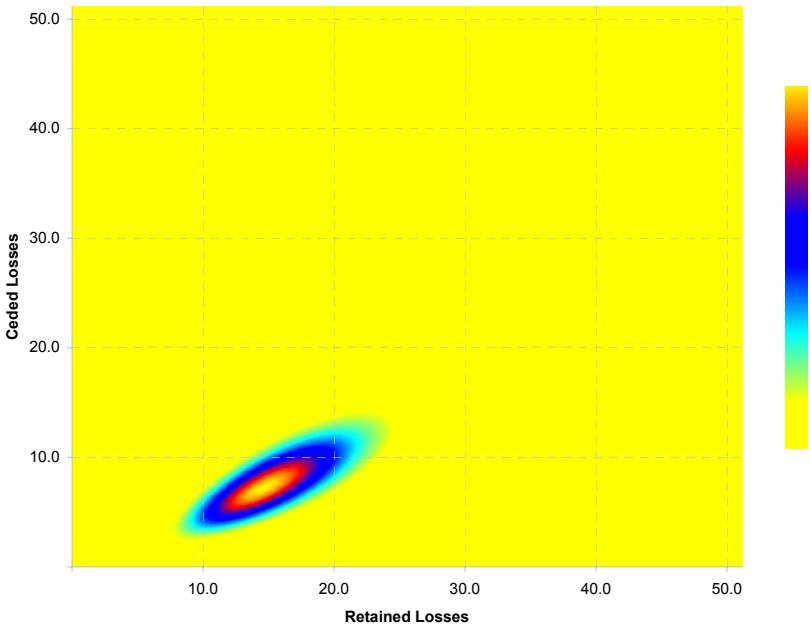

Figure 5.5: Two dimensional FFT estimate of bivariate distribution of ceded and retained losses.

5.3.2 IC Method

Next we apply the IC method to the marginal retained and ceded distribution computed in the previous section. Individual ceded and retained losses are a good example of comonotonic variables, since they are both increasing functions of gross losses. Aggregate ceded and retained losses will not generally be comonotonic. To apply IC we need the correlation coefficient between ceded and retained losses which can be computed using (3.47). The only missing value from that equation is the covariance between retained severity R and ceded severity C. However, because of the simple form of the bivariate severity, viz. ceded losses are zero until gross losses hit the retention $a = \$200,000$ and then ceded losses increase, the covariance is easy to compute:

$$\mathrm{Cov}(R,C) = \mathrm{E}(RC) - \mathrm{E}(R)\mathrm{E}(C) = \mathrm{E}(C)(a - \mathrm{E}(R)). \tag{5.6}$$

Substituting into (3.47) gives a correlation of 0.786 between aggregate retained losses and aggregate ceded losses. We can now apply the Iman Conover method. Here we used samples of 10,000 observations from the univariate distributions of ceded and retained losses. The result of the IC method will be a 10000×2 matrix sample from the bivariate distribution. In order to visualize the result we produced a bivariate histogram, as shown in Figure 5.6. The approximation is very similar to the previous "exact" FFT contour plot, as you can see if you overlay the two plots.

The IC method underlying Figure 5.6 used a normal copula reference distribution. As we have already discussed there are many other possible reference distributions we could chose to use. Figure 5.7 shows the resulting two dimensional histogram if we use a t-copula with two degrees of freedom, which is a very extreme choice. Just as we saw in Figure 4.3 the result of using a t-copula is to introduce more extreme value dependence and the contours have a pinched look—both in the slope 1 and slope -1 directions.

Clearly the normal copula IC method produces bivariate distribution closer to the FFT actual than the t-copula, which should not be a surprise. There is no

generator of extreme tail correlation in our example. However, in other modeling situations, such as modeling the bivariate movement of stock prices or foreign exchange movements, there may be empirical evidence of strong tail correlation and a t-copula (or other non-normal) copula approach would be more appropriate.

Finally, Figure 5.8 shows the distribution of the sum of ceded and retained losses using the normal-copula, t-copula, and actual dependence relationships. As expected the normal copula model is closest to the actual. The t-copula sum is too peaked and is more thick tailed than the actual distribution.

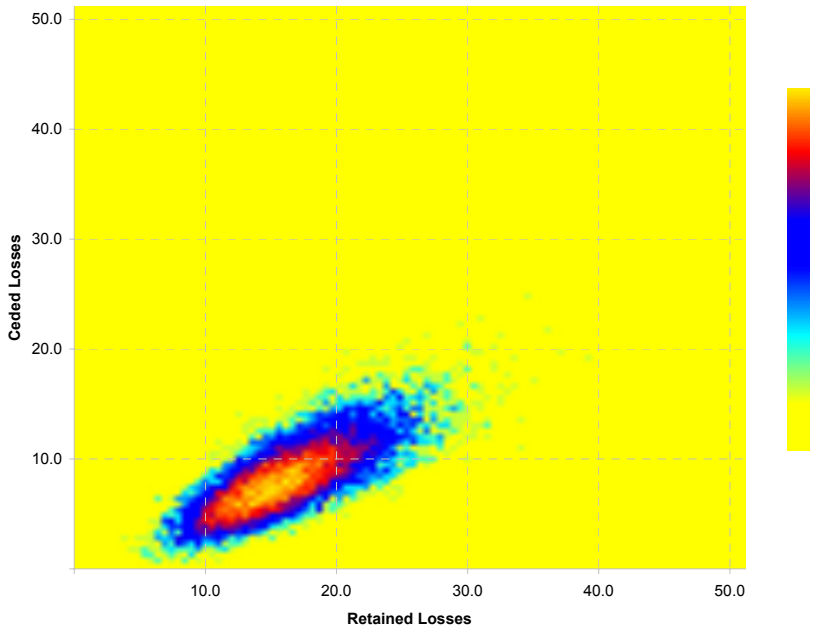

Figure 5.6: Iman-Conover approximation to bivariate distribution of ceded and retained losses.

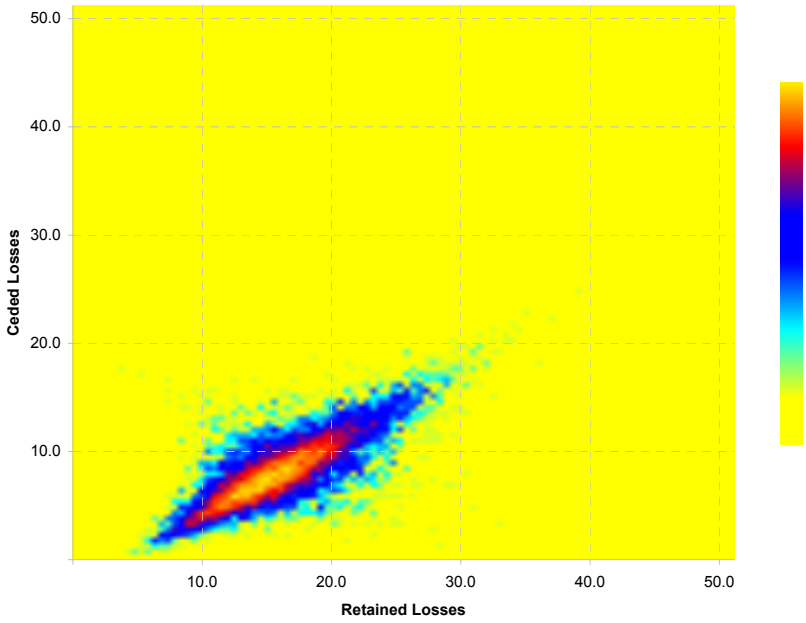

Figure 5.7: Iman-Conover approximation to bivariate distribution of ceded and retained losses using the t-copula as a reference distribution.

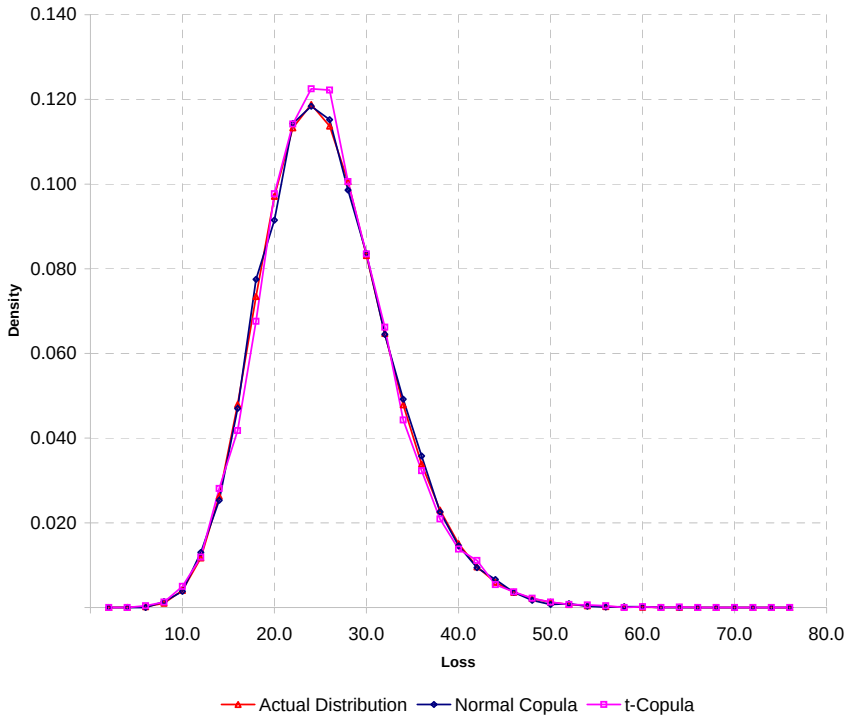

Figure 5.8: Distribution of total losses (ceded + retained) under normal copula, t-copula and actual.

Chapter 6

THEORETICAL UNDERPINNINGS OF THE IMAN-CONOVER METHOD

The theoretical foundations of the Iman-Conover method are elegantly justified by Vitale's Theorem [35]. We will state Vitale's theorem, explain its relationship to the IC method, and sketch the proof. The result should give a level of comfort to practitioners using a simulation approach to modeling multivariate distributions. It is not necessary to follow the details laid out here in order to understand and use the IC method, so the uninterested reader can skip the rest of the section. The presentation we give follows Vitale's original paper [35] closely.

Functional dependence and independence between two random variables are clearly opposite ends of the dependence spectrum. It is therefore surprising that Vitale's Theorem says that any bivariate distribution (U, V) can be approximated arbitrarily closely by a functionally dependent pair (U, TU) for a suitable transformation T.

In order to explain the set up of Vitale's theorem we need to introduce some notation. Let n be a power of 2. An interval of the form $((j-1)/n, j/n)$ for some $n \geq 1$ and $1 \leq j \leq n$ is called a dyadic interval of rank n. An invertible (Borel)

measure-preserving map which maps by translation on each dyadic interval of rank n is called a permutation of rank n. Such a T just permutes the dyadic intervals, so there is a natural correspondence between permutations of n elements and transformations T. If the permutation of dyadic intervals has a single cycle (has order n in the symmetric group) then T is called a cyclic permutation.

Theorem 3 *(Vitale) Let U and V be uniformly distributed variables. There is a sequence of cyclic permutations $T_1, T_2, \ldots$ such that $(U, T_n U)$ converges in distribution to (U, V) as $n \to \infty$.*

Recall convergence in distribution means that the distribution function of $(U, T_n U)$ tends to that of (U, V) at all points of continuity as $n \to \infty$.

The proof of Vitale's theorem is quite instructive and so we give a detailed sketch.

The proof is in two parts. The first constructs a sequence of arbitrary permutations T_n with the desired property. The second part shows it can be approximated with cyclic permutations. We skip the second refinement.

Divide the square $[0,1] \times [0,1]$ into sub-squares. We will find a permutation T such that the distributions of (U, V) and (U, TU) coincide on sub-squares. Reducing the size of the sub-squares will prove the result.

Fix n, a power of two. Let $I_j = ((j-1)/n, j/n)$, $j = 1, \ldots, n$. We will find an invertible permutation T such that

$$\Pr(U \in I_j, TU \in I_k) = \Pr(U \in I_j, V \in I_k) := p_{jk} \tag{6.1}$$

for $j, k = 1, \ldots, n$. Define

$$\begin{aligned} I_{j1} &= ((j-1)/n, (j-1)/n + p_{j1}) && (6.2)\\ I_{j2} &= ((j-1)/n + p_{j1}, (j-1)/n + p_{j1} + p_{j2}) && (6.3)\\ &\cdots && (6.4)\\ I_{jn} &= ((j-1)/n + p_{j1} + \cdots + p_{j,n-1}, j/n) && (6.5) \end{aligned}$$

and

$$\begin{aligned}
\tilde{I}_{j1} &= ((j-1)/n, (j-1)/n + p_{1j}) && (6.6)\\
\tilde{I}_{j2} &= ((j-1)/n + p_{1j}, (j-1)/n + p_{1j} + p_{2j}) && (6.7)\\
&\cdots && (6.8)\\
\tilde{I}_{jn} &= ((j-1)/n + p_{1j} + \cdots + p_{n-1,j}, j/n). && (6.9)
\end{aligned}$$

By construction the measure of I_{jk} equals the measure of $\tilde{I}_{kj}$. The invertible map T which sends each I_{jk} to $\tilde{I}_{kj}$ by translation is the map we need because

$$\begin{aligned}
\Pr(U \in I_j, T(U) \in I_k) &= \Pr(U \in I_j, U \in T^{-1}(I_k)) && (6.10)\\
&= \Pr(U \in I_j \cap T^{-1}(\bigcup_l \tilde{I}_{kl})) && (6.11)\\
&= \Pr(U \in \bigcup_l I_j \cap I_{lk}) && (6.12)\\
&= \Pr(U \in I_{jk}) && (6.13)\\
&= p_{jk}, && (6.14)
\end{aligned}$$

since the only I_{lk} which intersects I_j is I_{jk} by construction, and U is uniform. The transformation T is illustrated schematically in Table 6.1 for $n = 3$. The fact 3 is not a power of 2 does not invalidate the schematic!

If each p_{jk} is a dyadic rational then T is a permutation of the interval. If not then we approximate and use some more heavy duty results (a 1946 theorem of Birkhoff on representation by convex combinations of permutation matrices) to complete the proof.

Vitale's theorem can be extended to non-uniform distributions.

Corollary 1 *(Vitale) Let U and V be arbitrary random variables. There is a sequence of functions $S_1, S_2, \ldots$ such that (U, S_nU) converges in distribution to (U, V) as $n \to \infty$.*

Let F be the distribution function of U and G for V. Then $F(U)$ and $G(V)$ are uniformly distributed. Apply Vitale's theorem to get a sequence of functions T_n. Then $S_n = G^{-1}T_nF$ is the required transformation.

Table 6.1: Schematic of the Vitale transformation for $n = 3$

	$\tilde{I}_{33}$									p_{33}
I_3	$\tilde{I}_{32}$						p_{23}			
	$\tilde{I}_{31}$			p_{13}						
	$\tilde{I}_{23}$								p_{32}	
I_2	$\tilde{I}_{22}$					p_{22}				
	$\tilde{I}_{21}$		p_{12}							
	$\tilde{I}_{13}$							p_{31}		
I_1	$\tilde{I}_{12}$				p_{21}					
	$\tilde{I}_{11}$	p_{11}								
		I_{11}	I_{12}	I_{13}	I_{21}	I_{22}	I_{13}	I_{31}	I_{32}	I_{33}
		I_1			I_2			I_3		

Appendix A

SOFTWARE IMPLEMENTATIONS

Having laid out the IC method and given some explicit examples, we now discuss implementation issues. We will follow the Detailed Algorithm laid out in Section 4.2.

A.1 General Implementation and Design Considerations

A good general rule in writing software is to ensure that the steps which execute most frequently are coded as efficiently as possible. Cutting 50% from the execution time of a step which runs once and takes 1 second will have a barely perceptible impact. Cutting 50% from a step which takes 10 msecs, but executes 10,000 times will have a material and perceptible impact. See Hennessy and Patterson [18] for more discussion.

Matrix and linear algebra operations can be hard to code efficiently because of the design of modern computer chips and the strains matrix operations put on memory management. Modern CPUs have on-chip cache memory, which operates very quickly. Processors are "smart" enough to partially anticipate future memory calls and ensure the relevant locations have been pre-loaded into cache.

For example, arrays are usually stored in contiguous blocks of memory, and if you ask for $x[i]$, it is likely you will ask for $x[i+1]$. Processors will pull in a block of memory each side of $x[i]$ to speed operation. If the required memory is not in cache the CPU has a "cache-miss". These are very costly and result in a lot of lost processing time. Certain operations used extensively in the IC algorithm tend to generate lots of cache-misses: matrix multiplication being the worst (you pull in a row and a column; only one of these will be contiguous in memory). There are ways around these problems, but they are not ways you would want to navigate yourself! Fortunately, professional software designers spend a lot of effort to code matrix operations as efficiently as possible. Many processor benchmarks use matrix operations, so chip manufacturers have a vested interest here.

The Lapack [1] package is an example of a very efficiently coded set of matrix algebra operations. It is build on BLAS, Basic Linear Algebra Subprograms, which implements fundamental operations like matrix multiplication. Lapack implementations are available for most platforms, including Windows. See http://www.netlib.org/lapack for a non-commercial implementation. See http://-www.intel.com/software/products/mkl for a version optimized for Intel processors. It will automatically multi-thread operations if there are two or more CPUs available.

The implementation in Section 4.2 describes the appropriate Lapack functions for all the matrix operations, such as Choleski decomposition and solving a system of linear equations. I cannot claim that the implementation is optimal, but it is very fast.

A.2 SCARE

SCARE, a Simulating, Correlated Aggregation and Risk Engine, is a COM object (DLL) program which can be used from Excel/VBA. It implements the IC method, some other useful copula functions and bivariate normal and t distributions. It can be downloaded from www.mynl.com/wp. It was originally designed and implemented for Scor Re US, who have kindly given permission for it to

be made available to CAS members as part of the Working Party on Correlation effort.

Before programming with SCARE you need to install it and then reference it from your VBA project. Within the VBA editor, click Tools, References to bring up the References dialog. Make sure the SCARE library is selected, as shown in Figure A.1. In your application the location should show as C:/Program Files/Scare/Bin. Libraries are listed alphabetically, except those in use by an open project, which appear at the top.

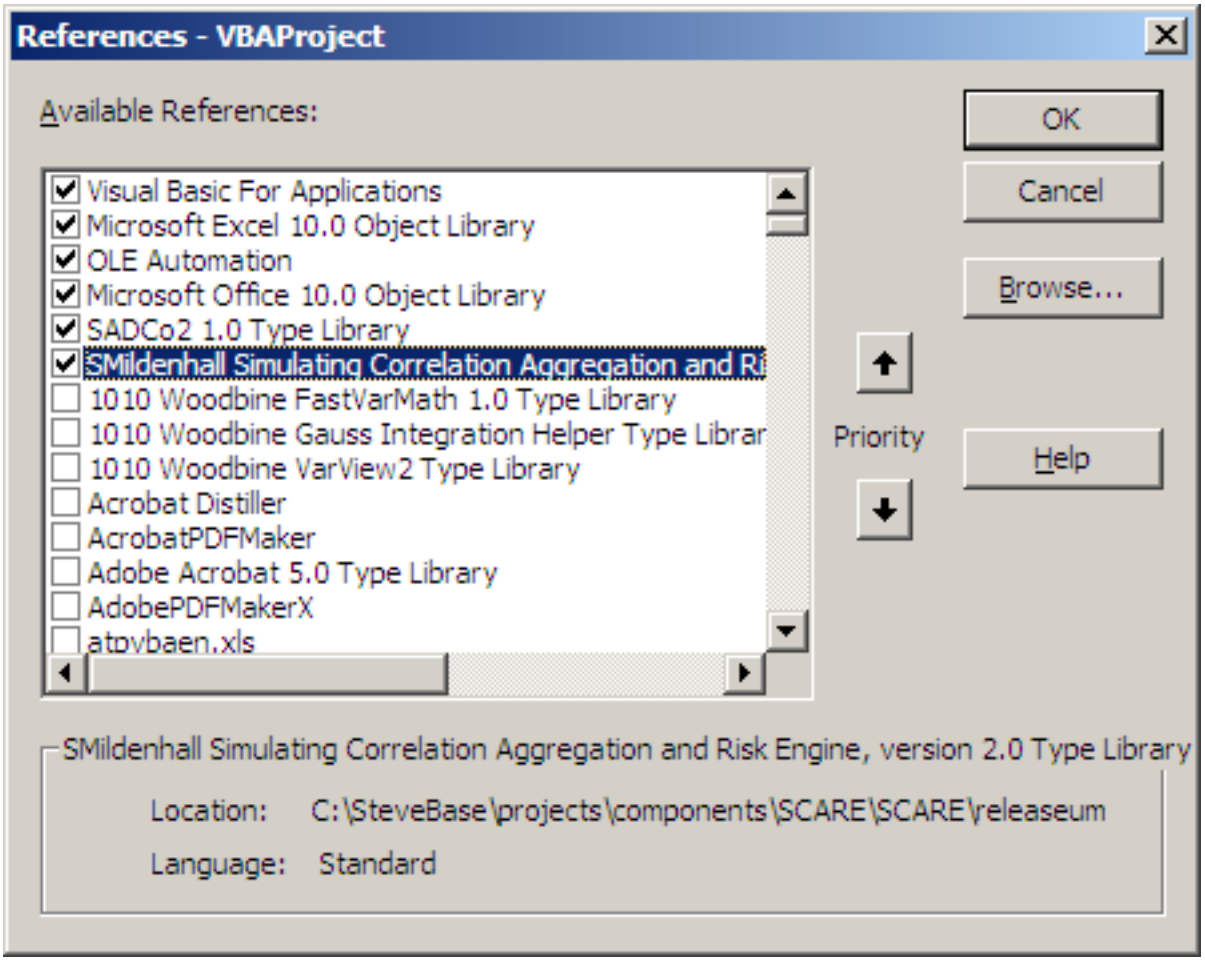

Figure A.1: Adding a reference to the SCARE component in Excel VBA.

A.3 SCARE Functions

All of the functions exposed by the SCARE DLL are described in detail in the User's Guide [30]. Here we give a quick overview of the key functions aimed at the Excel user. The SCARE DLL functions can call be referenced from VBA but they cannot be called directly from an Excel spreadsheet.

A.4 SCARE Wrapper Functions

A wrapper function "wraps" a COM function call so that it can be used directly from a worksheet[1]. For example, here is how we wrap the sort-by-aggregate function. The scSortByAgg function takes four arguments: the input multivariate array, an indicator to sort in ascending or descending order, a holder for the sorted output with the aggregate appended in the last column and a holder for the sorted output without the aggregate appended. Here is the VBA code.

```
Function scSortByAgg(v, ad As Long)
    Dim xx as new Shuffler
    Dim w, agg
    ' ad=1 ascending order, ad=-1 descending order
    xx.SortByAgg v, ad, w, agg
    scSortByAgg = w
End Function
```

The argument v is the multivariate sample and the argument ad is +/-1 for ascending or descending order respectively. Within the function, new variables w and agg are defined to hold the answers, and xx is defined as a new Shuffler object to access the member function. The SortByAgg method of Shuffler is then called. Finally scSortByAgg=w sets the answer to be returned to Excel.

In a spreadsheet, the function would be called as =scSortByAgg(A1:D100,1) input as an array function in a range large enough to hold the answer. Array functions are entered using control+shift+enter, rather than just enter. They appear in the spreadsheet as {=scSortByAgg(A1:D100,1)}.

[1]Excel 2002 will automatically create wrappers for all functions using Tools, Add-Ins and selecting the Automation button.

The implementation in the Wrappers module of SCARE.xla uses a single variable xx which is shared between all the functions in the work book. It is defined as a private variable at the beginning of the workbook.

Several variables below are defined as "variants". Variants are a useful hold-all variable type in VB/VBA. Almost all method output variables are Variants. They can hold a single number, a string, an array of numbers, or a mixed array, or even a reference to an entire open Excel application. Code such as v = Range("a1:b10").value will set v equal to a 2 x 10 variant array. Depending on the contents of the range it could be an array of doubles or an array of variants. Code like set v = Range("a1:b10") sets v equal to a reference to the range object. It is then possible to write v.ClearContents to clear Range("a1:b10") or v.Value = 10 to set all the cells A1:B10 equal to the value 10. Variants need to be used with care. In some situations they are convenient, fast and efficient—but in others they are convenient, slow and inefficient. Their use in SCARE is confined to the former.

The main various functions in SCARE.xla are as follows.

```
Function scSortByAgg(v, ad As Long)
```

Sums the input $n \times r$ multivariate density over columns to get an $n \times 1$ aggregate. Sorts the whole input array by the aggregate. Use ad=1 to sort in ascending order, and ad=-1 for descending order.

```
Public Function scCholeski(x As Variant) As Variant
```

Returns the Choleski decomposition of the input $r \times r$ matrix x. Note that the C++ object only populates the upper half of the matrix. The VBA method "tidies-up" that return by zero filling the lower portion.

```
Function scCorr(v As Variant) As Variant
```

Computes the mean by column, covariance matrix and correlation matrix of the input $n \times r$ multivariate density v. Only the correlation matrix is returned to Excel, but it would be easy for the user to alter to return the means vector or the covariance matrix.

```
Public Function scIsCovMat(x As Variant) As Boolean
```

Tests input matrix X to determine if it is positive definite. Returns true or false.

```
Function scNormDist(x As Double) As Double
```

Computes $\Phi(x)$ the standard normal distribution function evaluated at x. It is more accurate, and from VBA quicker, than the worksheet function NormDist.

```
Function scNormInv(x As Double) As Double
```

Computes the inverse standard normal probability distribution function at probability level x. Also quicker and more accurate than the built in functions.

```
Function scTDist(nu As Long, x As Double) As Double
```

Computes the t-distribution function with ν degrees of freedom at x.

```
Function scTInv(nu As Long, x As Double) As Double
```

Computes the inverse to the t distribution function with ν degrees of freedom at probability level x.

```
Function scBVN(h As Double, k As Double, rho As Double) As Double
```

Computes the probability $\Pr(X < h, Y < k)$ where (X, Y) have a bivariate normal distribution with standard normal marginals and correlation coefficient ρ.

```
Function scBVT(nu As Long, h As Double, k As Double, rho As Double)
As Double
```

Computes the bivariate t distribution function $\Pr(X < h, Y < k)$ where (X, Y) have a bivariate t distribution with $n\nu$ degrees of freedom and correlation ρ.

```
Function scQuickShuffle(rgIn, corrIn) As Variant
```

Returns the input $n \times r$ range rgIn re-ordered to have correlation approximately equal to the $r \times r$ correlation matrix corrIn.

```
Function scQuickShuffleParam(spec, n As Long, covMat) As Variant
```

As scQuickShuffle, except the input values are parameters for shifted lognormal variables. The spec input is a $n \times r$ array where there are n input lines of business and nine columns. The meaning of the nine columns is as follows:

1. Not used
2. ± 1, where $+1$ is used to represent losses and -1 liabilities
3. μ
4. σ
5. s, the shift parameters
6. 0 or 1 indicator where 1 means there is layer and attachment information for the current row.
7. Layer value
8. Attachment value; the sample is from a shifted lognormal with parameters s, μ and σ, conditional on losses being greater than the attachment. The attachment is subtracted and losses are limited by layer value input.
9. Not used.

Appendix B

VBA CODE FOR INDEXING

```
Private Sub indexx(n As Long, arr, colNo As Long, indx() As Long)

'  Indexes an array arr[1..n], i.e., outputs the array indx[1..n] such
'  that arr[indx[j]] is in ascending order for j = 1, 2, . . . ,N. The
'  input quantities n and arr are not changed.  Translated from [31].

    Const m As Long = 7
    Const NSTACK As Long = 50

    Dim i As Long, indxt As Long, ir As Long, itemp As Long, j As Long
    Dim k As Long, l As Long
    Dim jstack As Long, istack(1 To NSTACK) As Long
    Dim a As Double

    ir = n
    l = 1

    For j = 1 To n
        indx(j) = j
    Next j

    Do While 1
        If (ir - l < m) Then
            For j = l + 1 To ir
                indxt = indx(j)
                a = arr(indxt, colNo)
                For i = j - 1 To l Step -1
                    If (arr(indx(i), colNo) <= a) Then Exit For
                    indx(i + 1) = indx(i)
```

```
            Next i
            indx(i + 1) = indxt
        Next j
        If (jstack = 0) Then Exit Do
        ir = istack(jstack)
        jstack = jstack - 1
        l = istack(jstack)
        jstack = jstack - 1
    Else
        k = (l + ir) / 2
        itemp = indx(k)
        indx(k) = indx(l + 1)
        indx(l + 1) = itemp
        If (arr(indx(l), colNo) > arr(indx(ir), colNo)) Then
            itemp = indx(l)
            indx(l) = indx(ir)
            indx(ir) = itemp
        End If
        If (arr(indx(l + 1), colNo) > arr(indx(ir), colNo)) Then
            itemp = indx(l + 1)
            indx(l + 1) = indx(ir)
            indx(ir) = itemp
        End If
        If (arr(indx(l), colNo) > arr(indx(l + 1), colNo)) Then
            itemp = indx(l)
            indx(l) = indx(l + 1)
            indx(l + 1) = itemp
        End If
        i = l + 1
        j = ir
        indxt = indx(l + 1)
        a = arr(indxt, colNo)
        Do While 1
            Do
                i = i + 1
            Loop While (arr(indx(i), colNo) < a)
            Do
                j = j - 1
            Loop While (arr(indx(j), colNo) > a)
            If (j < i) Then Exit Do
            itemp = indx(i)
            indx(i) = indx(j)
            indx(j) = itemp
        Loop
        indx(l + 1) = indx(j)
```

```
            indx(j) = indxt
            jstack = jstack + 2
            If (jstack > NSTACK) Then MsgBox (‘‘NSTACK too small in indexx.")
            If (ir - i + 1 >= j - l) Then
                istack(jstack) = ir
                istack(jstack - 1) = i
                ir = j - 1
            Else
                istack(jstack) = j - 1
                istack(jstack - 1) = l
                l = i
            End If
        End If
    Loop

End Sub
```

Bibliography

[1] E. Anderson. *LAPACK Users' Guide (www.netlib.org/lapack)*. SIAM, 1999.

[2] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. Coherent measures of risk. *Math. Finance*, pages 203–28, 1999.

[3] P. Billingsley. *Probability and Measure*. J. Wiley and Sons, second edition, 1986.

[4] N. Bowers, H. Gerber, J. Hickman, D. Jones, and C. Nesbitt. *Actuarial Mathematics*. Society of Actuaries, 1986.

[5] D. R. Clark and C. A. Thayer. A primer on the exponential family of distributions. *Casualty Actuarial Society Spring Forum*, pages 117–148, 2004.

[6] W. J. Conover. *Practical nonparametric statistics*. John Wiley and Sons, third edition, 1999.

[7] C. D. Daykin, T. Pentikainen, and M. Pesonen. *Practical Risk Theory for Actuaries*. Chapman and Hall, 1994.

[8] P. Diaconis. Patterns in eigenvalues: The 70th Josiah Willard Gibbs Lecture. *Bulletin of the American Mathematical Society*, 40(2):155–178, 2003.

[9] P. Embrechts, F. Lindskog, and A. McNeil. Modelling dependence with copulas and applications to risk management. *ETHZ Zurich*, 2001.

[10] P. Embrechts, A. McNeil, and D. Straumann. Correlation and dependency in risk managment: Properties and pitfalls. *ETHZ Zurich*, 1998.

[11] K.-T. Fang and Y.-T. Zhang. *Generalized Multivariate Analysis*. Springer-Verlag/Science Press Beijing, 1990.

[12] W. Feller. *An Introduction to Probability Theory and Its Applications, two volumes*. J. Wiley and Sons, second edition, 1971.

[13] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, third edition, 1996.

[14] L. Goodman and W. Kruskal. Measures of association for cross-classifications. iii: Approximate sample theory. *Journal of ASA*, 58:310–364, 1963.

[15] J. Grandell. *Mixed Poisson Processes*. Chapman and Hall, 1977.

[16] R. W. Hamming. *Digital Filters*. Dover, third edition, 1989.

[17] P. Heckman and G. G. Meyers. The calculation of aggregate loss distributions from claim severity and claim count distributions. *Proceedings of the Casualty Actuarial Society*, LXXX:22–61, 1983.

[18] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, second edition, 1996.

[19] T. Herzog. Part 3 study note. *Education and Examination Committee of the Society of Actuaries*, 1984.

[20] D. L. Homer and D. R. Clark. Insurance applications of bivariate distributions. *Proceedings of the Casualty Actuarial Society*, XC:274–307, 2003.

[21] R. Iman and W. Conover. A distribution-free approach to inducing rank correlation amoung input variables. *Commun. Statist. Simula. Computa.*, 11(3):311–334, 1982.

[22] N. L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions*, volume 1. John Wiley and Sons, second edition, 1994.

[23] N. L. Johnson, S. Kotz, and N. Balakrishnan. *Discrete Multivariate Distributions*. J. Wiley and Sons, 1996.

[24] N. L. Johnson, S. Kotz, and A. W. Kemp. *Univariate Discrete Distributions*. J. Wiley and Sons, second edition, 1992.

[25] S. A. Klugman, H. H. Panjer, and G. E. Willmot. *Loss Models, From Data to Decisions*. John Wiley and Sons, 1998.

[26] D. Knuth. *The Art of Computer Programming, vol. 2 Seminumerical Algorithms*. Addison-Wesley, 1999.

[27] S. Kotz, T. Kozubowski, and K. Podgorski. *The Laplace Distribution and Generalizations*. Birkhauser, 2001.

[28] T. Mack. The stochastic model underlying the chain ladder method. *ASTIN Bulletin*, Vol. 21(1):22–61, 1991.

[29] C. Menn and S. T. Rachev. Calibrated FFT-based density approximations for α-stable distributions. *Elsevier Science*, 2004.

[30] S. J. Mildenhall. *SCARE: Simulating Correlated Aggregation and Risk Engine User's Guide*. www.mynl.com/wp, 2002.

[31] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipies in C, The Art of Scientific Computing*. Cambridge University Press, second edition, 1992.

[32] D. Sornette, P. Simonetti, and J. V. Andersen. ϕ^q-field theory for portfolio optimization: "fat-tails" and non-linear correlations (working title: Non-linear covariance matrix and portfolio theory for non-gaussian multivariate distributions). *Physics Report*, 335(2):19–92, 2000.

[33] G. G. Venter. Tails of copulas. *Proceedings of the Casualty Actuarial Society*, LXXXIX:68–113, 2002.

[34] R. J. Verrall. A Bayesian generalized linear model for the Bornhuetter-Ferguson method of claims resering. *NAAJ*, 8 No. 3:67–89, 2004.

[35] R. Vitale. *On Stochastic Dependence and a Class of Degenerate Distributions*, pages 459–469. In Topics in Statistical Dependence, Edited by H. Block, A. Sampson and T. Savits. Institute of Mathematical Statistics, Hayward CA, 1990.

[36] S. Wang. Premium calculation by transforming the layer premium density. *ASTIN Bulletin*, 26:71–92, 1995.

[37] S. Wang. Aggregation of correlated risk portfolios: Models and algorithms. *Proceedings of the Casualty Actuarial Society*, LXXXV:848–939, 1998.

[38] S. Wang. A class of distortion operators for pricing financial and insurance risks. *Journal of Risk and Insurance*, 67(1):15–36, 2000.

Biography

Stephen Mildenhall is Executive Vice President at Aon Re Services in Chicago IL. He has an undergraduate degreee in Mathematics from the University of Warwick, UK, and a masters and PhD in Mathematics from the University of Chicago, IL. He is a Fellow of the Casualty Actuarial Society, an Associate of the Society of Actuaries and Member of the American Academy of Actuaries. His research interests include capital allocation and applications of statistics to reserving and ratemakeing problems.