



Quantifying Portfolio Credit Risk Using CCruncher

Gerard Torrent-Gironella

Version 2.4

Abstract

This document provides a comprehensible view of portfolio credit risk modeling using the copula approach (t-Student case). It emphasizes the practical aspects, including short introductions of methods, examples, and a discussion of the implementation details. It takes into account the sensitivity of the risk with respect to the estimated parameters and provides a practical method to incorporate this uncertainty in the portfolio credit risk measure. Finally, a method for optimizing the portfolio credit risk is provided.

The BibTeX entry of this document for L^AT_EX users is

```
@manual{ccruncher,  
  title = {{Quantifying Portfolio Credit Risk Using CCruncher}},  
  author = {Torrent-Gironella, Gerard},  
  organization = {Tatine},  
  address = {Barcelona, Catalonia},  
  year = {2014},  
  month = Mar,  
  url = {http://www.ccruncher.net/},  
  note = {Version 2.4}  
}
```

Version 2.4 [R1213] March 2014

Version 2.3 [R1151] September 2013

Copyright © 2013–2014 Gerard Torrent-Gironella

This work is licensed to the public under Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/>. We have invested much time and effort in creating CCruncher; please cite it when using it for credit risk analysis.

Contents

	Glossary	5
1	Introduction	7
2	Modeling Default Times	9
2.1	The Copula Model	9
2.2	The Multi-Factor Formulation	10
2.3	The Bernoulli Mixture Model	12
2.4	Frequentist Inference	15
2.5	Bayesian Inference	17
3	Modeling Portfolio Loss	23
3.1	Portfolio Loss Distribution	23
3.2	Credit Risk Measurement	25
3.3	Simulation Details	30
4	Additional Topics	33
4.1	Risk Sensitivity	33
4.2	Parameters Uncertainty	35
4.3	Portfolio Optimization	38
A	Appendices	43
A.1	The Probabilities of Default	43
A.2	Copula Basics	48
A.3	Multivariate t-Student Distribution	49
A.4	Covariance Block Matrix	51
A.5	Multi-Factor Gaussian Model	52
A.6	Metropolis-Hastings Algorithm	54
A.7	Variance Reduction	57
	Bibliography	60

Glossary

n	Number of obligors in the portfolio.
s	Number of sectors/factors.
r	Number of ratings.
n_i	Number of obligors in the i -th sector/factor. It is a component of vector \vec{n} that satisfies $\sum_{i=1}^s n_i = n$.
n_{ij}^t	Number of obligors in the i -th sector/factor having the j -th rating at the beginning of the t -th period.
k_{ij}^t	Number of defaults in the i -th sector/factor having the j -th rating at the beginning of the t -th period.
T_i	Default time (aka time-until-default, aka time-to-default, aka survival time) of the i -th obligor.
F	Multivariate distribution of default times or time-until-default. It has n components with marginal distributions F_i .
F_i	Default time distribution of the i -th obligor. Sometimes refers to default time distribution of the i -th rating because obligor's F_i depends only on his rating.
p_i	Probability that i -th obligor/rating defaults within the next 1-year, i.e., $F_i(1)$.
\widehat{R}	Obligors' correlation block matrix of dimension $n \times n$.
R	Factor correlation matrix of dimension $s \times s$.
\vec{w}	Factor loadings, vector of dimension s .
$B_s(\vec{n}, \vec{d}, M)$	Covariance $s \times s$ block matrix in which \vec{n} are the number of individuals per block, \vec{d} are the diagonal values, and M are the non-diagonal block values. See section A.4.
$t_d(v, \vec{\mu}, \Sigma)$	Multivariate t-Student distribution of dimension d with v degrees of freedom, mean vector $\vec{\mu}$, and dispersion or scatter matrix Σ .
$C_{v, \widehat{R}}^t$	t-Student copula with v degrees of freedom and correlation matrix \widehat{R} .
$C_{v, \vec{w}, R}^t$	Multi-factor t-Student copula with v degrees of freedom, factor loadings \vec{w} , and factor correlation matrix R . It is merely a t-Student copula.

Notes

1. Factors = Sectors. CCruncher considers that dependence between obligors is determined by obligor's industrial sectors causing that sectors and factors be equivalents. This text uses sectors and factors indistinctly to refer to factors.
2. Time unit. We use the 1-year time unit without loss of generality in order to simplify the exposition.

1 — Introduction

CCruncher is a project for quantifying portfolio credit risk. It was developed and is maintained by Tatine, a consulting firm located in Barcelona (Catalonia). The latest version of the project can be downloaded from the CCruncher's site, <http://www.ccruncher.net>.

The CCruncher framework comprises this document and a piece of software. Both elements are released under open-source type licenses. CCruncher is a mature and active project in which users may participate by sending suggestions and contributions to gtorrent@ccruncher.net. Because CCruncher is an evolving project, the version identifier should be mentioned when referring to the project name (e.g., CCruncher-2.4).

This document provides a comprehensible view of portfolio credit risk modeling using the copula approach (t-Student case). This document emphasizes the practical aspects, including short introductions of methods, examples, and a discussion of the implementation details.

Portfolio credit risk tries to respond to questions such as: “What is the expected portfolio loss for the next year?” and “How much could I lose in a really bad year?” There are several approaches to answering these questions. All of them attempt to ascertain the portfolio loss distribution and measure the credit risk using a set of selected statistics (e.g., mean, quantile), but they differ in how to obtain this distribution. The following list briefly present the most used models [8, 5, 13][3, chap. 2.4].

Actuarial models consider that the frequency of defaults (Poisson type distributions) is the proper manner in which to analyze observed defaults. Compounding default frequency with the obligors' exposure gives the portfolio loss distribution. The main representative of this approach is the CreditRisk+ model [38].

Intensity models consider that an obligor defaults when occurs the first jump of a Poisson process with random intensity given by a Cox process. Dependency between defaults is induced by assuming that each intensity is a function of a common process and an idiosyncratic process. Actuarial models may be included within this family to the extent that they have a static intensity.

Bernoulli mixture models represent the i -th obligor's default using a Bernoulli variable (with probability of default p_i). These variables are conditionally independent given a vector of default probabilities with a known density. The sum of the defaulted obligors' exposure gives the portfolio loss distribution.

Factor models represent the i -th obligor using a random variable and consider that this obligor defaults when this variable crosses a preset threshold. These variables depend on a low-dimensional random vector of common factors. Conditional on factors, factor models are Bernoulli mixture models. CreditMetricsTM [19] and KMV [7] are models of this type.

Copula models represent the i -th obligor using a random variable and consider that this obligor defaults when this variable satisfies a given condition. The dependence between these variables is explicitly modeled using a copula [11]. The sum of the defaulted obligors' exposure gives the portfolio loss distribution. In the Li copula model [25], random variables are the obligors default times and the default trigger is to cross a preset threshold. The multi-factor Gaussian case of the Li model covers the CreditMetricsTM model.

CCruncher uses the copula approach to formulate the model, then restrict the model into a factor

model in order to reduce the dimensionality, and finally interprets the model as a Bernoulli mixture model to estimate the parameters. The CCruncher model implements the t-Student multi-factor copula even for high-dimensional portfolios, and it is formulated according to the Basel II concepts (PD, EAD, LGD).

CCruncher quantifies the portfolio credit risk by sampling the portfolio loss distribution and computing the appropriate statistics (EL, VaR, ES). The portfolio losses are obtained simulating the time-until-default of obligors and simulating the EADs and LGDs of their assets. The time-until-default of obligors is a multivariate distribution in which the marginals are the probabilities of default given by obligors' ratings and in which the dependence between obligors is determined by obligors' industrial sectors. CCruncher assumes that the dependence structure is a t-Student copula, specifically a t-Student multi-factor copula in which the factors are the industrial sectors.

This document details the obligors' default times model, gives methods to estimate the model parameters, details the procedure to simulate the portfolio loss distribution, and provides the formulas to quantify the credit risk. Furthermore, it explores the risk sensitivity for some parameters, encourages transferring the parameters uncertainty to the credit risk measures, and exposes a method to optimize the portfolio credit risk.

CCruncher requires a portfolio with a buy-and-hold policy in which each obligor has a rating assigned and belongs to a single economic sector. An example of such a credit portfolio could be an SMEs loan portfolio (buy-and-hold policy, rated obligors, and multiple sectors in which each obligor belongs to a single sector) or a portfolios of bonds. Unlike CreditMetrics™, CCruncher does not follow a mark-to-market account, but rather follows a two-state model (default/no-default).

The advantages of CCruncher are as follows: 1) the model allows multiple factors; 2) the model allows multiple ratings; 3) obligors can have multiple assets; 4) each asset has its own EAD and LGD; 5) these EADs and LGDs can be stochastic values; 6) EADs and LGDs can vary over time; 7) CCruncher allows risk disaggregation; 8) CCruncher does not presuppose a uniform transition matrix; 9) factor correlations are appropriately considered; and 10) the uncertainty of parameters is considered.

The disadvantages of CCruncher are as follows: 1) it is computationally intensive; 2) the convergence rate of the Monte Carlo method is rather slow; 3) currently the obligors cannot belong to a weighted sum of factors; 4) dependence only relies on sectors; 5) correlations are constant over time; 6) currently only Gaussian and t-Student copulas are supported; and 7) mark-to-market is not supported.

The content is organized into three chapters. The first chapter details the distribution used to simulate the obligors' default times, and provides methods to estimate the copula parameters. The second chapter describes how to obtain the portfolio loss distribution and quantify the portfolio credit risk. Finally, the third chapter covers three important topics: risk sensitivity, parameters uncertainty, and portfolio optimization. In order to facilitate reading, those issues that are important to understanding the content and require a detailed discussion have been relegated to appendices.

2 — Modeling Default Times

Financial institutions have statistical models that assigns a probability of default to each obligor as well as statistical models for the exposure and recovery of each operation. These models allow us to estimate the expected loss of the portfolio, but they have nothing to say about measures related to economic capital such as VaR or ES. This is because these measures strongly depend on the dependence between obligors, and models mentioned above do not refer to this concept. This chapter develops the multivariate model exploiting the existing PD models.

2.1 The Copula Model

The multivariate time-until-default model described in this section is identical to that described in [25, 14] [3, chap. 2.6]. We recommend perusing the appendices before proceeding. The appendices contain definitions, statements, and notation regarding subjects used in this section.

The copula approach considers that the lifetimes of obligors are the proper manner in which to explain the observed defaults. Obligor's default times (T_1, \dots, T_n) are modeled by a multivariate distribution F such that

$$F(t_1, \dots, t_n) = \Pr\{T_1 \leq t_1, \dots, T_n \leq t_n\}$$

in which n is the number of obligors in the portfolio and T_i is the time-until-default (in years) of the i -th obligor. Applying the Sklar's theorem, we obtain

$$F(t_1, \dots, t_n) = C(F_1(t_1), \dots, F_n(t_n))$$

in which marginal $F_i(t) = \Pr\{T_i \leq t\}$ is the time-until-default distribution of the i -th obligor (see appendix A.1 to obtain this distribution from existing PD models) and C is a copula. CCruncher considers that C is a t-Student copula for the following practical reasons:

- it is determined by the correlation matrix (elliptical property).
- it is symmetrical (elliptical property), allowing application of the antithetic reduction variance technique in the Monte Carlo simulation.
- it allows a wide range of dependence structures by parameter ν .
- it exhibits tail dependence when $\nu \neq \infty$.
- it can be expressed as a t-Student multi-factor copula.
- an efficient simulation algorithm exists.
- the Gaussian case ($\nu = \infty$) leads to the CreditMetrics™ model.

The following proposition reformulates F in terms of the multivariate t-Student distribution. This result, combined with the t-Student multi-factor model, provides an efficient simulation algorithm and allows estimating the parameters.

Proposition 2.1 — Multivariate default times distribution. Let $T = (T_1, \dots, T_n)$, the time-until-default of obligors with marginals F_1, \dots, F_n and a t-Student copula $C_{\nu, \hat{R}}^t$. Then its distribution F is

$$F(t_1, \dots, t_n) = H(t_{\nu}^{-1}(F_1(t_1)), \dots, t_{\nu}^{-1}(F_n(t_n)))$$

in which H is the cdf of the multivariate t-Student distribution $t_n(\mathbf{v}, \vec{0}, \widehat{R})$, and t_v^{-1} is the inverse of the univariate t-Student distribution $t_1(\mathbf{v}, 0, 1)$.

Proof. We note that $X = (X_1, \dots, X_n)$ a vector having a multivariate t-Student distribution $t_n(\mathbf{v}, \vec{0}, \widehat{R})$ and (U_1, \dots, U_n) to represent the components of the copula $C_{\mathbf{v}, \widehat{R}}^t$.

$$\begin{aligned}
F(t_1, \dots, t_n) &= \Pr\{T_1 \leq t_1, \dots, T_n \leq t_n\} = \\
&\text{we use the corollary A.3: } (T_1, \dots, T_n) = (F_1^{-1}(U_1), \dots, F_n^{-1}(U_n)) \\
&= \Pr\{F_1^{-1}(U_1) \leq t_1, \dots, F_n^{-1}(U_n) \leq t_n\} = \\
&\text{we use the corollary A.2: } (U_1, \dots, U_n) = (t_v(X_1), \dots, t_v(X_n)) \\
&= \Pr\{F_1^{-1}(t_v(X_1)) \leq t_1, \dots, F_n^{-1}(t_v(X_n)) \leq t_n\} = \\
&= \Pr\{t_v(X_1) \leq F_1(t_1), \dots, t_v(X_n) \leq F_n(t_n)\} = \\
&= \Pr\{X_1 \leq t_v^{-1}(F_1(t_1)), \dots, X_n \leq t_v^{-1}(F_n(t_n))\} = \\
&= H(t_v^{-1}(F_1(t_1)), \dots, t_v^{-1}(F_n(t_n)))
\end{aligned}$$

□

Corollary 2.1 — Default times simulation (I). Let $T = (T_1, \dots, T_n)$, a random vector with marginals F_1, \dots, F_n and a t-Student copula $C_{\mathbf{v}, \widehat{R}}^t$. Then we can simulate this distribution using a multivariate t-Student distribution $X \sim t_n(\mathbf{v}, \vec{0}, \widehat{R})$:

$$(T_1, \dots, T_n) = (F_1^{-1}(t_v(X_1)), \dots, F_n^{-1}(t_v(X_n)))$$

in which t_v is the cdf of the univariate t-Student distribution $t_1(\mathbf{v}, 0, 1)$.

The current situation is not yet satisfactory because the dimension $n \times n$ of the correlation matrix \widehat{R} indicates that any attempt to simulate random values or estimate the parameters cannot be performed in practice when n is large. For these reason we restrict the copula model to the multi-factor formulation.

2.2 The Multi-Factor Formulation

To develop a correlation matrix that is easy to handle, we assume that a reduced number s of industrial sectors exists and that each obligor belongs to a single sector. In addition, we assume that obligors' time-until-default dependence only relies on the involved industrials sectors. With these restrictions and by sorting the obligors by sector, we see that the obligors' time-until-default Spearman's rank correlation is an $n \times n$ correlation block matrix $\widetilde{R} = B_s(\vec{n}, \vec{1}, \widetilde{\Sigma})$.

$$\begin{aligned}
\widetilde{R} = B_s(\vec{n}, \vec{1}, \widetilde{\Sigma}) &= \begin{pmatrix} 1 & \cdots & \sigma_{11} & & \sigma_{1s} & \cdots & \sigma_{1s} \\ \vdots & \ddots & \vdots & \cdots & \vdots & & \vdots \\ \sigma_{11} & \cdots & 1 & & \sigma_{1s} & \cdots & \sigma_{1s} \\ & \vdots & & \ddots & \vdots & & \\ \sigma_{1s} & \cdots & \sigma_{1s} & & 1 & \cdots & \sigma_{ss} \\ \vdots & & \vdots & \cdots & \vdots & \ddots & \vdots \\ \sigma_{1s} & \cdots & \sigma_{1s} & & \sigma_{ss} & \cdots & 1 \end{pmatrix} & \widetilde{\Sigma} = \begin{pmatrix} \sigma_{11} & \cdots & \sigma_{1s} \\ \vdots & \ddots & \vdots \\ \sigma_{1s} & \cdots & \sigma_{ss} \end{pmatrix} \\
& & \vec{n} = (n_1, \dots, n_s) \\
& & n = \sum_{i=1}^s n_i
\end{aligned}$$

We use the Spearman's rank correlation because it is a measure of dependence that does not depend on marginals (F_i in our case), unlike the usual Pearson's correlation, which does depend

on marginals. t-Student copulas with Spearman's rank correlation block matrix are those that come from a multivariate t-Student $t_n(\mathbf{v}, \vec{0}, \widehat{R})$ in which $\widehat{R} = B_s(\vec{n}, \vec{1}, \Sigma)$ ¹. Consequently, the multivariate time-until-default distribution F can be expressed by its marginals F_i and a t-Student copula $C_{\mathbf{v}, \widehat{R}}^t$ in which $\widehat{R} = B_s(\vec{n}, \vec{1}, \Sigma)$.

As we have observed, proposition 2.1 establishes the link between the t-Student copula and the multivariate t-Student distribution. Now, we are interested in formulating this multivariate t-Student distribution with correlation matrix $\widehat{R} = B_s(\vec{n}, \vec{1}, \Sigma)$ as a t-Student multi-factor model.

The multivariate t-Student distribution relies on the multivariate Gaussian distribution (see proposition A.6). The multivariate Gaussian with a correlation block matrix can be expressed as a multi-factor model in most cases (see proposition A.12). The cases that aren't covered are those that contain negative intra-sectorial correlations and those containing inter-sectorial correlations that are too high. The number of these cases tends toward 0 when the number of obligors by sector tends to ∞ (see proposition A.13). Because we are modeling sectors composed of thousands of obligors (although our portfolio may contain only a few), we can assume that for our purposes, the t-Student multi-factor model maps the multivariate t-Student distribution.

Proposition 2.2 — t-Student distr. as a multi-factor model. Let a random vector $X = (X_1, \dots, X_n)$ be with a multivariate t-Student distribution $t_n(\mathbf{v}, \vec{0}, \widehat{R})$ in which $\widehat{R} = B_s(\vec{n}, \vec{1}, \Sigma)$ is a correlation block matrix and n_i sizes are high enough. Then X can be written as a t-Student multi-factor model as follows:

$$X_i^j = \sqrt{\frac{\mathbf{v}}{V}} \cdot \left(w_i \cdot Z_i + \sqrt{1 - w_i^2} \cdot \varepsilon_i^j \right), \text{ where } \begin{cases} i = 1, \dots, s & s \text{ is the number of factors} \\ j = 1, \dots, n_i & n_i \text{ is } i\text{-th factor size} \\ V \sim \chi_{\mathbf{v}}^2 & 2 \leq \mathbf{v} \text{ degrees of freedom} \\ Z \sim N(\vec{0}, R) & R \text{ a } s \times s \text{ correlation matrix} \\ w_i \in (0, 1) & \text{the factor loadings} \\ \varepsilon_i^j \sim N(0, 1) \text{ iid} & \forall i, j \\ V, Z, \varepsilon_i^j \text{ indep.} & \forall i, j \end{cases}$$

The relationship between the multi-factor coefficients \vec{w} , R and the multivariate t-Student correlation matrix $\widehat{R} = B_s(\vec{n}, \vec{1}, \Sigma)$ is

$$\begin{aligned} R_{ij} &= \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii} \cdot \Sigma_{jj}}} & \forall i, j \\ w_i &= \sqrt{\Sigma_{ii}} & i = 1, \dots, s \end{aligned}$$

We call t-Student multi-factor copula to the copula of the t-Student multi-factor model. As we have observed, the t-Student multi-factor copula ($C_{\mathbf{v}, \vec{w}, R}^t$) is merely a t-Student copula ($C_{\mathbf{v}, \widehat{R}}^t$).

Note that we expressed the multi-factor model in terms of the factor loadings and the factor correlation matrix instead of the more concise form based on the covariance matrix. We do this because the factor loadings have a treatment distinct from the factor correlation in the estimation procedure.

The t-Student multi-factor model combined with the corollary 2.1 provides an efficient algorithm to simulate the obligors' default times disclosed in the next corollary. Later we will use this algorithm to simulate the portfolio losses.

¹In the Gaussian case, the relation between Spearman's and Pearson's correlations is $\rho_S = \frac{6}{\pi} \cdot \arcsin\left(\frac{\rho}{2}\right)$.

Corollary 2.2 — Default times simulation (II). Let $T = (T_1, \dots, T_n)$ be a random vector with marginals F_1, \dots, F_r and a t-Student multi-factor copula with ν degrees of freedom, factor loadings \vec{w} , and factor correlation matrix R . Then we can simulate this distribution:

1. Simulate the factors, $\vec{z} \sim N(\vec{0}, R)$.
2. Simulate the degrees of freedom, $\nu \sim \chi_\nu^2$.
3. For each obligor, repeat the following steps:
 - (a) Simulate $\varepsilon \sim N(0, 1)$.
 - (b) Simulate the multivariate t-Student component using the obligor's sector:

$$x = \sqrt{\frac{\nu}{\nu - 1}} \cdot \left(w_i \cdot z_i + \sqrt{1 - w_i^2} \cdot \varepsilon \right)$$
 - (c) Obtain the obligor's default time using the obligor's rating:

$$t = F_j^{-1}(t_\nu(x))$$

We want to estimate the copula parameters of the multivariate default times distribution. These parameters are the degrees of freedom ν , the factor loadings \vec{w} , and the factor correlation matrix R . The historical data available for parameters calibration are panel data type. Calibration with such data leads to the interpretation of the current model as a Bernoulli mixture model.

2.3 The Bernoulli Mixture Model

The available historical data lead us to interpret obligors' defaults as conditionally independent Bernoulli random variables. In this section we reinterpret the multi-factor copula model as a Bernoulli mixture model in order to estimate the copula parameters.

Definition 2.1 — Annual number of defaults. We define the annual number of defaults by sector and rating such as a discrete multivariate random variable $K = (K_{11}, \dots, K_{sr})$ in which

$$K_{ij} = \left\{ \begin{array}{l} \text{number of defaults in the } i\text{-th sector having the} \\ \text{j-th rating at the beginning of the 1-year period} \end{array} \right\}$$

taking in account that

$$N_{ij} = \left\{ \begin{array}{l} \text{number of obligors in the } i\text{-th sector having the} \\ \text{j-th rating at the beginning of the 1-year period} \end{array} \right\}$$

Note that $K_{ij} \leq N_{ij}$. We use the superscript to indicate the observation at time t (e.g., K_{ij}^t).

Example 2.1 — Historical data. Suppose that our portfolio has two sectors, S1 and S2, and four ratings, A, B, C, and D, in which D is the *defaulted* status. We collected the number of defaults by sector and rating from the past 20 years. We organized the collected data as follows:

Year	N (# obligors)						K (# defaults)					
	S1			S2			S1			S2		
	A	B	C	A	B	C	A	B	C	A	B	C
1	529	932	293	685	1249	206	3	12	15	3	25	10
2	522	977	278	638	1315	194	1	18	13	7	28	8
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
20	492	962	310	602	1227	183	6	28	21	4	31	16

Below, we provide an algorithm to simulate the annual number of defaults. The algorithm is a direct application of the multivariate default times distribution and the t-Student multi-factor model (see propositions 2.1 and 2.2). We will use it to generate samples with which to test the parameters estimation procedures.

Algorithm 2.1 — Annual number of defaults simulation. Let a multivariate default times distribution with annual probabilities of default p_1, \dots, p_r and a t-Student multi-factor copula $C_{\mathbf{v}, \vec{w}, R}^t$. Then we can simulate a sample of size T of the annual number of defaults by factor and rating doing the following:

1. Compute the Cholesky decomposition of $R = L \cdot L^\top$
2. For $t = 1, \dots, T$ repeat the following steps:
 - Simulate s independent $N(0, 1)$, \vec{y}
 - Simulate factors doing $\vec{z} = L \cdot \vec{y}$
 - Simulate $v \sim \chi_v^2$
 - For $i = 1, \dots, s$ and $j = 1, \dots, r$ repeat the following steps:
 - Simulate n_{ij}^t independent $N(0, 1)$, $\vec{\epsilon}$
 - Compute $\vec{x} = \sqrt{\frac{v}{v}}$ · $\left(w_i \cdot z_i + \sqrt{1 - w_i^2} \cdot \vec{\epsilon} \right)$
 - $k_{ij}^t = \{ \text{number of components of } \vec{x} \text{ less than } t_v^{-1}(p_j) \}$

Before proceeding to estimate the parameters, we derive the density of the number of observed defaults from the multivariate default times distribution. More information is available in [17, 10].

Proposition 2.3 — Density of the annual number of defaults (I). The density of the annual number of defaults, assuming that multivariate default times have marginals F_1, \dots, F_r and a t-Student copula $C_{\mathbf{v}, \hat{R}}^t$ with $\hat{R} = B_s(\vec{n}, \vec{1}, \Sigma)$, is

$$\begin{aligned}
K(k_{11}, \dots, k_{sr}; \mathbf{v}, \hat{R}) &= \Pr\{K_{11} = k_{11}, \dots, K_{sr} = k_{sr}; \mathbf{v}, \hat{R}\} = \\
&= \left(\prod_{i=1}^s \prod_{j=1}^r \binom{n_{ij}}{k_{ij}} \right) \cdot \underbrace{\int_{-\infty}^{t_v^{-1}(p_1)}}_{k_{11} \text{ times}} \underbrace{\int_{t_v^{-1}(p_1)}^{\infty}}_{n_{11}-k_{11} \text{ times}} \dots \underbrace{\int_{-\infty}^{t_v^{-1}(p_r)}}_{k_{sr} \text{ times}} \underbrace{\int_{t_v^{-1}(p_r)}^{\infty}}_{n_{sr}-k_{sr} \text{ times}} f(x_1, \dots, x_n) dx_1 \dots dx_n
\end{aligned}$$

in which s is the number of sectors, r is the number of ratings, n_{ij} and k_{ij} are respectively the number of obligors and defaults in the i -th sector with the j -th rating, $p_j = F_j(1)$ is the probability of default within 1 year of the j -th rating, $f(\vec{x})$ is the pdf of $t_n(\mathbf{v}, \vec{0}, \hat{R})$, and t_v^{-1} is the inverse of the cdf of the univariate t-Student. The combinatorial coefficients consider all combinations of exchangeable obligors.

We are not aware of an analytical expression for the density of the annual number of defaults. Furthermore, the numerical solution of the integral is difficult when the number of obligors is larger than just a few. Therefore, we attempt to solve the problem recurring in the multi-factor formulation.

Proposition 2.4 — Conditional density of the annual number of defaults. The conditional density of the annual number of defaults, assuming that multivariate default times have marginals F_1, \dots, F_r and a t-Student multi-factor copula $C_{\mathbf{v}, \bar{\mathbf{w}}, R}^t$, is

$$\begin{aligned} K(k_{11}, \dots, k_{sr} \mid \mathbf{v}, \bar{\mathbf{z}}; \mathbf{v}, \bar{\mathbf{w}}, R) &= \Pr\{K_{11} = k_{11}, \dots, K_{sr} = k_{sr} \mid \mathbf{v}, \bar{\mathbf{z}}; \mathbf{v}, \bar{\mathbf{w}}, R\} = \\ &= \prod_{i=1}^s \prod_{j=1}^r \binom{n_{ij}}{k_{ij}} \cdot H_{ij}(\mathbf{v}, z_i)^{k_{ij}} \cdot (1 - H_{ij}(\mathbf{v}, z_i))^{n_{ij} - k_{ij}} \end{aligned}$$

in which

$$H_{ij}(\mathbf{v}, z_i) = \Phi \left(\frac{\sqrt{\frac{\mathbf{v}}{\mathbf{v}}} \cdot t_{\mathbf{v}}^{-1}(p_j) - w_i \cdot z_i}{\sqrt{1 - w_i^2}} \right)$$

and s is the number of sectors, r is the number of ratings, n_{ij} and k_{ij} are respectively the number of obligors and defaults in the i -th sector with the j -th rating, $p_j = F_j(1)$ is the probability to default within 1 year of the j -th rating, Φ is the cdf of the univariate $N(0, 1)$, and $t_{\mathbf{v}}^{-1}$ is the inverse of the cdf of the univariate t-Student. The combinatorial coefficients consider all combinations of exchangeable obligors.

Proof. We use the pdf of the default times distribution (see proposition 2.1) to obtain the probability that an obligor k belonging to i -th sector with j -th rating defaults the first year conditioned to z_i and \mathbf{v} :

$$H_{ij}(\mathbf{v}, z_i) = \Pr\{T_k \leq 1 \mid \mathbf{v}, z_i\} = \Pr\{X_k \leq t_{\mathbf{v}}^{-1}(F_j(1)) \mid \mathbf{v}, z_i\} = \Pr\{X_k \leq t_{\mathbf{v}}^{-1}(p_j) \mid \mathbf{v}, z_i\}$$

in which T_k is the k -th component of the multivariate default times and X_k is the k -th component of the multivariate t-Student. Now we express X in the multi-factor form:

$$H_{ij}(\mathbf{v}, z_i) = \Pr \left\{ \sqrt{\frac{\mathbf{v}}{\mathbf{v}}} \cdot \left(w_i \cdot z_i + \sqrt{1 - w_i^2} \cdot \varepsilon_i^j \right) \leq t_{\mathbf{v}}^{-1}(p_j) \mid \mathbf{v}, z_i \right\}$$

Because \mathbf{v} and z_i are fixed and independent from $\varepsilon_i^j \sim N(0, 1)$, we obtain

$$H_{ij}(\mathbf{v}, z_i) = \Phi \left(\frac{\sqrt{\frac{\mathbf{v}}{\mathbf{v}}} \cdot t_{\mathbf{v}}^{-1}(p_j) - w_i \cdot z_i}{\sqrt{1 - w_i^2}} \right)$$

H_{ij} is the probability that an obligor defaults in 1 year. The probability of observing k_{ij} failures in a set of n_{ij} independent obligors follows a binomial distribution:

$$\text{Bin}(k_{ij}; n_{ij}, H_{ij}) = \binom{n_{ij}}{k_{ij}} \cdot H_{ij}^{k_{ij}} \cdot (1 - H_{ij})^{n_{ij} - k_{ij}}$$

The product for all sectors and ratings gives the stated formula. We can do the product because \mathbf{v} and z_i are fixed. □

Corollary 2.3 — Density of the annual number of defaults (II). We can express the density of the annual number of defaults using the conditional density:

$$\begin{aligned} K(k_{11}, \dots, k_{sr}; \nu, \vec{w}, R) &= \Pr\{K_{11} = k_{11}, \dots, K_{sr} = k_{sr}; \nu, \vec{w}, R\} = \\ &= \int_2^\infty \int_{\mathbb{R}^s} \psi(\nu) \cdot \phi(\vec{z}) \cdot K(k_{11}, \dots, k_{sr} | \nu, \vec{z}; \nu, \vec{w}, R) dz_1 \dots dz_s d\nu \end{aligned}$$

in which ψ is the pdf of χ_ν^2 and ϕ is the pdf of $N(\vec{0}, R)$.

This expression has a dimension significantly less than before. Despite this, this expression does not have a known analytical form and cannot be approximated numerically with accuracy when the number of factors is high (e.g., 10). The following sections provide effective methods for parameter estimation.

2.4 Frequentist Inference

We have failed to obtain reasonable estimates of the covariances using the nonparametric estimator described by Nagpal & Bahar [28], even in the Gaussian case. Tests consisted on simulating a panel data of defaults (see algorithm 2.1) and trying to infer the covariance matrix using the nonparametric estimator.

The cohort approach provides a parametric estimator [6, sec. 3.1] whose estimates are acceptable. As the nonparametric estimator, this estimator is based on estimates p_i (probability that an obligor defaults) and p_{ij} (probability that two obligors default simultaneously). We have adapted to the general case (t-Student) assuming the ν parameter is fixed and known.

Definition 2.2 — Implied covariance estimator. We assume that all obligors in a sector are exchangeable (identical rating per sector) and that we know the degrees of freedom of the t-Student multi-factor copula. We define the estimator of the factors' covariance, $\sigma_{ij} = w_i \cdot w_j \cdot R_{ij}$, as that value $\widetilde{\sigma}_{ij}$ that satisfies

$$p_{ij} = \int_{-\infty}^{t_v^{-1}(p_i)} \int_{-\infty}^{t_v^{-1}(p_j)} f(x, y; \nu, \widetilde{\sigma}_{ij}) dx dy \quad \text{where} \quad \begin{cases} p_i = \frac{\sum k_i^t}{\sum n_i^t} \\ p_j = \frac{\sum k_j^t}{\sum n_j^t} \\ p_{ii} = \frac{\sum k_i^t \cdot (k_i^t - 1)}{\sum n_i^t \cdot (n_i^t - 1)} & i = j \\ p_{ij} = \frac{\sum k_i^t \cdot k_j^t}{\sum n_i^t \cdot n_j^t} & i \neq j \end{cases}$$

in which n_i^t and k_i^t are respectively the number of obligors and defaults in the i -th sector and t -th year, $f(x, y; \nu, \widetilde{\sigma}_{ij})$ is the pdf of the bivariate t-Student, and t_v^{-1} is the inverse of the cdf of the univariate t-Student. Listing 2.1 implements this estimator in the R programming language.

Listing 2.1: Implied covariance (R script)

```
#install.packages("mvtnorm")
library(mvtnorm)

pri <- function(i, K, N)
```

```

{
  if (sum(dim(K) != dim(N))) stop("K and N dim differs")
  if (i <= 0 | ncol(K) < i) stop("i out-of-range")
  ret = sum(K[,i])/sum(N[,i])
  return(ret)
}

prij <- function(i, j, K, N)
{
  if (sum(dim(K) != dim(N))) stop("K and N dim differs")
  if (i <= 0 | ncol(K) < i) stop("i out-of-range")
  if (j <= 0 | ncol(K) < j) stop("j out-of-range")
  if (i != j) {
    ret = sum(K[,i]*K[,j])/sum(N[,i]*N[,j])
  }
  else {
    ret = sum(K[,i]*(K[,i]-1))/sum(N[,i]*(N[,i]-1))
  }
  return(ret)
}

f <- function(r, nu, p1, p2, p12)
{
  q1 = qt(p1, df=nu)
  q2 = qt(p2, df=nu)
  R = matrix(nrow=2, ncol=2, 1)
  R[1,2] = R[2,1] = r
  val = pmvt(lower=c(-Inf,-Inf), upper=c(q1,q2), corr=R, df=nu)
  return(p12-val)
}

implied_covariance <- function(nu, K, N)
{
  if (sum(dim(K) != dim(N))) stop("K and N dim differs")
  n = ncol(K)
  ret = matrix(nrow=n, ncol=n, 0)
  for(i in 1:n) {
    for(j in i:n) {
      p1 = pri(i, K, N)
      p2 = pri(j, K, N)
      p12 = prij(i, j, K, N)
      ret[i,j] = uniroot(f, nu, p1, p2, p12, interval=c(-0.9,0.9))$root
      ret[j,i] = ret[i,j]
    }
  }
  return(ret)
}

```

Example 2.2 — Implied covariance estimator. Consider the following scenario:

- Two sectors $\{S_1, S_2\}$
- Number of obligors by sector $\vec{n} = (1250, 750)$
- Default probability by sector $p_{S_1} = 2.5\%$ and $p_{S_2} = 7\%$
- Degrees of freedom $\nu = 15$
- Factor loadings $\vec{w} = (0.3, 0.25)$
- Factors correlation $R_{12} = 0.1$

We simulate the number of defaults of this scenario 1000 times using algorithm 2.1. Then, we compute the implied covariance estimator and display the results in the table below.

Exact	Estimated
$w_1^2 = 0.09$	$w_1^2 \approx \widetilde{\sigma}_{11} = 0.0882$
$w_2^2 = 0.0625$	$w_2^2 \approx \widetilde{\sigma}_{22} = 0.0662$
$w_1 \cdot w_2 \cdot R_{12} = 0.0075$	$w_1 \cdot w_2 \cdot R_{12} \approx \widetilde{\sigma}_{12} = 0.0072$

The drawbacks of the implied covariance estimator are that: 1) it does not consider parameter uncertainty, 2) it requires knowing in advance the value of ν , and 3) it only supports one rating per sector. This last point is crucial because it implies the absence of an economic cycle. For these reasons, one should not use this estimator. The rest of this chapter is devoted to developing a generic estimation procedure that avoids these problems.

2.5 Bayesian Inference

The sample correlation coefficient of 20 Gaussian values has a high uncertainty [22, pp. 217-221]. This stylized fact reflects our situation because we are trying to estimate factor correlations among other values. Uncertainty is exacerbated because our variables are latent and we are estimating from observations of type count-data. Thus, considering the estimated parameters as fixed values such as the Maximum Likelihood estimator is a naive and risky proposition. The Bayesian approach appears more appropriate [18, 34] because it considers the uncertainty of the parameters. Later we will see how to transfer this uncertainty to the assessment of credit risk.

We define a numerically feasible likelihood function using the conditional density of the number of defaults, thus implying that the parameters to be inferred are ν, \vec{w}, R and the latent variables $\nu^1, \vec{z}^1, \dots, \nu^T, \vec{z}^T$.

Definition 2.3 — Conditional likelihood. Assume a set of observations $\{K^t, N^t\}_{t=1, \dots, T}$ in which N^t and K^t are vectors representing the number of obligors and defaulted obligors in each sector-rating bucket. We define the conditional likelihood of these observations as

$$\begin{aligned} L(\nu, \vec{w}, R \mid \nu^1, \vec{z}^1, \dots, \nu^T, \vec{z}^T ; K^1, \dots, K^T) &= \\ &= \prod_{t=1}^T K(k_{11}^t, \dots, k_{sr}^t \mid \nu^t, \vec{z}^t ; \nu, \vec{w}, R) = \\ &= \prod_{t=1}^T \prod_{i=1}^s \prod_{j=1}^r \binom{n_{ij}^t}{k_{ij}^t} \cdot H_{ij}(\nu^t, z_i^t)^{k_{ij}^t} \cdot (1 - H_{ij}(\nu^t, z_i^t))^{n_{ij}^t - k_{ij}^t} \end{aligned}$$

We apply the Metropolis-Hastings algorithm to this conditional likelihood to infer the distribution of variables $\nu, \vec{w}, R, \nu^1, \vec{z}^1, \dots, \nu^T, \vec{z}^T$. Refer to appendix A.6 for a description of this algorithm adapted to our case.

Proposition 2.5 — Parameters estimation. We can estimate the parameters of the t-Student multi-factor copula from the observed annual number of defaults using the Metropolis-Hastings algorithm detailed in algorithm A.3, considering

- parameters $\theta \equiv \{v, \vec{w}, R, v^1, \vec{z}^1, \dots, v^T, \vec{z}^T\}$
- observations $y \equiv \{K^t, N^t\}_{t=1, \dots, T}$
- likelihood $f(y | \theta) \equiv L(v, \vec{w}, R | v^1, \vec{z}^1, \dots, v^T, \vec{z}^T; K^1, \dots, K^T)$ see definition 2.3
- proposal distributions $q_j(\theta_j | \theta_j^j) \equiv N(\theta, \sigma_j)$ all parameters
- prior distribution $v \equiv \text{Unif}(2, 1000)$
- prior distribution $w_i \equiv \text{Unif}(0, 1)$ $i = 1, \dots, s$
- prior distribution $r_{ij} \equiv \Xi_{ij}(\vec{z}^1, \dots, \vec{z}^T, R)$ $i = 1, \dots, s \quad j > i$
- prior distribution $\vec{z}^t \equiv N(\vec{0}, R)$ $t = 1, \dots, T$
- prior distribution $v^t \equiv \chi_v^2$ $t = 1, \dots, T$

The pdf of the distribution $\Xi_{ij}(\vec{z}^1, \dots, \vec{z}^T, R)$ is

$$\xi_{ij}(r_{ij}; \vec{z}^1, \dots, \vec{z}^T, R) = \frac{\prod_{t=1}^T \phi(\vec{z}^t; \vec{0}, R(r_{ij}, i, j))}{\int \prod_{t=1}^T \phi(\vec{z}^t; \vec{0}, R(x, i, j)) dx}$$

in which ϕ is the pdf of the multivariate $N(\vec{0}, R)$ distribution, and $R(x, i, j)$ is the matrix R with the cell (i, j) value replaced by x . We do not need to know the limits of the integral because this is constant and cancels into the M-H algorithm.

If you are interested in the Maximum Likelihood estimator, you can obtain an approximation of it taking the simulated parameter $\theta^{(0)}$ that maximizes $L(\theta^{(0)}) \cdot \prod_{t=1}^T \psi(\theta^{(0)}) \cdot \phi(\theta^{(0)})$ in which L is the conditional likelihood exposed in definition 2.3, ψ is the pdf of χ_v^2 , and ϕ is the pdf of $N(\vec{0}, R)$.

Example 2.3 — Model parameters estimation. Let us consider the following scenario:

- t-Student copula with $v = 20$.
- Three ratings {A, B, D} in which D is the *defaulted* status and three sectors {S1, S2, S3} having the following 1-year probabilities of default, factor loadings, and factor correlation matrix:

$$\begin{aligned} p_A &= 2\% \\ p_B &= 8\% \end{aligned} \quad , \quad \vec{w} = (0.4, 0.25, 0.3) \quad , \quad R = \begin{pmatrix} 1 & 0.5 & 0.3 \\ 0.5 & 1 & 0.4 \\ 0.3 & 0.4 & 1 \end{pmatrix}$$

- The number of obligors by sector-rating is

Obs	N (# obligors)					
	S1		S2		S3	
	A	B	A	B	A	B
1	250	750	500	500	750	250
2	250	750	500	500	750	250
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1000	250	750	500	500	750	250

1000 observations case. We simulate 1000 observations of the number of defaults by sector-rating using algorithm 2.1. Then we infer the parameters distribution from this sample using

Metropolis-Hastings algorithm A.3. The results are displayed in figure 2.1. Note that the variance of factor loadings is lower than the factor correlations.

20 observations case. To appreciate the effect of the number of observations in the final result, we repeated the estimation procedure but considered only the first 20 observations instead of the 1000 available. The results are displayed in figure 2.2. Note that the parameters uncertainty increases dramatically. The scenario depicted here is common; generally, we have just a few decades of observations. Later we will see how to transfer this parameters uncertainty to the assessment of portfolio risk.

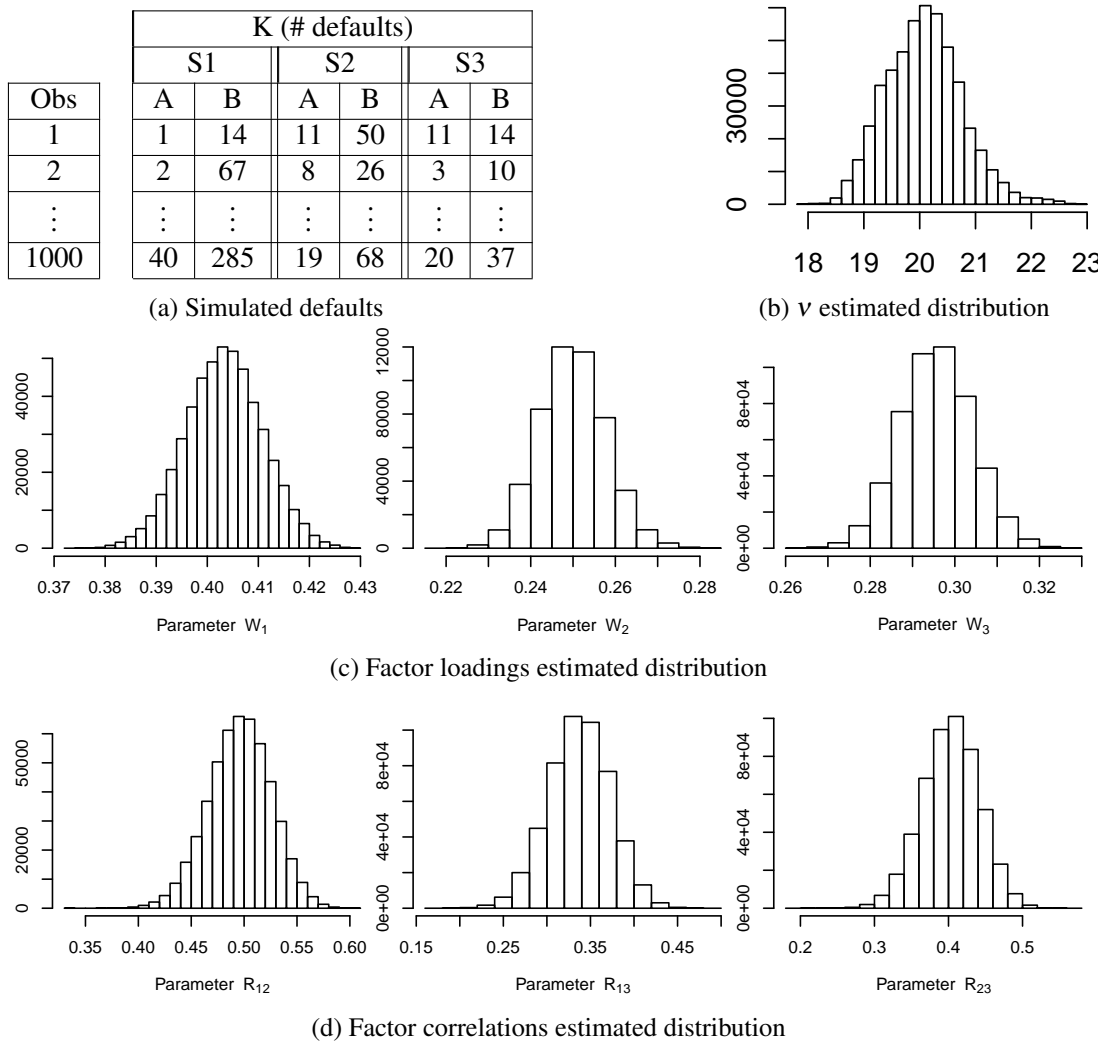


Figure 2.1: Model parameters estimation (1000 obs.)

Averaged PDs. We may be tempted to consider an averaged PD by sector when we lack data disaggregated by rating. The tests we have conducted indicate that in the Gaussian case ($\nu = \infty$), this could be an option because the estimated parameters using the averaged PDs approximately match the expected values. By contrast, when $\nu \neq \infty$, this does not work. In example 2.3, the

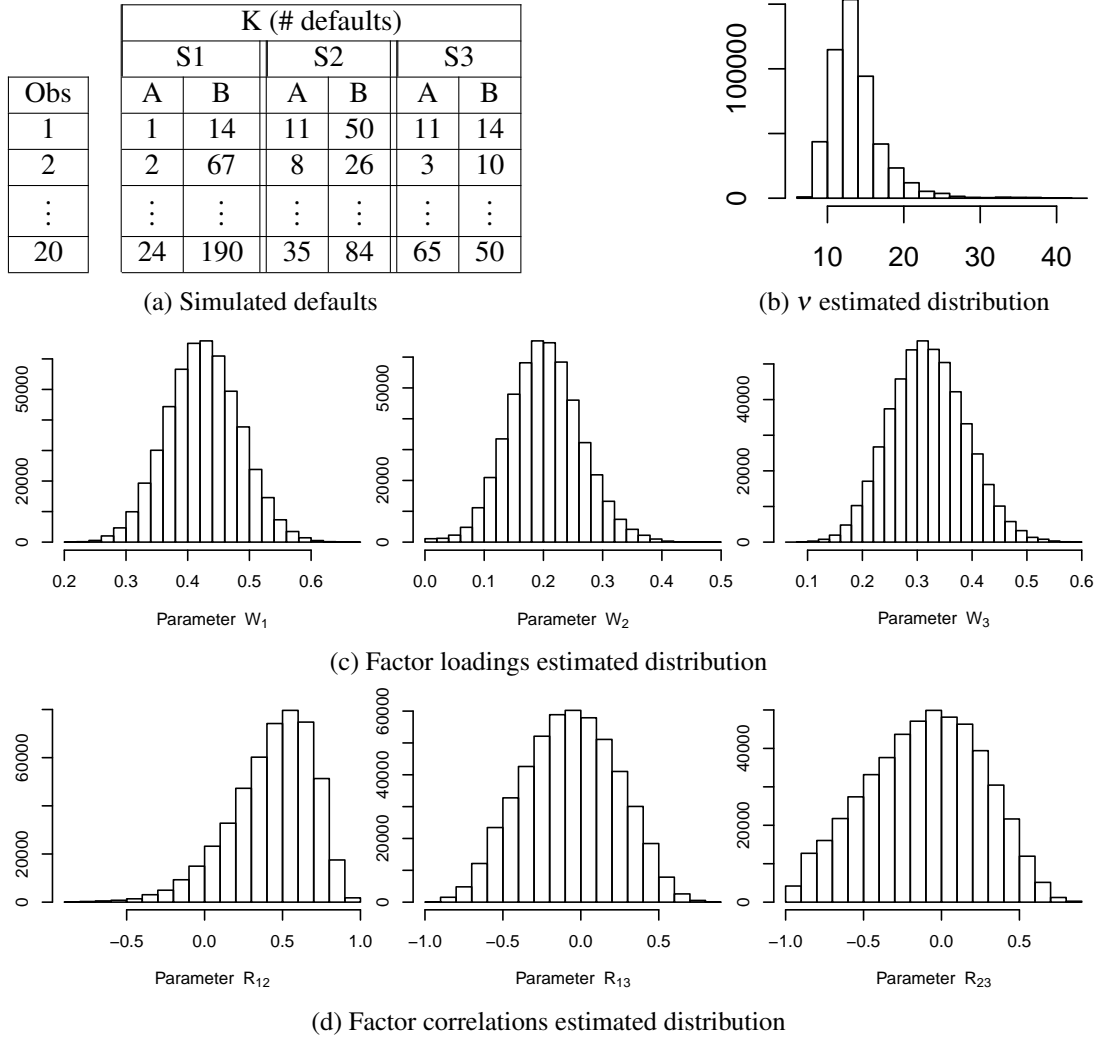


Figure 2.2: Model parameters estimation (20 obs.)

averaged 1-year PDs by sectors are

$$\begin{aligned}
 p_{S_1} &= \frac{250}{1000} \cdot p_A + \frac{750}{1000} \cdot p_B = 6.5\% & N_{S_1} &= 250 + 750 = 1000 \\
 p_{S_2} &= \frac{500}{1000} \cdot p_A + \frac{500}{1000} \cdot p_B = 5.0\% & N_{S_2} &= 500 + 500 = 1000 \\
 p_{S_3} &= \frac{750}{1000} \cdot p_A + \frac{250}{1000} \cdot p_B = 3.5\% & N_{S_3} &= 750 + 250 = 1000
 \end{aligned}$$

Constrained model. When we have a significant number of factors (e.g., 10) and a reduced set of observations (e.g., 20), then the uncertainty of the factor correlation matrix can be high. In these cases, one may consider a constrained model [10] that assumes that all factor correlations have the identical value or similar assumptions that reduce the number of correlations from which to estimate $s \times (s - 1) / 2$ to only a few.

Degrees of freedom. We observed that factor correlations and degrees of freedom are interwoven. When we fix one of them with a value more distinct than generated data and we estimate the other one, the latter reacts to compensate for the modification. This effect is more pronounced when the number of observations is reduced and the amount of information available is scarce, causing the estimations of v and R to have a high uncertainty. The Gaussian case ($v = \infty$) avoids this problem and reduces the number of parameters to estimate (we do not need to estimate v

and v^t in which $t = 1, \dots, T$). The risk analyst must evaluate the appropriateness of the Gaussian model considering the amount of information available, the number of factors, the renounce to capture the tail dependence, the Deviance Information Criterion (DIC) values, etc.

3 — Modeling Portfolio Loss

3.1 Portfolio Loss Distribution

Figure 3.1 displays the hierarchy of the portfolio components. As we observed in chapter 2, each obligor belongs to a single sector/factor and has a PD with the distribution F_i , depending on its rating. CCruncher considers that obligors' default times distribution has marginals F_i and a t-Student multi-factor copula with ν degrees of freedom, factor loadings \vec{w} , and factor correlation matrix R .

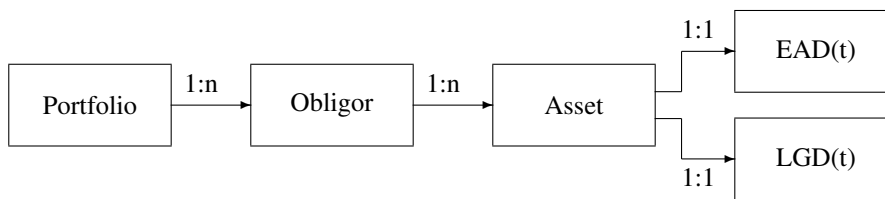


Figure 3.1: Hierarchy of the portfolio components

We define the EAD and LGD concepts used to model the credit risk emphasizing the temporal dimension. These concepts and the PD defined in section A.1 are those of the Basel II accord [30], the de-facto international financial risk regulation framework.

Definition 3.1 — Exposure at Default (EAD). The Exposure at Default of the j -th asset of the i -th obligor at time t indicates the total amount that the bank is exposed if the obligor defaults:

$$EAD_i^j(t) = \left\{ \begin{array}{l} \text{amount in risk of the } j\text{-th asset of the } i\text{-th obligor} \\ \text{if this defaults at time } t, \text{ measured in currency} \end{array} \right\}$$

EAD_i^j can be a function dependent of time when these amounts are fixed and known in advance (e.g., loan, bond) or can be a distribution defined over time when we have a model that supports it (e.g., line of credit).

Definition 3.2 — Loss Given Default (LGD). The Loss Given Default of the j -th asset of the i -th obligor at time t indicates the percentage of effective loss over the total exposure. This value depends on endorsements, guarantees, type of asset, default time, etc.

$$LGD_i^j(t) = \left\{ \begin{array}{l} \text{percentage of the } EAD_i^j(t) \text{ that is definitively} \\ \text{lost if the } i\text{-th obligor defaults at time } t \end{array} \right\}$$

LGD_i^j can be a function dependent of time or can be a distribution defined over time when we have a model that supports it.

Example 3.1 — 10-year Bond. Assume a 10-year, fixed-rate bond with an annual coupon of 4% issued by a company rated BB. The EAD for this asset can be obtained from its

expected cash flow because it is known in advance. We suppose that the bank's internal models indicate that in this case the LGD is a Beta(5, 2) distribution regardless of the default time. Figure 3.2 illustrates the defined concepts applied to this asset.

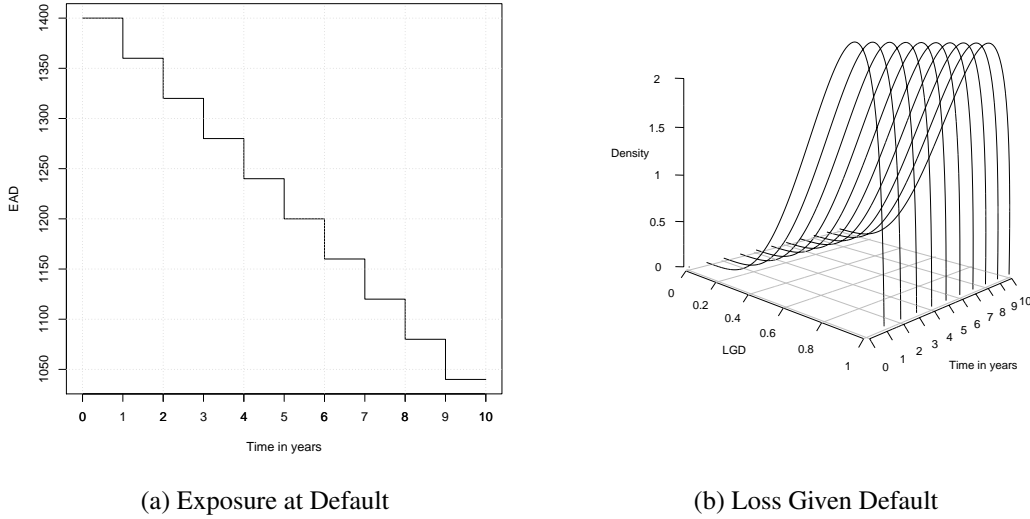


Figure 3.2: 10-year bond example

Definition 3.3 — Portfolio Loss at time T . The Portfolio Loss at time T is the accumulated losses in the time range $[0, T]$ caused by obligors that default (fail to make payments that they are obligated to make).

$$L_T = \{ \text{sum of portfolio losses in time range } [0, T] \}$$

Assume a credit portfolio in which the obligors' default times (T_1, \dots, T_n) is a multivariate distribution. In addition, we consider that each i -th obligor has m_i assets with known EAD and LGD. Thus, the portfolio loss distribution at time horizon t is

$$L_t = \sum_{i=1}^n \mathbb{1}_{[0,t]}(T_i) \cdot \left(\sum_{j=1}^{m_i} \text{EAD}_i^j(T_i) \cdot \text{LGD}_i^j(T_i) \right)$$

The portfolio loss distribution L_t does not have a closed form except in a few cases. One of those cases is the Large Homogeneous Pool (LHP). Vasicek [36] developed an analytical formula in the Gaussian case for the loss distribution of an infinitely large portfolio. Later, Schloegl and O'Kane [32] extended the formula to the t-Student case. The assumptions of these approximations are 1) a one-factor model, 2) an infinitely large portfolio, 3) equal PDs, and 4) equal exposure. These simplified models must be managed with care because they can cause problems [26] when the assumptions are not realized.

CCruncher assumes that the multivariate default times distribution has marginals F_i and a t-Student multi-factor copula with degrees of freedom ν , factor loadings \vec{w} , and factor correlation matrix R . CCruncher generates samples of the portfolio loss distribution simulating the obligors' default times. Portfolio loss distribution is approximated by the empirical distribution of the sample, and credit risk is measured using the appropriate statistics.

Algorithm 3.1 — Portfolio losses sampling. Let a portfolio be categorized in s sectors and r ratings. Assume a multivariate default times distribution with marginals F_i and a t-Student multi-factor copula $C_{v,\vec{w},R}^t$. We can then simulate a sample of size M of the portfolio loss distribution at time T by performing the following:

1. Compute the Cholesky decomposition of $R = L \cdot L^T$
2. For $m = 1, \dots, M$ repeat the following steps:
 - $\text{Loss}^m = 0$
 - Simulate s independent $N(0, 1)$, \vec{y}
 - Simulate factors doing $\vec{z} = L \cdot \vec{y}$
 - Simulate $v \sim \chi_v^2$
 - For each obligor $i = 1, \dots, n$ repeat the following steps:
 - Set $si = i$ -th obligor sector index
 - Set $ri = i$ -th obligor rating index
 - Simulate $\varepsilon \sim N(0, 1)$
 - Compute $x_i = \sqrt{\frac{v}{v}} \cdot \left(w_{si} \cdot z_{si} + \sqrt{1 - w_{si}^2} \cdot \varepsilon \right)$
 - Compute $t_i = F_{ri}^{-1}(t_v(x_i))$
 - if $t_i \leq T$, then for each asset j belonging to i -th obligor
 - * Evaluate or simulate $\text{EAD}_i^j(t_i)$
 - * Evaluate or simulate $\text{LGD}_i^j(t_i)$
 - * $\text{Loss}^m = \text{Loss}^m + \text{EAD}_i^j(t_i) \cdot \text{LGD}_i^j(t_i)$
 - print Loss^m

Example 3.2 — Large Homogeneous Pool. Assume a portfolio composed of 1000 exchangeable obligors (with identical ratings and sectors) in which every obligor owes 1€ to 1 year, the 1-year default probability is $p = 10\%$, and the factor loading is 20%. This problem is close to fulfilling the LHP assumptions. In the Gaussian case, the density is [3, chap. 2.5]:

$$f_{p,w}(x) = \frac{\sqrt{1-w^2}}{w} \cdot \exp \left(-\frac{(N^{-1}(x))^2}{2} - \frac{(N^{-1}(p) - \sqrt{1-w^2} \cdot N^{-1}(x))^2}{2 \cdot w^2} \right)$$

in which w is the factor loading, p is the probability to a 1-year horizon, and $x \in (0, 1)$ represents the percentage of loss in the portfolio.

File `$CCRUNCHER/samples/test04.xml` contains the CCruncher input file of this example. Figure 3.3 displays the theoretical distribution provided by the LHP approach and the distribution approximated by CCruncher, which, as expected, are quite similar. If we reduce the number of obligors from 1000 to 100, then the LHP assumptions are broken and the distribution provided by the LHP approach fails.

3.2 Credit Risk Measurement

After N simulations (e.g., 20,000, 500,000 or more) of the portfolio loss, we measure the portfolio credit risk computing the corresponding statistics and their standard errors according to

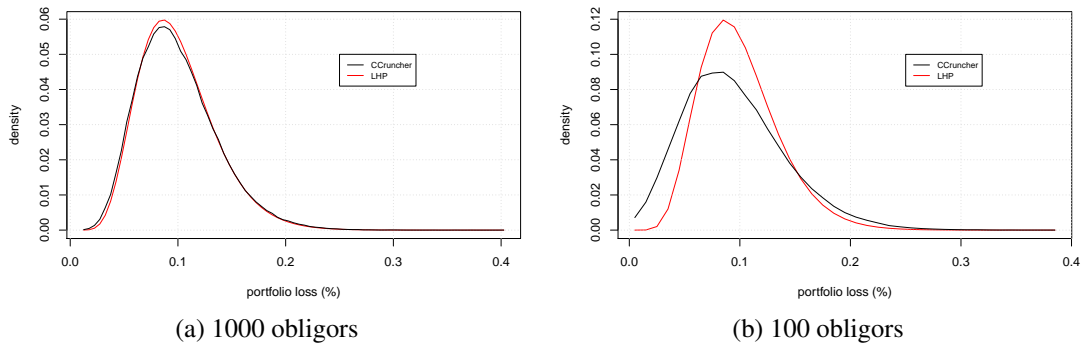


Figure 3.3: Large Homogeneous Pool example

sample size.

3.2.1 Portfolio risk

In this section we provide the definitions and formulas of the most usual credit risk measures. Figure 3.4 illustrates the concepts presented in this section. Any other user-defined risk measure can be estimated from the simulated sample, and its standard error can be obtained using resampling methods.

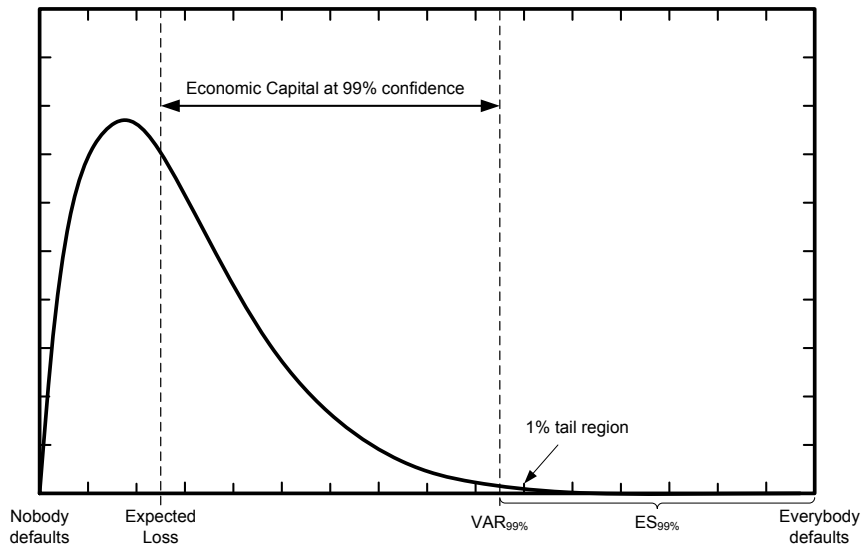


Figure 3.4: Portfolio Loss Distribution at time T

Expected Loss

The Expected Loss attempts to answer the question, “What is my expected loss for the next year?”

Definition 3.4 — Expected Loss (EL). The Expected Loss of a portfolio at time horizon T

is the mean of the portfolio loss distribution.

$$EL(L_T) = E(L_T)$$

Expected Loss can be approximated using the following statistic:

$$EL = \hat{\mu} \pm \Phi^{-1}\left(\frac{1-\alpha}{2}\right) \cdot \frac{\hat{\sigma}}{\sqrt{N}} \quad \text{being} \quad \begin{cases} \hat{\mu} = \frac{1}{N} \sum_{i=1}^N x_i \\ \hat{\sigma} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \hat{\mu})^2} \end{cases}$$

in which x_i are the simulated portfolio losses, α is the error confidence level, Φ^{-1} is the $N(0, 1)$ inverse cumulative distribution function, and $\hat{\mu}$ and $\hat{\sigma}$ are the mean and standard deviation estimators.

Value at Risk

The Value at Risk [21] is the most commonly used risk measure to respond to the question, ‘‘How much could I lose in a really bad year?’’

Definition 3.5 — Value at Risk (VaR). The Value at Risk of a portfolio at time horizon T and confidence level β is the β -quantile of the portfolio loss distribution at time T . Consider

$$\text{VaR}_\beta(L_T) = \inf\{x \in \mathbb{R} \mid F(x) \geq \beta\}$$

in which $F(x) = \Pr\{L_T \leq x\}$ is the portfolio loss distribution at time T .

Value at Risk can be approximated using the following statistic:

$$\text{VaR}_\beta = \widehat{q}_\beta \pm \Phi^{-1}\left(\frac{1-\alpha}{2}\right) \cdot \text{stderr}(q_\beta) \quad \text{being} \quad \begin{cases} \frac{k}{N} \leq \beta < \frac{k+1}{N} \\ \widehat{q}_\beta = x_{k:N} \end{cases}$$

$$\text{stderr}(q_\beta) = \sqrt{C_2 - C_1^2} \quad \text{being} \quad \begin{cases} M = [N \cdot \beta + 0.5] \\ a = M - 1 \\ b = N - M \\ W_i = B(a, b, \frac{i+1}{N}) - B(a, b, \frac{i}{N}) \\ C_k = \sum_{i=1}^N W_i \cdot x_i^k \end{cases}$$

in which x_i are the simulated portfolio losses, $x_{k:N}$ is the k -th element of ascendant-sorted values, α is the error confidence level, Φ^{-1} is the $N(0, 1)$ inverse cumulative distribution function, \widehat{q}_β is the quantile estimator, $\text{stderr}(q_\beta)$ is the estimation of the standard error using the Maritz-Jarrett method described in [37, chap. 3.5.3], $[x]$ is the integer part of x , and $B(a, b, x)$ is the regularized incomplete beta function. When the sample size is high, the incomplete beta function can be difficult to ascertain due to convergence problems. Also, a large sample size causes the evaluation of the incomplete beta function multiple being this computation numerically costly.

Expected Shortfall

VaR is not a coherent risk measure because it does not fulfill the sub-additive property [20]. Expected Shortfall is a coherent risk measure similar to VaR [1].

Definition 3.6 — Expected Shortfall (ES). The Expected Shortfall (aka Conditional VaR, aka Conditional Tail Expectation) of a portfolio at time horizon T and confidence level β is the average of the worst $\beta\%$ portfolio losses at time T ,

$$ES_{\beta}(L_T) = E(L_T \mid L_T \geq \text{VaR}_{\beta}(L_T))$$

Expected Shortfall can be approximated using the following statistic:

$$ES_{\beta} = \hat{\mu} \pm \Phi^{-1}\left(\frac{1-\alpha}{2}\right) \cdot \frac{\hat{\sigma}}{\sqrt{K}} \quad \text{being} \quad \begin{cases} \hat{\mu} = \frac{1}{K} \sum_{i=1}^K y_i \\ \hat{\sigma} = \sqrt{\frac{1}{K-1} \sum_{i=1}^K (y_i - \hat{\mu})^2} \end{cases}$$

in which $\{y_1, \dots, y_K\} = \{x_i \mid x_i \geq \text{VaR}_{\beta}\}_{i=1, \dots, N}$ are the simulated portfolio losses larger than VaR_{β} , α is the error confidence level, Φ^{-1} is the $N(0, 1)$ inverse cumulative distribution function, and $\hat{\mu}$ and $\hat{\sigma}$ are the mean and standard deviation estimators. This estimator assumes that the estimated VaR_{β} has the exact value. Despite this, if the sample size is high enough, then the estimated value is good enough.

Example 3.3 — test05.xml. Assume a portfolio composed of 1000 obligors in which every obligor owes 1€ to 1 year. Then $EAD_i^1(t) = \mathbb{1}_{[0,1]}(t)$, with a $LGD_i^1(t) = 100\%$. There are 4 ratings {A, B, C, D}; D is the *defaulted* status, and the remaining ratings have the following 1-year default probability: $p_A = 5\%$, $p_B = 10\%$, $p_C = 20\%$. Factor loadings are $\vec{w} = (0.4, 0.35)$, and the factors correlation is $\text{Cor}(S_1, S_2) = 0.25$. The t-Student copula has $\nu = 15$ degrees of freedom, and the number of obligors by sector and rating are

S1			S2		
A	B	C	A	B	C
167	166	167	167	167	166

File \$CCRUNCHER/samples/test05.xml contains the CCRuncher input file of this example. Figure 3.5 displays the portfolio loss density and risk measures obtained using 200,000 values simulated by CCRuncher.

It is difficult to ascertain the validity of this example. Some authors [15] obtained quasi-analytical expressions for the Gaussian two-factor model with a common PD; however, those expressions are not applicable to this example (t-Student, multiple PDs). One manner in which to check that the result is correct involves disaggregating the simulated portfolio losses by sector-rating (see section 3.2.2) and estimating the parameters ν , \vec{w} , and R with the first 1000 simulations using the procedure described in section 2.5. If the estimated parameters match the parameters of the example, we have consistent simulation and estimation algorithms. Observe that in this case, the simulated losses can be equated to the number of defaults because each defaulted obligor is computed as a 1€ loss.

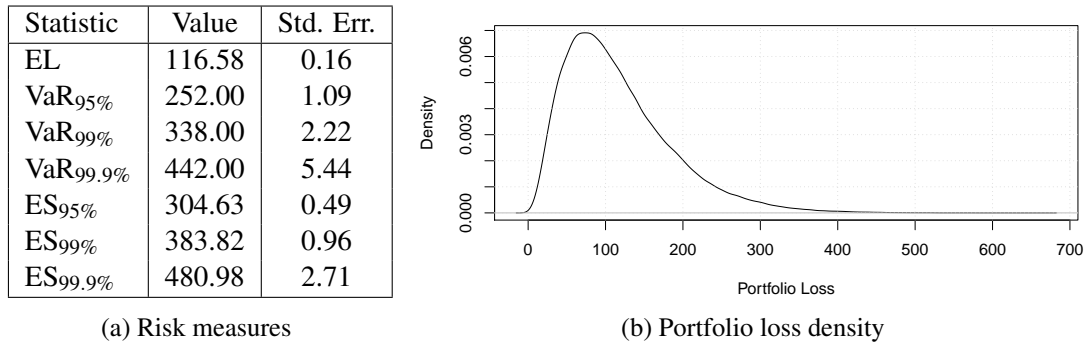


Figure 3.5: 2-factors example

3.2.2 Risk Disaggregation

Once we have measured the credit risk, naturally arises the question “how much of this risk corresponds to the business unit X?” CCruncher allows defining and simulating sub-portfolio losses in order to respond this question.

CCruncher simulates the entire portfolio loss at time T . It also allows (in the same execution) the simulation of sub-portfolios losses. In CCruncher, a sub-portfolio is called a *segment*. The disjoint union of segments covering the entire portfolio is called a *segmentation*, i.e., a segmentation is composed by non-overlapped segments that constitute the entire portfolio. For example, we can define the segmentation of geographic areas where the bank operates composed by four segments {N, S, E, W}. When an obligor defaults, the loss of its assets is added into the entire portfolio loss and into the segment loss where the obligor resides. CCruncher allows defining and managing multiple segmentations simultaneously. The segmentations can be composed of obligors or assets. In the first case, the obligor loss is aggregated to segment loss when the obligor defaults. In the second case, the asset losses are aggregated to segment losses when the obligor defaults.

Expected Loss

Expected Loss is additive respect the portfolio composition, i.e., if L , L_1 , and L_2 are respectively the portfolio, sub-portfolio 1, and sub-portfolio 2 loss distributions in which $L = L_1 + L_2$, then $EL(L_1) + EL(L_2) = EL(L)$. The EL attributable to the k -th segment, in cash, is given by the following statistic:

$$EL_k = \frac{1}{N} \sum_{i=1}^N x_i^k$$

in which x_i^k are the simulated losses of the k -th sub-portfolio. Dividing EL_k by the portfolio EL gives the percentage of EL consumed by this sub-portfolio.

Value at Risk

Value at Risk is not additive, it is not even sub-additive [20]. As mentioned in [3, chap. 5.2.2], calculating VaR risk measure contributions is a natural but difficult attempt. In the CCruncher case, problems arise when we try to estimate $E(L_k | L = VaR_\beta(L))$ because the subsample of simulated sub-portfolio losses having a portfolio loss equal to VaR_β has a very small size. For this reason, CCruncher don't provides support for VaR disaggregation.

Expected Shortfall

Expected Shortfall can be disaggregated using the simulated losses of the portfolio and its sub-portfolios. As indicated in [3, chap. 5.2.3], the contribution of k-th sub-portfolio to ES is given by $E(L_k | L \geq \text{VaR}_\beta(L))$. The ES attributable to the k-th segment, in cash, is given by the following statistic:

$$\text{ES}_k = \frac{1}{K} \sum_{i=1}^K y_i^k$$

in which x_i are the simulated portfolio losses, x_i^k are the simulated losses of the k-th sub-portfolio, and $\{y_1^k, \dots, y_K^k\} = \{x_i^k | x_i \geq \text{VaR}_\beta\}_{i=1 \dots N}$ are the simulated sub-portfolio losses having a portfolio loss large than VaR_β . Dividing ES_k by the portfolio ES gives the percentage of ES consumed by this sub-portfolio.

Example 3.4 — Risk disaggregation. We use the identical scenario as that described in example 3.3 and that corresponds to CCruncher input file `test05.xml`. This example defines a segmentation named *sector-rating* with segments S1-A, S1-B, S1-C, S2-A, S2-B, S2-C. Segment X-Y is composed by the obligors located at sector X with the rating Y; each segment has ≈ 166 obligors. Figure 3.6 displays the risk disaggregated respect this segmentation.

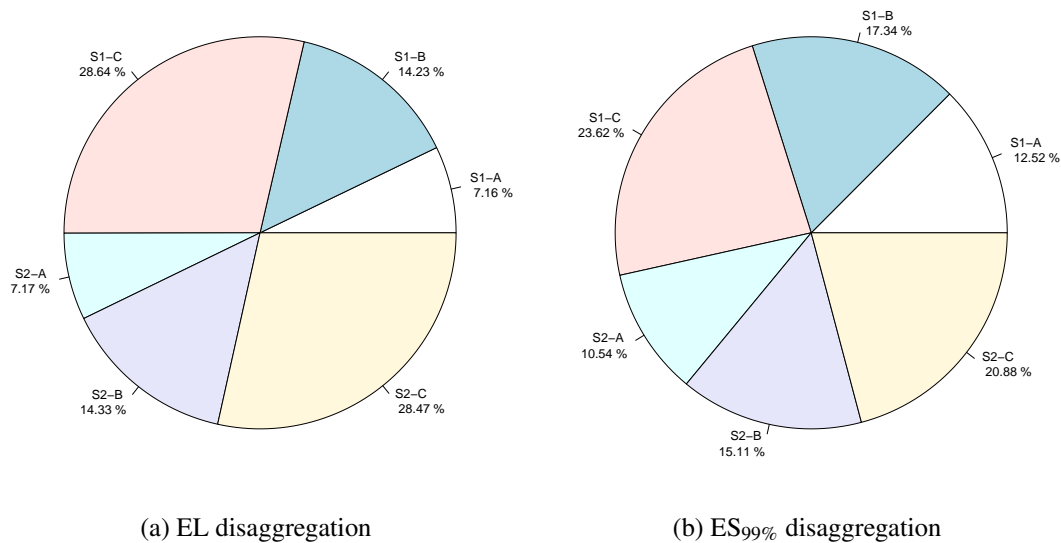


Figure 3.6: Risk disaggregation

3.3 Simulation Details

Variance reduction. CCruncher generates random samples from the portfolio loss distribution. This sample is used to estimate the risk measures (e.g., VaR) using the statistics described in section 3.2 or any other required by the analyst. Any one of these measures, considered individually, can be formulated as a Monte Carlo problem and can apply some variance reduction techniques (e.g., importance sampling [5]), but not simultaneously. For this reason, the variance reduction techniques used in CCruncher do not seek an increase in precision, but an improvement in the overall results in some manner. The applied variance reduction procedures implemented

in CCruncher are antithetic variates and Latin hypercube sampling. You can find more info at appendix A.7.

Antithetic variates. In CCruncher the random sampling of the portfolio loss distribution is performed using $g(X)$ in which X is a multivariate t-Student distribution and $g(x) = F^{-1}(t_v(x))$. We use the fact that the multivariate t-Student distribution X is symmetrical, that is, (x_1, \dots, x_n) is equiprobable to $(-x_1, \dots, -x_n)$, to apply the antithetic method. Expected Loss is the unique statistic that benefits from the variance reduction. However, rather than reducing the variance, we are interested in the reusing of the simulated random numbers to increase the simulation speed. When we use the antithetic method to sample the default times and there are some stochastic EADs or LGDs, we use distinct EADs' and LGDs' random values for each variate.

Latin hypercube sampling. In CCruncher, the random variables that can be sampled using the LHS technique are the degrees of freedom v , and the factors \vec{z} . The aim is to ensure a uniform scanning of these variables so that the extreme cases are represented.

EAD and LGD interpolation. Given an asset, CCruncher knows its EAD and LGD values at fixed times $\{t_1, \dots, t_n\}$ provided by the user. When we evaluate these functions in a time value distinct from those provided, we use the following piecewise constant interpolation:

$$\left. \begin{array}{l} \text{EAD}(t) = \text{EAD}(t_k) \\ \text{LGD}(t) = \text{LGD}(t_k) \end{array} \right\} \quad \text{where } t_{k-1} < t \leq t_k$$

PD functions approximation. CCruncher allows defining the PD functions using the transition matrix or defining explicitly the PD values at some fixed times. When PDs are defined using the transition matrix, CCruncher uses proposition A.2 to derive F_i for each month, and intermediate values are interpolated using cubic splines. When PDs are defined explicitly at some fixed times, then intermediate values are interpolated using cubic splines; however, if the resulting function is not strictly increasing, then CCruncher uses linear splines.

Function composition. To speed up the evaluation of the function $t = F_r^{-1}(t_v(x))$ involved in the simulation of obligors' default times, we interpolate the function using linear splines on a selected set of points $\mathbb{S} = \{(x_1, d_1), \dots, (x_k, d_k)\}$. Let $\{t_1, \dots, t_m\}$ represent the days in which PDs have known values (not interpolated), and let $t = P(x)$ represent the function $t = F_r^{-1}(t_v(x))$ linearly interpolated using the current points of \mathbb{S} . CCruncher determines the set \mathbb{S} using the following procedure:

- Add the first day of the simulation time range to \mathbb{S} , d_1 .
- Add the last day of the simulation time range to \mathbb{S} , d_T .
- Repeat until $\max_{d_1 \leq d \leq d_T} |d - P(t_v^{-1}(F_r(d)))| < 1$ hour
 - Let d be the day on which maximum error is achieved
 - Let $t_i \leq d \leq t_{i+1}$ be the nearest days on which PDs are user defined
 - If t_i is not contained in \mathbb{S} , add t_i to \mathbb{S}
 - Else if t_{i+1} is not contained in \mathbb{S} , add t_{i+1} to \mathbb{S}
 - Else add d to \mathbb{S}

Interest rate. CCruncher allows considering the Net Present Value of the simulated losses. This is computed using an interest rate curve defined at fixed times and interpolated using linear or cubic splines (selected by user). Allowable methods are simple interest, compound interest and continuous interest. The formulas used to compute the present value are as follows:

$$\begin{array}{lll} V_0 = \frac{V_t}{1+r \cdot \Delta t} & V_0 = \frac{V_t}{(1+r)^{\Delta t}} & V_0 = \frac{V_t}{\exp(r \cdot \Delta t)} \\ \text{(Simple)} & \text{(Compound)} & \text{(Continuous)} \end{array}$$

in which V_t is the value at time t and r is the interest rate given by the interest curve at time t . Don't consider this option if your EADs are the sum of cash flow on distinct dates. In this case, we recommend filling the input file with the EADs' present value and setting to 0 the interest rate curve.

Parallel computing. CCruncher parallelizes the portfolio loss sampling, spawning multiple simulation threads, each with its own Mersenne twister Random Number Generator seeded with consecutive values. When the number of obligors is high, data do not fit in the cache memory L1 and L2. The memory transfer is a time-consuming operation that must be minimized. To achieve this, each thread performs simultaneously *blocksize* simulations. Each obligor is simulated *blocksize* times (each of them with the corresponding v and \bar{z}) before moving to the next obligor.

Parameterization. The CCruncher input file contains the data required to simulate the portfolio losses (obligors, assets, EADs, LGDs, ratings, factor correlations, factor loadings, copula to use, interest rates, segmentations, etc.). This file can be quite large (e.g., 1GB). When we evaluate the risk sensitivity or the effect of parameters' uncertainty, we must slightly modify the input file and reevaluate the portfolio loss distribution multiple times. To simplify this task, CCruncher provides the macro mechanism, that allows using variables that are replaced by the user-defined values anywhere they are located in the input file, and the automatic evaluation of numerical expressions. Combining these two features, macros and the evaluation of numerical expressions, provides a flexible manner in which to parameterize one's input file (e.g., 2000*\$EUR2USD).

4 — Additional Topics

4.1 Risk Sensitivity

In this section we make a basic sensitivity analysis evaluating the variation of a parameter s , in the risk measure V . The goal is to determine the appearance of the curve $V(s)$ (increasing/decreasing, concave/convex, linear/quadratic/logarithmic, asymptotic behavior) and an estimate of the slope given by $\frac{\partial V}{\partial s}$.

It is difficult to draw general guidelines in the absence of a closed formula for the portfolio credit risk that takes into account all the parameters: number of ratings, PDs over time, number of sectors, number of obligors, EADs and LGDs for each asset, factor loadings, factor correlations, and v . For this reason, we lay out this section analyzing the case `test05.xml` exposed in example 3.3. In this case, we consider that risk measure is $ES_{99\%}$ and parameters subject to sensibility analysis are: n, \vec{w}, R, v .

We sketch the function $ES_{99\%}(s)$ by calculating $ES_{99\%}$ in various consecutive values of the analyzed parameter. Each of these values is obtained by applying the simulation procedure described in Chapter 4. In order to reduce variations caused by the simulation procedure, we use the same RNG seed in each evaluation.

Example 4.1 — Sensitivity to portfolio size. Common sense tells us that a large and homogeneous portfolio must have the same risk (as percentage of the total exposure) that a portfolio with the same characteristics but with twice obligors. Figure 4.1 shows the $ES_{99\%}$ by varying the number of obligors that compose the portfolio described in `test05.xml`. Note that this value is almost equal in range $[200, \infty)$. In the range $[50, 200]$ we can appreciate the impact of the small number of individuals in the risk measure, not very exaggerated otherwise.

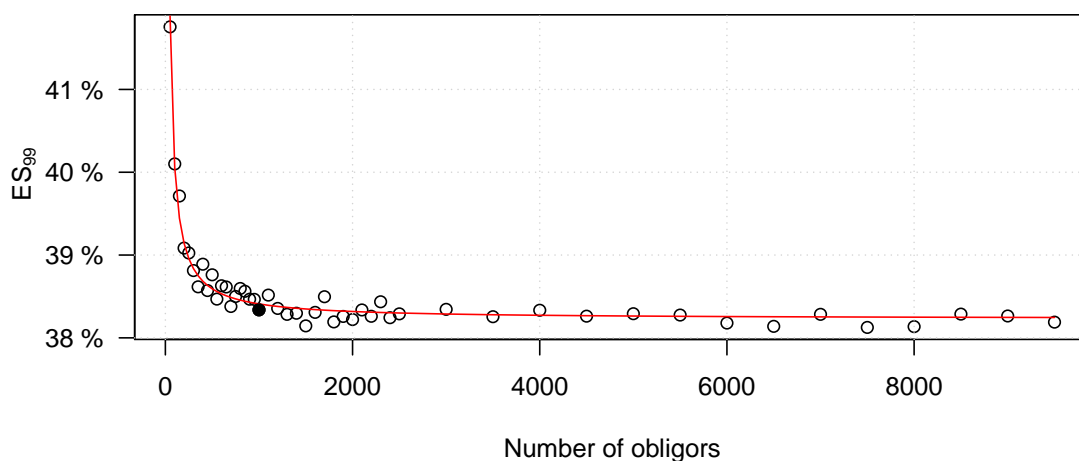


Figure 4.1: $ES_{99\%}$ vs. portfolio size

Example 4.2 — Sensitivity to copula parameters. Figure 4.2 shows the portfolio risk value $ES_{99\%}$ depending on the copula parameters. Each dot is a risk value computed using a simulation, the black dot corresponds to the original value in example 3.3, and the confidence bounds shows the impact of the parameter value in the error attributable to simulation (this value is magnified in order to appreciate the impact).

If we consider only the range of feasible values (eg. $w_i \in (0, 0.5)$, or $v \in (5, \infty)$), we note that the impact of factor loading loadings and v are similar. However, the impact of factor correlations is much smaller. This is good news, because as we saw in the estimation of the parameters, the uncertainty of factor correlations is much higher than that of factor loadings. We also recall that in the estimation procedure, the values of factor loadings and v are interlaced.

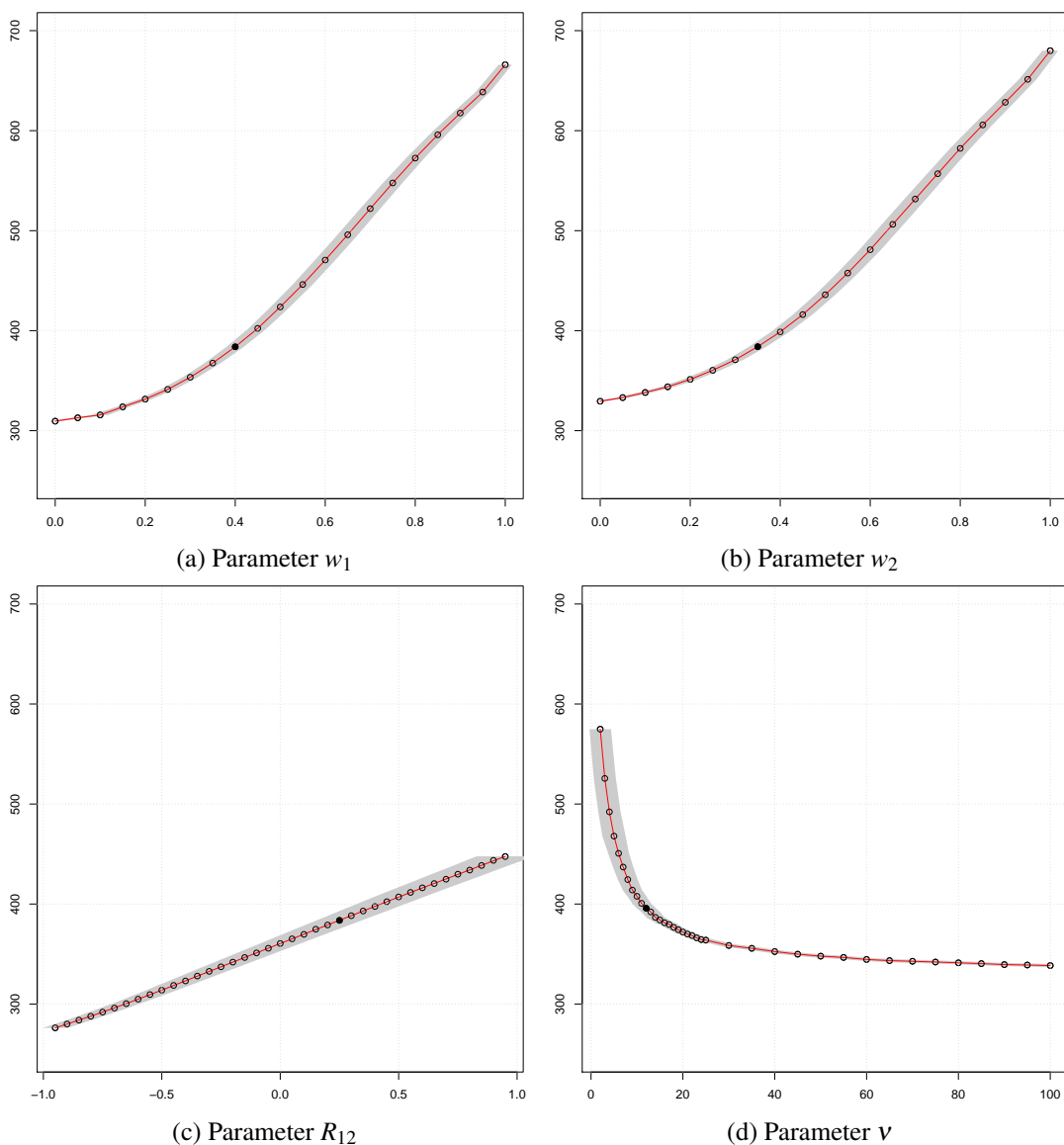


Figure 4.2: $ES_{99\%}$ sensitivity to copula parameters

The main drawback of sensitivity analysis is that it measures the impact of the uncertainty of a parameter fixing the other ones. Another drawback is that it does not consider the distribution of

the parameters. The section that follows provides a solution to these problems.

4.2 Parameters Uncertainty

The estimation of dependence parameters (degrees of freedom ν , factor loadings \vec{w} , and factor correlation matrix R) has an uncertainty that cannot be ignored [34, 18]. Example 2.3 illustrates this phenomenon and its magnitude (see the *20 observations* case).

The CCruncher framework encourages incorporating the parameter uncertainty into the credit risk valuation. This indicates that credit risk measures (e.g., $\text{VaR}_{99\%}$, $\text{ES}_{99\%}$) are regarded as distributions instead of values. For example, in the case of the Value at Risk, we consider the function $\text{VaR}_{99\%}(L_T(\nu, \vec{w}, R))$ that depends on the multivariate random variables ν, \vec{w} and R . Note that Expected Loss is not affected by the uncertainty of dependence parameters because the expectation satisfies $E(X + Y) = E(X) + E(Y)$ even if X is not statistically independent of Y .

The solution recommended by CCruncher to determine the credit risk measure distribution is the obvious one and is based on the identical procedure used to determine the distribution of the portfolio loss or the distribution of the parameters: the random sampling of the risk measure distribution.

On one hand, we have a procedure that generates random samples from the multivariate parameters distribution (see proposition 2.5). These samples can be filtered to obtain a sample of independent parameters values.

On the other hand, we are able to measure the credit risk of a portfolio considering that dependence parameters are fixed values. This is performed sampling the portfolio loss (see algorithm 3.1) and computing the credit risk measure using the appropriate statistic (see section 3.2).

We combine both components to define an algorithm that samples the desired credit risk measure. The empirical density given by this sample approximates the risk measure distribution and allows evaluating the effect of the parameter uncertainty. One must be cautious because this procedure unveils the copula parameters uncertainty, but other uncertainties remain hidden such as: 1) copula selection, 2) factors/sectors selection and assignment, 3) PDs uncertainty, and 4) EAD and LGD models accuracy.

Algorithm 4.1 — Credit risk measure distribution. Assume a portfolio (ratings, sectors, PDs, EADs, LGDs) and the number of defaults observed by sector-rating in the previous years. We can determine the distribution of a credit risk measure (e.g., $\text{VaR}_{99\%}$) by performing the following:

1. Obtain an ergodic sample of the parameters applying the Metropolis-Hastings algorithm (see proposition 2.5).
2. Remove the first iterations (burn-in period), and determine the minimum common length B at which the ACFs are 0.
3. Let $\{\nu^t, \vec{z}^t, R^t\}_{t=1, \dots, N}$ be the subsample of the parameters distribution composed by the indexes multiples of B .
4. For each simulated parameter, simulate the portfolio loss distribution $L_T(\nu^t, \vec{w}^t, R^t)$ using the algorithm 3.1.
5. For each simulated portfolio loss distribution, compute its risk measure using the statistic described in section 3.2: $\{\text{VaR}_{99\%}(L_T(\nu^t, \vec{w}^t, R^t))\}_{t=1, \dots, N}$.

Example 4.3 — Credit risk considering parameters uncertainty. We use the identical scenario as that described in example 3.3 and that corresponds to CCruncher input file `test05.xml`. We generate 20 observations of annual numbers of defaults using the algorithm 2.1 (in the real world, these observations are extracted from the historical records). In the following lines, we apply algorithm 4.1 to estimate the $ES_{99\%}$ distribution, considering the parameters' uncertainty.

The first step includes the generation of a sample of the parameters' distribution using the Metropolis-Hastings algorithm. We performed 1,000,000 iterations. Figure 4.3 displays the marginals of this distribution.

The second step is removing the first 20,000 iterations (burn-in period) and determining the thin interval using the ACFs. Figure 4.4 shows the ACF for each parameter. A thin interval of 3000 is sufficient in this example.

The third step is pruning the parameters sample preserving only the simulated values whose index is a multiple of 3000. Below is displayed part of the simulated values:

N	v	w_1	w_2	R_{12}
1	12.54	0.29	0.48	0.21
2	10.77	0.28	0.32	-0.28
\vdots	\vdots	\vdots	\vdots	\vdots
300	10.06	0.33	0.36	0.02

The fourth step is the simulation of the portfolio loss distribution for each simulated parameter. This is quite an intensive computation. The CCruncher's command line execution and the input file macro expansion can assist in performing this step. Listing 4.1 displays a simple batch script to sequentially execute the required simulations. Those who have a cluster, a cloud or a grid may consider to executing multiple parallel tasks using a framework such as MakeFlow¹ or something similar.

Finally, the fifth step is the computation of the risk statistic for each portfolio loss distribution. The density of the $ES_{99\%}$ can be approximated using the empirical density. Listing 4.2 provides an R script to perform this task.

The exact value of the $ES_{99\%}$ assuming that parameters are fixed and known is ≈ 383 € (see example 3.3). If we do not assume that parameters are known and we infer the parameters from 20 observations of the numbers of defaults, then we obtain the distribution displayed in figure 4.5. The key point is to dispose a distribution indicating the feasible values that can take the risk measure $ES_{99\%}$.

Listing 4.1: Execution of multiple CCrunchers (bash script)

```
mkdir data/MH001;
bin/ccruncher -cmd -o data/MH001 -DNU=12.54 -DW1=0.29 \
  -DW2=0.48 -DR12=0.21 samples/test05.xml > data/MH001/ccruncher.out;

mkdir data/MH002;
bin/ccruncher -cmd -o data/MH002 -DNU=10.77 -DW1=0.28 \
```

¹<http://www3.nd.edu/~ccl/software/makeflow/>

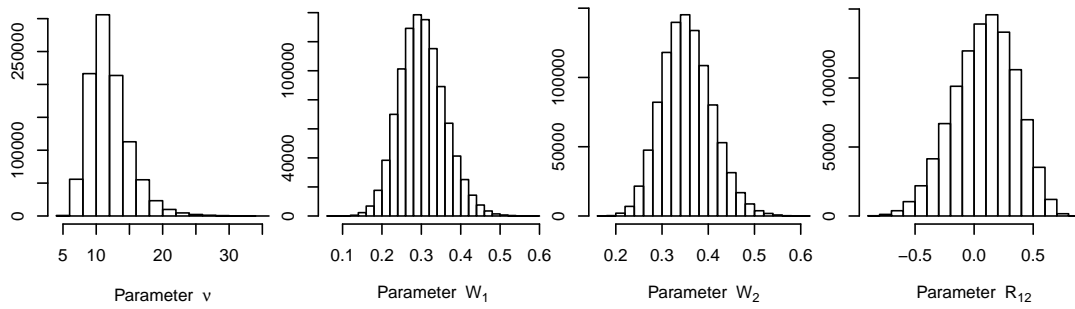


Figure 4.3: Step 1 – Parameters sampling using M-H

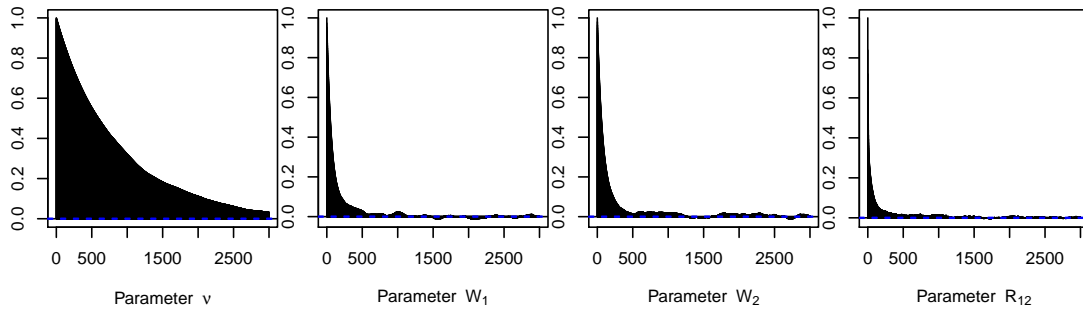


Figure 4.4: Step 2 – ACF of simulated parameters

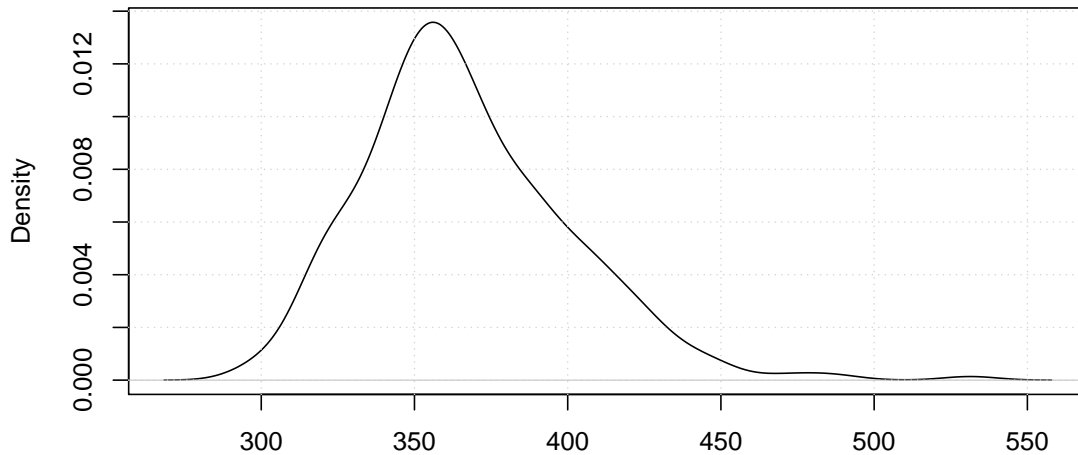
```
-DW2=0.32 -DR12=-0.28 samples/test05.xml > data/MH002/ccrunner.out;
...
mkdir data/MH300;
bin/ccrunner-cmd -o data/MH300 -DNU=10.06 -DW1=0.33 \
-DW2=0.36 -DR12=0.02 samples/test05.xml > data/MH300/ccrunner.out;
```

Listing 4.2: $ES_{99\%}$ distribution (R script)

```
getRisk <- function(dir)
{
  filename = paste(dir, "/portfolio.csv", sep="")
  data <- read.csv(filename, comment.char="#")
  X = sort(data[,1])
  n = length(X)
  Y = X[as.integer(n*0.99):n]
  ES99 = mean(Y)
  sde = sqrt(var(Y)/length(Y))
  return(c(ES99, sde))
}

dirs = dir("data", pattern="MH[[:digit:]]{3}*", full.names=TRUE)
values = matrix(ncol=2, nrow=length(dirs))
colnames(values) = c("ES99", "stderr")
for(i in 1:length(dirs)) {
  values[i,] = getRisk(dirs[i])
}

plot(density(values[,1]))
```

Figure 4.5: ES_{99%} distribution

4.3 Portfolio Optimization

Once we know the portfolio credit risk, naturally arises the question “How can I balance the weights assigned to each business unit so that the total portfolio risk is minimized?” In section 3.2.2, we have seen that CCruncher allows defining sub-portfolios and it can disaggregate the portfolio credit risk in terms of them. In this section we will see how to modify the weights assigned to each sub-portfolio in order to optimize the portfolio credit risk. First of all, let’s define the problems considered:

Definition 4.1 — Portfolio optimization. Let a portfolio composed of K sub-portfolios. We want to determine the sub-portfolio weights α_k that minimize one of the following problems².

EL minimization

$$\min_{\alpha_k} \text{EL}(L)$$

Subjecto to:

$$\left. \begin{array}{l} x^0 = \text{ES}_{99\%}(L) \\ 0 \leq \alpha_k \leq 1 \quad \forall k \\ \sum_{k=1}^K \alpha_k = 1 \end{array} \right\}$$

ES minimization

$$\min_{\alpha_k} \text{ES}_{99\%}(L)$$

Subjecto to:

$$\left. \begin{array}{l} y^0 = \text{EL}(L) \\ 0 \leq \alpha_k \leq 1 \quad \forall k \\ \sum_{k=1}^K \alpha_k = 1 \end{array} \right\}$$

EL-ES minimization

$$\min_{\alpha_k} \sqrt{\text{EL}^2 + \text{ES}_{99\%}^2}$$

Subjecto to:

$$\left. \begin{array}{l} 0 \leq \alpha_k \leq 1 \quad \forall k \\ \sum_{k=1}^K \alpha_k = 1 \end{array} \right\}$$

where $L = L_1 + \dots + L_k$ is the portfolio loss, and x^0 and y^0 are fixed values.

Sub-portfolio weights α_k are the sub-portfolio exposures relative to total portfolio exposure at 1-year. The way to balance the portfolio composition is to add or subtract obligors preserving the characteristics of each sub-portfolio. When we try to model this procedure we find two problems: 1) losses of portfolios with distinct composition are not comparable, 2) it is unclear the procedure to modify the composition of a sub-portfolio (we add half a debtor?, to which rating and sector we assign the new obligor?, new obligors have an averaged exposure and recovery?).

²We use ES_{99%} but you can use any other risk measure depending on copula parameters, such as ES_{95%} or VaR_{99%}.

Fortunately these two problems are solvable. The first one is avoided considering the relative losses on exposure rather than absolute losses. Thus, the relative portfolio loss at time horizon T is

$$L_T = \frac{\sum_{i=1}^n \mathbb{1}_{[0,T]}(t_i) \cdot \left(\sum_{j=1}^{m_i} \text{EAD}_i^j(t_i) \cdot \text{LGD}_i^j(t_i) \right)}{\sum_{i=1}^n \left(\sum_{j=1}^{m_i} \text{avg}_{[0,T]} \left(\text{EAD}_i^j(t) \right) \right)}$$

where T is the time horizon where portfolio loss is computed, n is the number of obligors, (t_1, \dots, t_n) are the obligors' default times, m_i is the number of assets of the i -th obligor, and $\text{EAD}_i^j(t)$ and $\text{LGD}_i^j(t)$ are the EAD and LGD functions of the j -th asset of the i -th obligor evaluated at time t .

The second problem, the portfolio composition, is avoided by equating the number of obligors to exposures. Suppose we have two sub-portfolios A and B with the same exposure, the first one with 500 obligors, and the second one with 500 obligors. We want to balance the portfolio so that ends up with 25% of the exposure in A and 75% in B. The way to implement it is eliminating half of the obligors from A and adding 250 new obligors similar to those of B. That said, if sub-portfolios A and B have a large number of homogeneous individuals, then we can maintain the original number of individuals (500 + 500) and balance the portfolio by multiplying by 0.5 the A's exposures and by 1.5 the B's exposures.

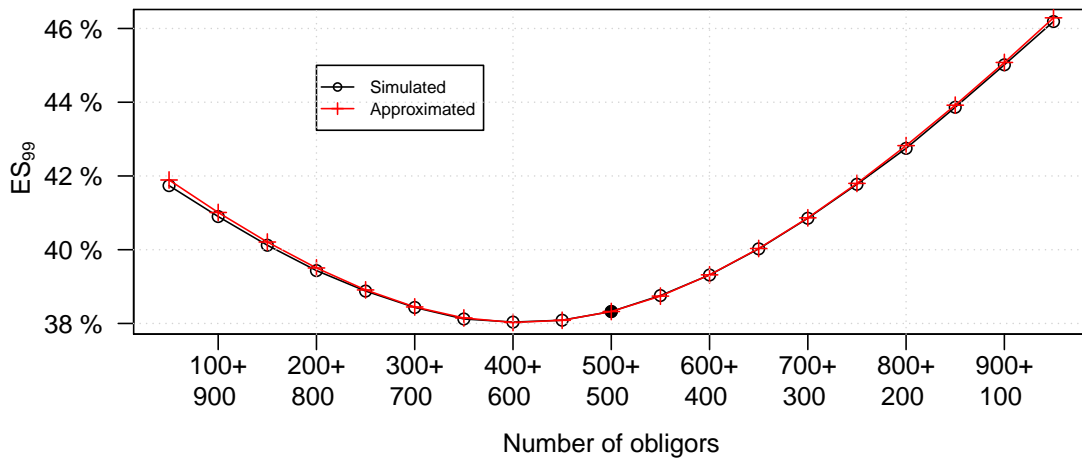


Figure 4.6: ES_{99%} for distinct percentages of S1 and S2

Example 4.4 — Sub-portfolio weights. To check the accuracy of the equivalence between the number of individuals and the exposures we analyze the case `test05.xml` (see example 3.3) considering the sub-portfolios S1 (= S1A + S1B + S1C) and S2 (= S2A + S2B + S2C). In this example we consider only two sub-portfolios, rather than the six defined, so that the values can be clearly represented in the plane. First, we simulate different portfolio compositions (50+950, 100+900, ..., 950+50) and calculate the ES_{99%} for each of them. Next, we consider a fixed number of individuals (500+500) and we change the exposure of individuals according to its sector so that we mimic the previous compositions. Figure 4.6 displays the values modifying the composition of the portfolio (simulated) and the values

modifying the exposure of the obligors (approximated). In this case, both values are similar, with an error less than 0.15%.

The equivalence between the number of debtors and the exposures has a very important advantage since it allows optimizing without having to do multiple simulations. This is achieved by simulating the losses of the current portfolio and by reusing these values to calculate any desired composition of the portfolio. The following algorithm computes the risk of a weighted portfolio reusing the simulated losses from the current portfolio.

Algorithm 4.2 — Risk of a weighted portfolio. Let a portfolio composed of K sub-portfolios with an average exposure E_k for each of them in the period $[0, T]$. We have simulated the portfolio losses broken down by sub-portfolios. We want compute the EL and $ES_{99\%}$ of this portfolio with weights $\alpha_1, \dots, \alpha_K$.

1. Obtain multiplicative factors: $\lambda_k = \alpha_k \cdot \frac{E}{E_k} = \frac{\alpha_k}{\alpha_k^0}$
2. Create weighted losses: $y_i = \sum_{k=1}^K \lambda_k \cdot x_i^k \quad i = 1, \dots, N$
3. Compute EL (see section 3.2) using values $\{y_i\}_{i=1, \dots, N}$ and divide this value by E to obtain the EL relative to total exposure.
4. Compute $ES_{99\%}$ (see section 3.2) using values $\{y_i\}_{i=1, \dots, N}$ and divide this value by E to obtain the $ES_{99\%}$ relative to total exposure.

where $E = \sum_{k=1}^K E_k$ is the total exposure, α_k^0 is the weight of the current k -th sub-portfolio, N is the number of simulated losses, and x_i^k is the i -th simulated loss of the k -th sub-portfolio.

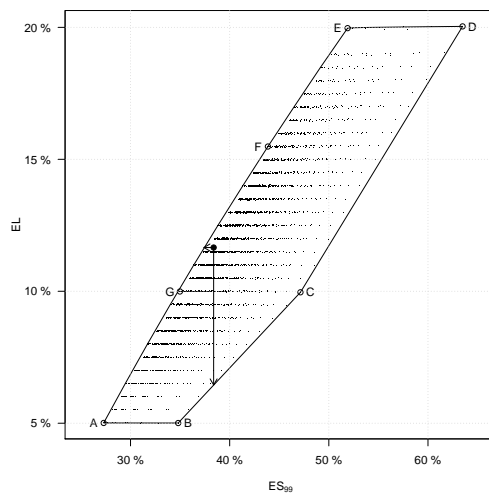
The risk of weighted portfolio is continuous with respect to $\alpha_1, \dots, \alpha_K$ but it is not linear with respect to this parameters because the rank statistic involved in the $VaR_{99\%}$ and $ES_{99\%}$ measures. This is clearly shown in figure 4.6.

Example 4.5 — Feasible portfolios. We consider the portfolio defined in `test05.xml` (see example 3.3). We want to know the values that can take the risk measures by changing the composition of the portfolio. To achieve this, we consider the following weights:

$$\left\{ (\alpha_1, \dots, \alpha_6) \mid \alpha_i \in \{0, 0.1, 0.2, \dots, 0.9, 1\} \text{ and } \sum_{i=1}^6 \alpha_i = 1 \right\}$$

This set of weights has 3003 elements. For each element of this set we compute the EL and $ES_{99\%}$ of the weighted portfolio applying the algorithm 4.2. Result is displayed in figure 4.7. We have drawn the convex hull of the region and we have identified the relevant points. The original portfolio is marked with a black dot and the corresponding optimal portfolios are indicated with a arrow. The appearance of the EL-ES region is determined by the involved sub-portfolios, in this case: S1A, S1B, S1C, S2A, S2B, S2C. Other segmentations (e.g., geographic areas, types of products) will lead to EL-ES regions with a different shape.

Now we have all the elements to solve the three optimization problems. Instead of presenting the algorithm in pseudo-code, this time we use a basic R script exposed in listing 4.3 to illustrate the procedure. The optimized portfolio is the case `test05.xml` and optimization results are listed in table 4.1. Note that $K - 1$ parameters are optimized rather than K . This is because we can obtain the former doing $\alpha_1 = 1 - \sum_{i=2}^K \alpha_i$. Note also that penalty functions are used to ensure the restrictions compliance.



(a) Portfolio values

	S1A	S1B	S1C	S2A	S2B	S2C
A	40%	0%	0%	60%	0%	0%
B	100%	0%	0%	0%	0%	0%
C	0%	100%	0%	0%	0%	0%
D	0%	0%	100%	0%	0%	0%
E	0%	0%	40%	0%	0%	60%
F	10%	0%	30%	20%	0%	40%
G	20%	10%	10%	40%	0%	20%

(b) Portfolio composition

Figure 4.7: test05.xml feasible portfolios

Listing 4.3: Portfolio optimization (R script)

```

evalPortfolio <- function(losses, w0, w1, level=0.99)
{
  if (length(w0) != length(w1)) stop("invalid_parameters")
  if (ncol(losses) != length(w0)) stop("invalid_losses")
  if (level <= 0 || level >= 1) stop("ES_level_out-of-range_(0,1)")
  n = nrow(losses)
  k = ncol(losses)
  lambda = w1/w0
  loss = rep(0, n)
  for(i in 1:k) {
    loss = loss + losses[,i]*lambda[i]
  }
  loss = sort(loss)
  e1 = mean(loss)
  es = mean(loss[as.integer(n*level):n])
  return(c(e1, es))
}

getW <- function(params)
{
  w = c(0, params)
  w[1] = 1-sum(w)
  return(w)
}

f <- function(params, losses, w0, option, val, level=0.99)
{
  if (!(option %in% (1:3))) stop("unrecognized_option")
  w1 = getW(params)
  aux = 0
  if (w1[1] < 0) {
    aux = w1[1]
    w1[1] = 0
  }
  eles = evalPortfolio(losses, w0, w1, level)
  if (aux != 0) {

```

```

    eles = eles + abs(aux)*10
  }
  if (option == 1) {
    ret = eles[1] + 10*(val-eles[2])^2
    return(ret)
  }
  else if (option == 2) {
    ret = eles[2] + 10*(val-eles[1])^2
    return(ret)
  }
  else if (option == 3) {
    ret = sqrt(sum(eles^2))
    return(ret)
  }
  else return(NA)
}

losses <- read.csv("data/sector-rating.csv", comment.char="#")
exposures = c(167, 166, 167, 167, 167, 166)
w0 = exposures/sum(exposures)
eles0 = evalPortfolio(losses, w0, w0)

# EL minimization
params = w0[2:length(w0)]
opt = nlminb(params, f, losses=losses, w0=w0, option=1, val=eles0[2],
  lower=rep(0,length(params)), upper=rep(1,length(params)))
w1 = getW(opt$par)
evalPortfolio(losses, w0, w1)

# ES minimization
params = w0[2:length(w0)]
opt = nlminb(params, f, losses=losses, w0=w0, option=2, val=eles0[1],
  lower=rep(0,length(params)), upper=rep(1,length(params)))
w2 = getW(opt$par)
evalPortfolio(losses, w0, w2)

# optimization
params = w0[2:length(w0)]
opt = nlminb(params, f, losses=losses, w0=w0, option=3, val=NA,
  lower=rep(0,length(params)), upper=rep(1,length(params)))
w3 = getW(opt$par)
evalPortfolio(losses, w0, w3)

```

	S1A	S1B	S1C	S2A	S2B	S2C	EL	ES
current	16.7%	16.6%	16.7%	16.7%	16.7%	16.6%	11.6%	38.3%
min EL	68.9%	31.1%	0%	0%	0%	0%	6.6%	38.3%
min ES	18.7%	0%	21.3%	36.8%	0%	23.2%	11.6%	37.2%
min EL-ES	41.0%	0%	0%	59.0%	0%	0%	5.0%	27.0%

Table 4.1: test05.xml optimization

A — Appendices

A.1 The Probabilities of Default

In this section, we construct the univariate time-until-default obligor distributions using available information such as the 1-year PDs or the ratings transition matrix. The objective is to determine the cumulative density functions $F_i(t) = \Pr\{T_i \leq t\}$ for each of the ratings.

Definition A.1 — Probability of Default (PD). T_i is the random variable that represents the default time (aka time-until-default, aka survival time, aka time-to-default) of the i -th obligor. Let $F_i(t)$ denote the distribution function of T_i :

$$F_i(t) = \Pr\{T_i \leq t\} = \Pr\{\text{i-th obligor defaults before } t \text{ years}\}$$

Note that $F_i(t)$ is the probability of default (PD) of the i -th obligor at time horizon t . PD is a key parameter used in the calculation of credit risk under Basel II. We indicate as p_i the probability that i -th obligor defaults within the time range $[0, 1]$.

$$p_i = \Pr\{T_i \leq 1 \text{ year}\} = F_i(1)$$

Each obligor has its own time-until-default distribution depending on its rating; therefore, all obligors with identical ratings have identical distributions. Below, we indicate three different manners of inferring F_i depending upon the amount of information available. All three manners produce similar results in the short term (e.g., 1 year) but differ in the middle and long term. Before detailing the different approaches, we define the transition matrix because it is used in all three cases.

Definition A.2 — Transition matrix. The T -years transition matrix gives the probability of changing from i -th rating to j -th rating in a period of T years:

$$M_T = \begin{pmatrix} m_{11} & \cdots & m_{1r} \\ \vdots & \ddots & \vdots \\ m_{r1} & \cdots & m_{rr} \end{pmatrix}$$

in which r is the number of ratings and m_{ij} is the probability that an obligor with i -th rating ends up having the j -th rating after T years. Note that row sums are 1 and that two consecutive transition matrices, M_{T_1} and M_{T_2} , give $M_{T_1+T_2} = M_{T_1} \cdot M_{T_2}$. A valid transition matrix must have a unique *defaulted* rating, and any rating ends up defaulting.

Example A.1 — 1-year transition matrix. Table A.1 shows a transition matrix, extracted from [19, p. 20], in which the probability that an obligor with rating AA changes to rating B in one year is 0.14%. Note that the last column contains the 1-year default probabilities for each rating and that the last row corresponds to the *defaulted* rating.

	AAA	AA	A	BBB	BB	B	CCC	Default
AAA	90.81	8.33	0.68	0.06	0.12	0.00	0.00	0.00
AA	0.70	90.65	7.79	0.64	0.06	0.14	0.02	0.00
A	0.09	2.27	91.05	5.52	0.74	0.26	0.01	0.06
BBB	0.02	0.33	5.95	86.93	5.30	1.17	0.12	0.18
BB	0.03	0.14	0.67	7.73	80.53	8.84	1.00	1.06
B	0.00	0.11	0.24	0.43	6.48	83.46	4.07	5.21
CCC	0.22	0.00	0.22	1.30	2.38	11.24	64.86	19.78
Default	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00

Table A.1: 1-year transition matrix %

A.1.1 Obtaining PDs from the transition matrix

This alternative implies that transition probabilities remain constant over time, undermining the economic cycle effect. In practice, this constancy is equivalent to considering an averaged economic cycle. See document [12, sec. 3.2.1] to obtain additional information related with this section and the CreditMetricsTM approach.

Proposition A.1 — Scaled transition matrix. The transition matrix can be scaled in time using the following rules:

- $M_{k,T} = M_T^k$
- $M_{T/k} = \sqrt[k]{M_T}$

The root of a matrix M can be obtained using the spectral decomposition $M = P \cdot D \cdot P^{-1}$ doing $M^k = P \cdot D^k \cdot P^{-1}$, in which P and D are the eigenvectors and eigenvalue matrices of M .

Sometimes, the scaled transition matrix does not satisfy the Markov conditions (the row sum is equal to one, and all elements are non-negatives). In this case, we must transform this matrix to the relevant Markov matrix. This process is called regularization. Below is the QOM regularization algorithm extracted from [24].

Algorithm A.1 — Transition matrix regularization. The QOM (Quasi-Optimization of the root Matrix) algorithm regularizes an $n \times n$ transition matrix M , row by row. The steps to regularizing the i -th row are

1. Compute the difference between the row sum and one. Divide by n and subtract this value from all non-zero components:

$$m_{ij} \neq 0 \implies m_{ij} = m_{ij} - \frac{1}{n} \left(\sum_{k=1}^n m_{ik} - 1 \right)$$

2. If all the row elements are non-negative and add up to one, then stop: the row is regularized.
3. Adjust any negative row element to zero and go to Step 1.

The algorithm stops after m steps, where $m \leq n$. Apply the previous algorithm to every row. The final matrix is a regularized matrix.

Proposition A.2 — PDs derived from transition matrix. If M_T is a transition matrix, then the time-until-default distribution for an obligor with the i -th rating is

$$F_i(t) = \Pr\{T_i \leq t\} = (M_t)_{id}$$

in which T_i is the time-until-default of an obligor with the i -th rating, d is the index of the *defaulted* rating, M_t is the transition matrix scaled to time t , and $(M)_{ij}$ is the matrix element located in the i -th row and the j -th column.

Example A.2 We derive the time-until-default distribution for each rating induced by the transition matrix presented in table A.1. As a first step, we scale the 1-year transition matrix M_1 to the 1-month transition matrix $M_{\frac{1}{12}}$.

$$M_{\frac{1}{12}} = \begin{pmatrix} 99.1972 & 0.7588 & 0.0320 & 0.0020 & 0.0112 & -0.0011 & -0.0001 & 0.0000 \\ 0.0635 & 99.1744 & 0.7083 & 0.0392 & 0.0015 & 0.0120 & 0.0018 & -0.0007 \\ 0.0074 & 0.2057 & 99.1980 & 0.5102 & 0.0557 & 0.0189 & -0.0001 & 0.0042 \\ 0.0014 & 0.0239 & 0.5507 & 98.8021 & 0.5164 & 0.0871 & 0.0077 & 0.0106 \\ 0.0027 & 0.0113 & 0.0396 & 0.7581 & 98.1557 & 0.8774 & 0.0889 & 0.0665 \\ -0.0007 & 0.0098 & 0.0196 & 0.0111 & 0.6447 & 98.4443 & 0.4463 & 0.4240 \\ 0.0233 & -0.0023 & 0.0170 & 0.1287 & 0.2166 & 1.2298 & 96.4213 & 1.9667 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 100 \end{pmatrix}$$

We observe that $M_{\frac{1}{12}}$ is not a regular matrix because of negative values. We apply the QOM algorithm to regularize it.

$$\bar{M}_{\frac{1}{12}} = \begin{pmatrix} 99.1969 & 0.7586 & 0.0317 & 0.0018 & 0.0110 & 0.0000 & 0.0000 & 0.0000 \\ 0.0634 & 99.1744 & 0.7082 & 0.0391 & 0.0014 & 0.0119 & 0.0017 & 0.0000 \\ 0.0074 & 0.2057 & 99.1980 & 0.5102 & 0.0557 & 0.0189 & 0.0000 & 0.0041 \\ 0.0014 & 0.0239 & 0.5507 & 98.8021 & 0.5164 & 0.0871 & 0.0077 & 0.0106 \\ 0.0027 & 0.0112 & 0.0395 & 0.7581 & 98.1557 & 0.8774 & 0.0889 & 0.0665 \\ 0.0000 & 0.0099 & 0.0197 & 0.0111 & 0.6447 & 98.4444 & 0.4463 & 0.4240 \\ 0.0228 & 0.0000 & 0.0165 & 0.1282 & 0.2161 & 1.2293 & 96.4208 & 1.9663 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 100 \end{pmatrix}$$

Finally, we use proposition A.2 to determine the PD function for each rating and month. The results are displayed in figure A.1.

A.1.2 Obtaining PDs from annual values

In some circumstances, the transition matrix is not available, and we only have the probability of default at 1-year for each rating, p_1, \dots, p_{r-1}, p_r in which the r -th rating is the *defaulted* status. In the absence of more information, we can consider a diagonal transition matrix such as the following:

$$M_1 = \begin{pmatrix} 1 - p_1 & \cdots & 0 & p_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 - p_{r-1} & p_{r-1} \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

This simplified case has a closed-form solution stated by the following proposition.

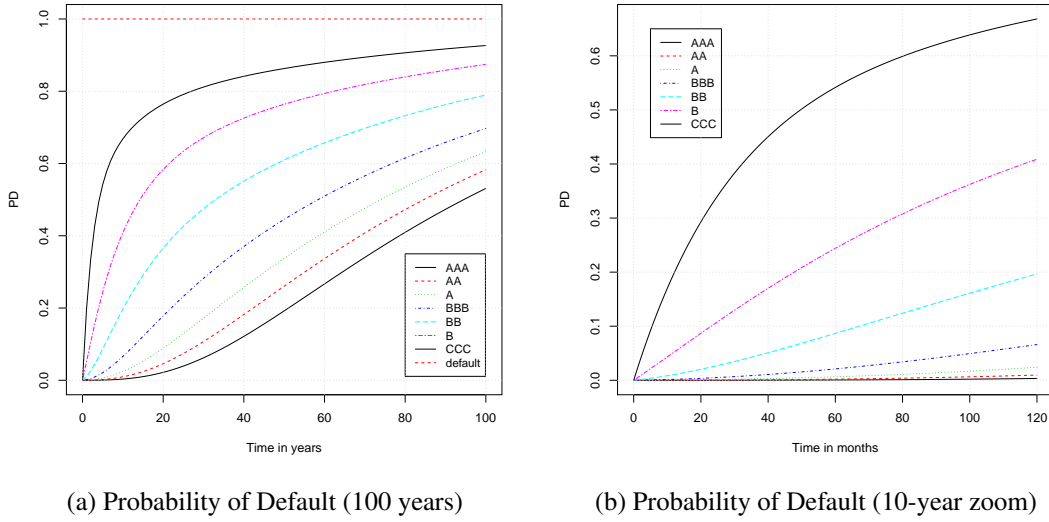


Figure A.1: PDs derived from transition matrix

Proposition A.3 — PDs derived from annual values. If the time-until-default of an obligor satisfies a transition matrix and only depends on the 1-year probability of default p_i , then this time-until-default follows the exponential distribution, i.e., $T_i \sim \text{Exp}(-\ln(1 - p_i))$.

$$F_i(t) = \Pr\{T_i \leq t\} = 1 - e^{-\lambda_i t} \quad \text{where } \lambda_i = -\ln(1 - p_i) \quad i = 1, \dots, r$$

Proof. We note that ratings are detached from one another. We use proposition A.2 to obtain the time-until-default distribution of the i -th rating:

$$\begin{pmatrix} 1 - p_i & p_i \\ 0 & 1 \end{pmatrix}^t = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 - p_i \end{pmatrix}^t \cdot \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} (1 - p_i)^t & 1 - (1 - p_i)^t \\ 0 & 1 \end{pmatrix}$$

Then,

$$F_i(t) = 1 - (1 - p_i)^t = 1 - e^{-\lambda_i t} \quad \text{where } \lambda_i = -\ln(1 - p_i)$$

□

Example A.3 Suppose that we only know the annual probabilities of default and that these values are those from the 1-year regularized transition matrix displayed in table A.1. We call the 1-year regularized transition matrix to $\bar{M}_1 = \bar{M}_{12}^{12}$ where \bar{M}_{12} is computed in example A.2. In this case, the 1-year diagonal transition matrix is

$$M_1 = \begin{pmatrix} 99.9991 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0009 \\ 0 & 99.9923 & 0 & 0 & 0 & 0 & 0 & 0.0077 \\ 0 & 0 & 99.9400 & 0 & 0 & 0 & 0 & 0.0600 \\ 0 & 0 & 0 & 99.8200 & 0 & 0 & 0 & 0.1800 \\ 0 & 0 & 0 & 0 & 98.9401 & 0 & 0 & 1.0599 \\ 0 & 0 & 0 & 0 & 0 & 94.7997 & 0 & 5.2003 \\ 0 & 0 & 0 & 0 & 0 & 0 & 80.2154 & 19.7846 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 \end{pmatrix}$$

We determine the PD's function for each rating using proposition A.3, i.e., considering that times-until-default are exponential distributions. These results are displayed in figure A.2. Observe that values around $T = 1$ are similar to those of example A.2; however, they differ as they move away from this.

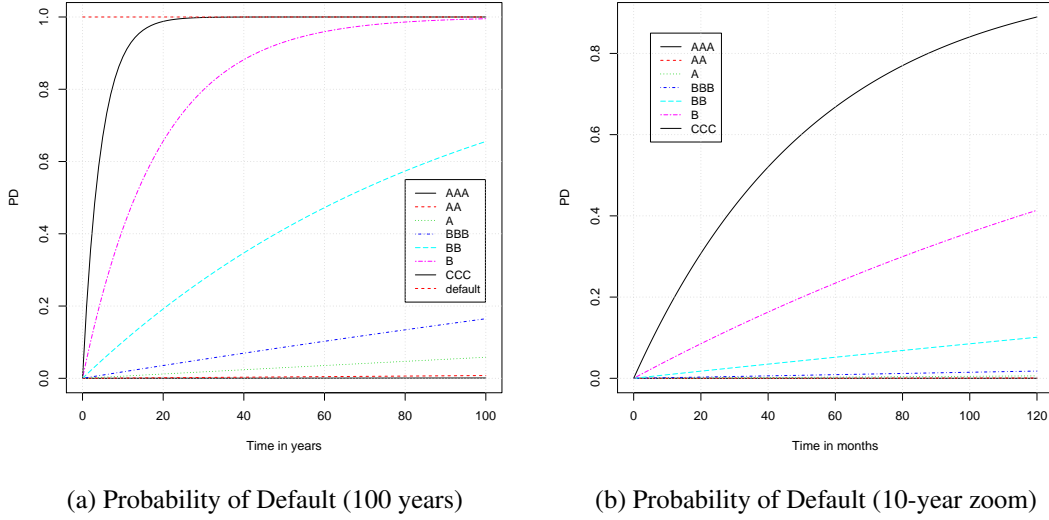


Figure A.2: PDs derived from annual values

A.1.3 Obtaining PDs from advanced models

There are financial institutions that may have models to forecast the transition matrix using some economic indicators such as GDP or inflation. We note as $M_T(t)$ the T-period transition matrix at time t . In these cases, the time-until-default distributions can be obtained successively composing the transition matrices for each T-period.

$$F_i(t) = \Pr\{T. \leq t\} = \left(\left(\prod_{j=1}^{\lfloor t/T \rfloor} M_T(j) \right) \cdot M_{t-T \cdot \lfloor t/T \rfloor}(\lfloor t/T \rfloor + 1) \right)_{id}$$

in which $[x]$ is the integer part of x , T is the time-until-default of an obligor with the i -th rating, d is the index of the *defaulted* rating, $M_t(j)$ is the T-period transition matrix evaluated at time j and scaled to time t , and $(M)_{ij}$ is the matrix element located in the i -th row and the j -th column.

The time-until-default distributions derived from an advanced model may show various steps according to the different stages of the economic cycle. This approach is the most appropriate for risk estimates in the medium and long term.

CCruncher allows description of the PDs derived from advanced models specifying the PDs' values at given time points. These must fulfill the cdf conditions:

- $F_i(0) = 0$ $i = 1, \dots, r$ $i \neq \text{default}$
- $F_i(t)$ strictly increasing $i = 1, \dots, r$ $i \neq \text{default}$
- $F_i(\infty) = 1$ $i = 1, \dots, r$
- $F_{\text{default}}(t) = 1$ $\forall t \geq 0$

The functions specified in this manner are interpolated using cubic splines when the above conditions are preserved and are interpolated using linear splines when they are not.

A.2 Copula Basics

It is common to use correlation as a measure of dependency between random variables. In most cases, this measure does not fully reflect the structure of dependence between the variables. Figure A.3 displays two cases with the identical mean, marginals and correlations but a distinct risk profile because the two cases' underlying dependence structure is distinct (Gaussian copula vs. t-Student(3) copula). The mathematical concept that does reflect the structure of dependence between random variables, however, is the copula, which we define below. The world of copulas is vast and quite interesting. See [27, chap. 5] for a pleasant introduction to copulas.

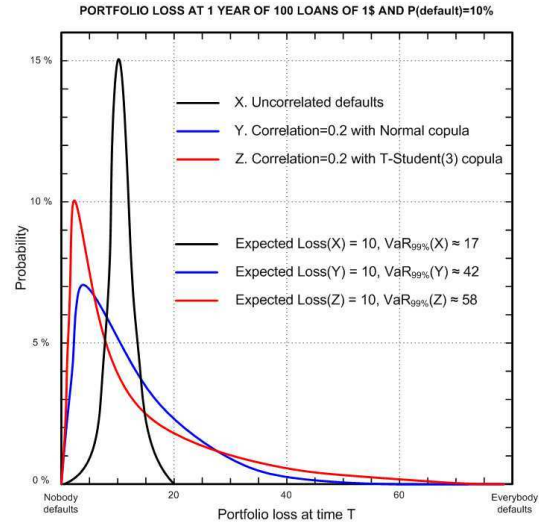


Figure A.3: Dependence structure effect

Definition A.3 — Copula. A copula function C is a multivariate distribution defined on the unit hypercube $[0, 1]^d$ with standard uniform marginals. More precisely,

$$C(u_1, \dots, u_d) = \Pr\{U_1 \leq u_1, \dots, U_d \leq u_d\}$$

in which $U_i \sim \text{Uniform}(0, 1)$ for $i = 1, \dots, d$.

Sklar's theorem [33] states that any multivariate distribution with continuous marginals can be decomposed into the marginals and a copula that reflects the structure of dependence between them. Later, we will use this statement to define and simulate the t-Student copula.

Theorem A.1 — Sklar's theorem. Let F be a d -dimensional distribution function with margins F_1, \dots, F_d . Then there is a d -copula C such that for all $x \in \mathbb{R}^d$,

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

If F_1, \dots, F_d are all continuous, the C is unique; otherwise, C is uniquely determined on $\text{Ran}F_1 \times \dots \times \text{Ran}F_d$. Conversely, if C is a d -copula and F_1, \dots, F_d are distribution functions, then the function F defined above is a d -dimensional distribution function with margins F_1, \dots, F_d .

Corollary A.1 — Copula of a multivariate distribution. Let $X = (X_1, \dots, X_d)$ be a random vector with a multivariate distribution F and continuous marginals F_1, \dots, F_d . Then its copula is

$$C(u_1, \dots, u_d) = F(F_1^{-1}(u_1), \dots, F_d^{-1}(u_d))$$

Corollary A.2 — Copula simulation. Let $X = (X_1, \dots, X_d)$ be a random vector with a multivariate distribution F and continuous marginals F_1, \dots, F_d . If we have a procedure to

simulate X , then we can simulate its copula C using:

$$(U_1, \dots, U_d) = (F_1(X_1), \dots, F_d(X_d))$$

in which U_i are the copula components.

Corollary A.3 — Multivariate distribution simulation. Let $X = (X_1, \dots, X_d)$ be a random vector with a copula C and continuous marginals F_1, \dots, F_d . If we have a procedure to simulate C , then we can simulate X using

$$(X_1, \dots, X_d) = (F_1^{-1}(U_1), \dots, F_d^{-1}(U_d))$$

in which U_i are the copula components.

Definition A.4 Spearman's rho is a measure of the linear relation between the two variables X and Y , and it is defined as the Pearson's correlation coefficient between the ranked variables, x and y :

$$\rho_s = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

It has the important property that its value only relies on the underlying copula C , not on marginals. It can be written as

$$\rho_s = 12 \int_0^1 \int_0^1 C(u, v) \, du \, dv - 3$$

A.3 Multivariate t-Student Distribution

The majority of the results presented below have been extracted from [23, 9].

Definition A.5 — Multivariate t-Student distribution. The d -dimensional random vector $X = (X_1, \dots, X_d)$ is said to have a (non-singular) multivariate t-Student distribution with ν degrees of freedom, mean vector $\vec{\mu}$ and positive-definite dispersion or scatter matrix Σ , denoted $t_d(\nu, \vec{\mu}, \Sigma)$, if its density is given by

$$f(\vec{x}) = \frac{\Gamma(\frac{\nu+d}{2})}{\Gamma(\frac{\nu}{2}) \sqrt{(\pi\nu)^d |\Sigma|}} \left(1 + \frac{(\vec{x} - \vec{\mu})^\top \Sigma^{-1} (\vec{x} - \vec{\mu})}{\nu} \right)^{-\frac{\nu+d}{2}}$$

in which Γ is the gamma function and $|\Sigma|$ is the determinant of the matrix.

Proposition A.4 — t-Student covariance. The covariance matrix of the $X \sim t_d(\nu, \vec{\mu}, \Sigma)$ is

$$\text{Cov}(X) = \frac{\nu}{\nu-2} \cdot \Sigma \quad \text{if } \nu > 2$$

Proposition A.5 — Gaussian as limit of the t-Student. The t-Student distribution converges to a Gaussian distribution when ν tends to ∞ .

$$\lim_{\nu \rightarrow \infty} t_d(\nu, \vec{\mu}, \Sigma) = N(\vec{\mu}, \Sigma)$$

Proposition A.6 — Multivariate t-Student characterization. A random vector $T \sim t_d(\nu, \vec{\mu}, \Sigma)$ can be expressed as:

$$T \stackrel{d}{=} \vec{\mu} + \sqrt{\frac{\nu}{V}} \cdot Z \quad \text{where } Z \sim N(\vec{0}, \Sigma) \text{ and } V \sim \chi_\nu^2$$

Proposition A.7 — Multivariate t-Student marginals. Assume $X \sim t_d(\nu, \vec{0}, \Sigma)$. Then its i -th marginal is $X_i \sim t_1(\nu, 0, \sigma_{ii})$.

Definition A.6 — t-Student copula. The t-Student copula, $C_{\nu, \Sigma}^t$, is the copula of the multivariate t-Student distribution $t_d(\nu, \vec{\mu}, \Sigma)$.

Proposition A.8 — t-Student copula equivalence. The copula of $t_d(\nu, \vec{\mu}, \Sigma)$ is identical to that of $t_d(\nu, \vec{0}, R)$ in which R is the correlation matrix implied by the dispersion matrix Σ .

Proposition A.9 — t-Student copula density. The t-Student copula, $C_{\nu, R}^t$, in which R is a correlation matrix, has the following distribution function:

$$C_{\nu, R}^t(u_1, \dots, u_d) = \int_{-\infty}^{t_\nu^{-1}(u_1)} \dots \int_{-\infty}^{t_\nu^{-1}(u_d)} f(x) \, dx$$

in which $f(x)$ is the density function of $t_d(\nu, \vec{0}, R)$ and t_ν^{-1} denotes the quantile function of the univariate distribution $t_1(\nu, 0, 1)$. The copula density is

$$c_{\nu, R}^t(u_1, \dots, u_d) = |R|^{-\frac{1}{2}} \frac{\Gamma(\frac{\nu+d}{2})}{\Gamma(\frac{\nu}{2})} \left[\frac{\Gamma(\frac{\nu}{2})}{\Gamma(\frac{\nu+1}{2})} \right]^d \frac{\left(1 + \frac{\zeta' R^{-1} \zeta}{\nu}\right)^{-\frac{\nu+d}{2}}}{\prod_{i=1}^d \left(1 + \frac{\zeta_i^2}{\nu}\right)^{-\frac{\nu+1}{2}}}$$

in which $\zeta = (t_\nu^{-1}(u_1), \dots, t_\nu^{-1}(u_d))$ is the vector of the t-Student univariate inverse distribution functions.

A.4 Covariance Block Matrix

The content presented in this appendix is detailed and extended in [35].

Definition A.7 — Covariance block matrix. A matrix A is a covariance block matrix $B_k(\vec{n}, \vec{d}, M)$ in which:

- $\vec{n} = (n_1, \dots, n_k)$ with $n_i \in \mathbb{N}$ and $1 \leq n_i$ (number of elements per block),
- $\vec{d} = (d_1, \dots, d_k)$ with $d_i \in \mathbb{R}$ (diagonal block values), and
- M is a $k \times k$ symmetric matrix with values $m_{ij} \in \mathbb{R}$ (block values)

when

- each block B_{ij} is an $n_i \times n_j$ constant matrix with value m_{ij}
- diagonal blocks B_{ii} have diagonal values d_i
- A is definite-positive.

Example A.4 — Correlation block matrix. We have 6 obligors sorted by sector. The first three belong to the banking sector, the next two belong to the energy sector and the last one belongs to the services sector. Obligor default time dependence is determined only by sectors. Thus, the correlation matrix of the underlying elliptical copula is a correlation block matrix.

$$\left. \begin{array}{l} \vec{n} = (3, 2, 1) \\ \vec{d} = (1, 1, 1) \\ M = \begin{pmatrix} 0.5 & 0.2 & 0.1 \\ 0.2 & 0.4 & 0.15 \\ 0.1 & 0.15 & 0.5 \end{pmatrix} \end{array} \right\} \rightarrow B_3(\vec{n}, \vec{d}, M) = \begin{pmatrix} 1 & 0.5 & 0.5 & 0.2 & 0.2 & 0.1 \\ 0.5 & 1 & 0.5 & 0.2 & 0.2 & 0.1 \\ 0.5 & 0.5 & 1 & 0.2 & 0.2 & 0.1 \\ \hline 0.2 & 0.2 & 0.2 & 1 & 0.4 & 0.15 \\ 0.2 & 0.2 & 0.2 & 0.4 & 1 & 0.15 \\ \hline 0.1 & 0.1 & 0.1 & 0.15 & 0.15 & 1 \end{pmatrix}$$

Proposition A.10 — Covariance block matrix eigenvalues. Let $A = B_k(\vec{n}, \vec{d}, M)$ a non-singular matrix, and let G be the $k \times k$ deflated matrix

$$G = \begin{pmatrix} d_1 + (n_1 - 1) \cdot m_{11} & n_2 \cdot m_{12} & \cdots & n_k \cdot m_{1k} \\ n_1 \cdot m_{21} & d_2 + (n_2 - 1) \cdot m_{22} & \cdots & n_k \cdot m_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ n_1 \cdot m_{k1} & n_2 \cdot m_{k2} & \cdots & d_k + (n_k - 1) \cdot m_{kk} \end{pmatrix}$$

Then, the eigenvalues of A are as follows:

- $d_i - m_{ii}$ with multiplicity $n_i - 1$ for $i = 1, \dots, k$.
- λ_i , the eigenvalues of G with multiplicity 1.

Corollary A.4 A covariance block matrix $B_k(\vec{n}, \vec{d}, M)$ is definite-positive if

- $d_i > m_{ii}$ for all $i = 1, \dots, k$, and
- the deflated matrix G is definite-positive.

A.5 Multi-Factor Gaussian Model

Definition A.8 — Gaussian multi-factor model. We say that a multivariate distribution X follows a Gaussian multi-factor model if it can be expressed in this form:

$$X_i^j = w_i \cdot Z_i + \sqrt{1 - w_i^2} \cdot \varepsilon_i^j \quad \text{where} \quad \begin{cases} i = 1, \dots, k & k \text{ is the number of factors} \\ j = 1, \dots, n_i & n_i \text{ is } i\text{-th factor size} \\ Z \sim N(\vec{0}, R) & R \text{ is a } k \times k \text{ correlation matrix} \\ w_i \in (0, 1) & \text{the factor loadings} \\ \varepsilon_i^j \sim N(0, 1) \text{ iid} & \forall i, j \\ Z, \varepsilon_i^j \text{ independents} & \forall i, j \end{cases}$$

An alternative definition based on the covariance matrix is

$$X_i^j = Z_i + \sqrt{1 - \Sigma_{ii}} \cdot \varepsilon_i^j \quad \text{where} \quad \begin{cases} i = 1, \dots, k & k \text{ is the number of factors} \\ j = 1, \dots, n_i & n_i \text{ is } i\text{-th factor size} \\ Z \sim N(\vec{0}, \Sigma) & \Sigma \text{ is a } k \times k \text{ covariance matrix} \\ \Sigma_{ii} \in (0, 1) & \text{diagonal values of } \Sigma \\ \varepsilon_i^j \sim N(0, 1) \text{ iid} & \forall i, j \\ Z, \varepsilon_i^j \text{ independents} & \forall i, j \end{cases}$$

Proposition A.11 — The Gaussian multi-factor model is a Gaussian distribution. The Gaussian multi-factor model defined by factor loadings \vec{w} , the factor correlation matrix R , and $n = \sum_{i=1}^k n_i$ is a multivariate Gaussian distribution with an $n \times n$ block correlation matrix $\hat{R} = B_k(\vec{n}, \vec{1}, \Sigma)$ with values $\Sigma_{ij} = w_i \cdot w_j \cdot R_{ij}$.

Proof. We check that the correlation between components of the Gaussian multi-factor model fulfill the equivalence.

Version with the correlation matrix R :

$$\begin{aligned} \text{Var}(X_i^j) &= w_i^2 \cdot \text{Var}(Z_i) + (1 - w_i^2) \cdot \text{Var}(\varepsilon_i^j) + 2 \cdot w_i \cdot \sqrt{1 - w_i^2} \cdot \text{Cov}(Z_i, \varepsilon_i^j) \\ &= w_i^2 + (1 - w_i^2) = 1 \end{aligned}$$

$$\begin{aligned} \text{Cor}(X_{i_1}^{j_1}, X_{i_2}^{j_2}) &= \text{Cov}(X_{i_1}^{j_1}, X_{i_2}^{j_2}) \\ &= w_{i_1} \cdot w_{i_2} \cdot \text{Cov}(Z_{i_1}, Z_{i_2}) + w_{i_1} \cdot \sqrt{1 - w_{i_2}^2} \cdot \text{Cov}(Z_{i_1}, \varepsilon_{i_2}^{j_2}) + \\ &\quad + \sqrt{1 - w_{i_1}^2} \cdot w_{i_2} \cdot \text{Cov}(\varepsilon_{i_1}^{j_1}, Z_{i_2}) + \sqrt{1 - w_{i_1}^2} \cdot \sqrt{1 - w_{i_2}^2} \cdot \text{Cov}(\varepsilon_{i_1}^{j_1}, \varepsilon_{i_2}^{j_2}) \\ &= w_{i_1} \cdot w_{i_2} \cdot \text{Cov}(Z_{i_1}, Z_{i_2}) \\ &= w_{i_1} \cdot w_{i_2} \cdot R_{i_1 i_2} \end{aligned}$$

Version with the covariance matrix Σ :

$$\begin{aligned} \text{Var}(X_i^j) &= \text{Var}(Z_i) + (1 - \Sigma_{ii}) \cdot \text{Var}(\varepsilon_i^j) + 2 \cdot \sqrt{1 - \Sigma_{ii}} \cdot \text{Cov}(Z_i, \varepsilon_i^j) \\ &= \Sigma_{ii} + (1 - \Sigma_{ii}) = 1 \end{aligned}$$

$$\begin{aligned}
\text{Cor}(X_{i_1}^{j_1}, X_{i_2}^{j_2}) &= \text{Cov}(X_{i_1}^{j_1}, X_{i_2}^{j_2}) \\
&= \text{Cov}(Z_{i_1}, Z_{i_2}) + \sqrt{1 - \Sigma_{i_2 i_2}} \cdot \text{Cov}(Z_{i_1}, \epsilon_{i_2}^{j_2}) + \\
&\quad + \sqrt{1 - \Sigma_{i_1 i_1}} \cdot \text{Cov}(\epsilon_{i_1}^{j_1}, Z_{i_2}) + \sqrt{1 - \Sigma_{i_1 i_1}} \cdot \sqrt{1 - \Sigma_{i_2 i_2}} \cdot \text{Cov}(\epsilon_{i_1}^{j_1}, \epsilon_{i_2}^{j_2}) \\
&= \Sigma_{i_1 i_2}
\end{aligned}$$

The characterization of the multivariate Gaussian [2, thm. 2.6.2] posits that if every linear combination of the components of a vector X is normally distributed, then X is normally distributed. It is straightforward to check that the multi-factor Gaussian model fulfills this property and consequently is a multivariate Normal. \square

Proposition A.12 — **Gaussian multi-factor model \subsetneq multivariate Gaussian block.**

Any multi-factor model can be expressed as a multivariate Gaussian block; however, there are multivariate Gaussian blocks that cannot be expressed as a Gaussian multi-factor model.

Proof. Proposition A.11 states that any multi-factor model is a multivariate Gaussian composed of blocks. Next we see that the reciprocal is false. Let $\hat{R} = B_k(\vec{n}, \vec{1}, \Sigma)$, the Gaussian distribution correlation matrix. We use the equivalence $\Sigma_{ij} = w_i \cdot w_j \cdot R_{ij}$ stated in proposition A.11 to determine the cases that cannot be expressed as a multi-factor model.

Case 1. Factor loadings with imaginary values.

$$\Sigma_{ii} = w_i \cdot w_i \cdot R_{ii} = w_i^2$$

If Σ_{ii} is negative, then w_i has an imaginary value. For example,

$$\hat{R} = \left(\begin{array}{cc|c} 1 & -0.5 & 0 \\ -0.5 & 1 & 0 \\ \hline 0 & 0 & 1 \end{array} \right) \rightarrow R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, w = (\sqrt{-0.5}, 1) !!$$

Case 2. Factors correlation with absolute value > 1 .

$$\Sigma_{ij} = w_i \cdot w_j \cdot R_{ij} \rightarrow R_{ij} = \frac{\Sigma_{ij}}{w_i \cdot w_j} = \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii} \cdot \Sigma_{jj}}}$$

If $\Sigma_{ij}^2 > \Sigma_{ii} \cdot \Sigma_{jj}$, then R_{ij} has an absolute value larger than 1, which is not possible because R_{ij} is a correlation value. For example,

$$\hat{R} = \left(\begin{array}{cc|cc} 1 & 0.5 & 0.6 & 0.6 \\ 0.5 & 1 & 0.6 & 0.6 \\ \hline 0.6 & 0.6 & 1 & 0.5 \\ 0.6 & 0.6 & 0.5 & 1 \end{array} \right) \rightarrow R = \begin{pmatrix} 1 & \frac{0.6}{0.5} \\ \frac{0.6}{0.5} & 1 \end{pmatrix} !!, w = (\sqrt{0.5}, \sqrt{0.5})$$

\square

It is disturbing that multivariate block Gaussian distributions exist that cannot be expressed as a multi-factor model. Fortunately, the cases in which this occurs have no practical application to credit risk modeling in which the number of obligors in each block is higher. The following proposition clarifies this theme.

Proposition A.13 — **Gaussian multi-factor model \approx multivariate Gaussian block.**

Assume $X \sim N(\vec{0}, \hat{R})$ in which $\hat{R} = B_k(\vec{n}, \vec{1}, \Sigma)$ is a correlation block matrix. Then X can be expressed as a multi-factor Gaussian model if n_i values are large enough.

Proof. We revise the cases in which the multivariate Gaussian distribution cannot be expressed as a multi-factor model, and we use proposition A.10 to ascertain that if n_i values are large enough, then that is indeed a valid multi-factor model.

Case 1. Factor loadings with imaginary values.

Factor loading w_i with imaginary value is caused by $\Sigma_{ii} < 0$. We use proposition A.10 to determine the restrictions regarding Σ_{ii} . We assume a unique factor. Because \widehat{R} is a covariance block matrix definite-positive, the deflated matrix has a positive determinant. Then,

$$\begin{aligned} 1 + (n_1 - 1) \cdot \Sigma_{11} &> 0 \\ \Sigma_{11} &> \frac{-1}{n_1 - 1} \\ \text{tending } n_1 &\rightarrow \infty \\ \Sigma_{11} &\geq 0 \end{aligned}$$

Case 2. Factors correlation with absolute value > 1 .

A factor correlation larger than 1 is caused by $R_{12} = \frac{\Sigma_{12}}{\sqrt{\Sigma_{11} \cdot \Sigma_{22}}}$ in which $\Sigma_{12}^2 > \Sigma_{11} \cdot \Sigma_{22}$. We use proposition A.10 to determine the restrictions regarding Σ_{12} . Because \widehat{R} is covariance block matrix definite-positive, the deflated matrix has a positive determinant. Then,

$$\begin{aligned} 1 + (n_1 - 1) \cdot \Sigma_{11} + (n_2 - 1) \cdot \Sigma_{22} + (n_1 - 1) \cdot (n_2 - 1) \cdot \Sigma_{11} \cdot \Sigma_{22} - n_1 \cdot n_2 \cdot \Sigma_{12}^2 &> 0 \\ \Sigma_{12}^2 &< \frac{1}{n_1 \cdot n_2} + \frac{(n_1 - 1)}{n_1 \cdot n_2} \cdot \Sigma_{11} + \frac{(n_2 - 1)}{n_1 \cdot n_2} \cdot \Sigma_{22} + \frac{(n_1 - 1) \cdot (n_2 - 1)}{n_1 \cdot n_2} \cdot \Sigma_{11} \cdot \Sigma_{22} \\ \text{tending } n_1 &\rightarrow \infty \text{ and } n_2 \rightarrow \infty \\ \Sigma_{12}^2 &\leq \Sigma_{11} \cdot \Sigma_{22} \end{aligned}$$

□

A.6 Metropolis-Hastings Algorithm

This section contains a brief introduction to Bayesian inference using the Metropolis-Hastings algorithm and is heavily influenced by the book *Bayesian Modeling Using WinBUGS* [29].

Bayesian statistics differ from classical statistical theory because all unknown parameters are considered as random variables. We want to calculate the distribution $f(\theta | y)$ of the parameters θ given the observed data y . According to the Bayes theorem, this can be written as

$$f(\theta | y) = \frac{f(y | \theta) \cdot f(\theta)}{f(y)} \propto f(y | \theta) \cdot f(\theta)$$

We call *posterior distribution* to $f(\theta | y)$ and *prior distribution* to $f(\theta)$. This prior distribution expresses the information available to the researcher before any data are involved in the statistical analysis. $f(y | \theta)$ is the probability of observing y given θ and is also known as the likelihood.

$$f(y | \theta) = \prod_{i=1}^n f(y_i | \theta)$$

When utilizing Bayesian inference, we can identify a closed form of the posterior distribution in only a few cases. Generally this must be determined by numerical methods. One of the most used is the Metropolis-Hastings (M-H) algorithm, which belongs to the family of Markov Chain Monte Carlo (MCMC) methods. These methods are based on the construction of a Markov chain that converges on the target distribution, which, in our case, is the posterior distribution.

Definition A.9 — Markov chain. A Markov chain is a stochastic process $\{\theta^{(1)}, \dots, \theta^{(T)}\}$ such that

$$f(\theta^{(t+1)} | \theta^{(t)}, \dots, \theta^{(1)}) = f(\theta^{(t+1)} | \theta^{(t)}),$$

that is, the distribution of θ at sequence $t + 1$, given all the preceding θ values, depends only on the previous value. When the Markov chain is irreducible, aperiodic, and positive-recurrent, the distribution of $\theta^{(t)}$ converges on its equilibrium distribution, which is independent of the initial values of the chain $\theta^{(0)}$.

The Markov chain is not an unfamiliar concept to us because the transition matrix (see definition A.2) is a Markov chain in which its equilibrium state is the *defaulted* rating (i.e., just any obligor ends up defaulting), allowing us to consider the PDs as cdfs. The Metropolis-Hastings algorithm generates a sample from $f(\theta)$ by constructing a Markov chain that has $f(\theta)$ as the equilibrium distribution.

Algorithm A.2 — Metropolis-Hastings. Let us assume a target distribution $f(x)$ from which we wish to generate a sample $\{x^{(0)}, \dots, x^{(T)}\}$. The Metropolis-Hastings algorithm can be described by the following iterative procedure:

1. Set initial values $x^{(0)}$
2. For $t = 1, \dots, T$ repeat the following steps:
 - Set $x = x^{(t-1)}$
 - Let x' be a random value from the proposal distribution $q(x \rightarrow x') = q(x' | x)$
 - Calculate the acceptance rate $\alpha = \min\left(1, \frac{f(x') \cdot q(x|x')}{f(x) \cdot q(x'|x)}\right)$
 - Update $x^{(t)} = \begin{cases} x' & \text{with probability } \alpha \\ x^{(t-1)} & \text{with probability } 1 - \alpha \end{cases}$

Because this is a Markov chain, the chosen starting point $x^{(0)}$ is irrelevant because the algorithm converges on the target distribution no matter the starting point. If the starting point has low probability in respect to the target distribution, then the initial samples do not come from the equilibrium distribution and must be discarded until there is not equilibrium. We call the first B iterations eliminated from the sample the *burn-in period* for this reason.

Bayesian inference. The Metropolis-Hastings algorithm outlined above can be directly implemented to conduct Bayesian inference by substituting x with the unknown parameters θ and the target distribution $f(x)$ with the posterior distribution $f(\theta | y)$. Then, using the Bayes theorem, the acceptance rate becomes

$$\alpha = \min\left(1, \frac{f(\theta' | y) \cdot q(\theta | \theta')}{f(\theta | y) \cdot q(\theta' | \theta)}\right) = \min\left(1, \frac{f(y | \theta') \cdot f(\theta') \cdot q(\theta | \theta')}{f(y | \theta) \cdot f(\theta) \cdot q(\theta' | \theta)}\right)$$

Random-walk Metropolis-Hastings is a special case with symmetrical proposals $q(\theta' | \theta) = q(\theta | \theta')$ in which $q(\theta' | \theta) \equiv N(\theta, \sigma)$. In this case, the acceptance rate depends only on the posterior distribution and in the Bayesian inference case becomes

$$\alpha = \min\left(1, \frac{f(y | \theta') \cdot f(\theta') \cdot q(\theta | \theta')}{f(y | \theta) \cdot f(\theta) \cdot q(\theta' | \theta)}\right) = \min\left(1, \frac{f(y | \theta') \cdot f(\theta')}{f(y | \theta) \cdot f(\theta)}\right)$$

The random walk fixes the shape of the proposal distribution and requires specifying the value

σ . This value has a direct non-linear effect on the acceptance ratio (#accepted/#iterations). There is an optimal acceptance ratio that minimizes the auto-correlation and in most cases relies on the interval $[0.234, 0.44]$. To avoid the manual tuning of the scale parameter σ , we use the Robbins-Monro algorithm [16], which performs an adaptive scaling to achieve the desired acceptance ratio.

Proposition A.14 — The Robbins-Monro process. To achieve an acceptance ratio p in the random-walk Metropolis-Hastings algorithm, consider the following adaptive procedure:

$$\sigma^{(t+1)} = \begin{cases} \sigma^{(t)} \cdot \left(1 + \frac{1}{p \cdot t}\right) & \text{if } \theta' \text{ is accepted} \\ \sigma^{(t)} \cdot \left(1 - \frac{1}{(1-p) \cdot t}\right) & \text{if } \theta' \text{ is rejected} \end{cases}$$

in which the starting value $\sigma^{(1)} > 0$ has an arbitrary value.

Metropolis-Hastings using logarithms. The product of functions involves complicated expressions and accuracy problems. Therefore, it is usual to operate with logarithms. The logarithmic version of the acceptance rate in the Bayesian inference case becomes

$$\begin{aligned} \ln(\alpha) &= \min\left(\ln(1), \ln\left(\frac{f(y|\theta') \cdot f(\theta)}{f(y|\theta) \cdot f(\theta')}\right)\right) = \\ &= \min(0, \ln f(y|\theta') + \ln f(\theta) - \ln f(y|\theta) - \ln f(\theta')) \end{aligned}$$

The single-component Metropolis-Hastings algorithm consists of updates of sequentially univariate components using Metropolis-Hastings steps. One generated observation $\theta^{(t)}$ is obtained after updating all components of the parameter vector. Generally, the sequence of updating the elements of θ does not influence the convergence of the algorithm. Nevertheless, to ensure randomness, a random selection of the updating sequence is recommended. We use the notation $\theta_{\setminus j}$ to indicate the vector θ , excluding its j -th component [$\theta_{\setminus j} = (\theta_1, \dots, \theta_{j-1}, \theta_{j+1}, \dots, \theta_d)$].

Algorithm A.3 — Bayesian inference using Metropolis-Hastings. Let $f(y|\theta)$ be the likelihood function, with priors $f_j(\theta_j)$ and desired acceptance ratios p_j . Then the parameters distribution can be sampled using the following Metropolis-Hastings algorithm:

1. Set initial values $\theta^{(0)}$
2. Set initial values $\sigma^{(0)}$
3. For $t = 1, \dots, T$ repeat the following steps:
 - Set $\theta = \theta^{(t-1)}$
 - Shuffle the update sequence
 - For $j = 1, \dots, d$ repeat the following steps:
 - a. $\theta'_j = N(\theta_j, \sigma_j)$
 - b. Calculate the logarithm of the acceptance rate

$$\ln(\alpha) = \ln f(y|\theta'_j, \theta_{\setminus j}) + \ln f_j(\theta'_j; \theta_{\setminus j}) - \ln f(y|\theta_j, \theta_{\setminus j}) - \ln f_j(\theta_j; \theta_{\setminus j})$$
 - c. Simulate $u \sim U(0, 1)$
 - d. Update $\theta_j = \begin{cases} \theta'_j & \text{if } \ln(u) < \ln(\alpha) \\ \theta_j & \text{otherwise} \end{cases}$
 - e. Update $\sigma_j^{(t)} = \begin{cases} \sigma_j^{(t-1)} \cdot \left(1 + \frac{1}{p_j \cdot t}\right) & \text{if } \ln(u) < \ln(\alpha) \\ \sigma_j^{(t-1)} \cdot \left(1 - \frac{1}{(1-p_j) \cdot t}\right) & \text{otherwise} \end{cases}$
 - Set $\theta^{(t)} = \theta$

The simulated values from the posterior distribution have ergodic properties; however, they are not independent. Independent values can be obtained by plotting the autocorrelation function (ACF) and estimating the length, B , for which the autocorrelation is canceled. Then only the values whose index is a multiple of B are considered. Notably, the estimated parameters are not independent of one another; in fact, they form a multivariate distribution.

Example A.5 — Beta distribution calibration. We have a sample $\{y_1, \dots, y_{100}\}$ coming from a $\text{Beta}(2, 5)$ distribution. We want to perform Bayesian inference to determine the value of the parameters α and β using the Metropolis-Hastings algorithm presented above. The density of the Beta distribution is

$$f(x; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \cdot \Gamma(\beta)} \cdot x^{(\alpha-1)} \cdot (1-x)^{(\beta-1)}$$

The log-likelihood function, then, is

$$\ln f(y | \alpha, \beta) = \sum_{i=1}^{100} \ln \Gamma(\alpha + \beta) - \ln \Gamma(\alpha) - \ln \Gamma(\beta) + (\alpha - 1) \cdot \ln(y_i) + (\beta - 1) \cdot \ln(1 - y_i)$$

We know that $\alpha \in (0, 10)$ and $\beta \in (0, 30)$. Therefore, we consider that prior distributions $f_\alpha()$ and $f_\beta()$ are uniform distributions. Then,

$$\begin{aligned} \ln f_\alpha(\alpha; \beta) &= \ln\left(\frac{1}{10}\right) \\ \ln f_\beta(\beta; \alpha) &= \ln\left(\frac{1}{30}\right) \end{aligned}$$

We apply the algorithm A.3 with starting point $(\alpha^{(0)}, \beta^{(0)}) = (1, 1)$ and desired acceptance ratios $(p_\alpha, p_\beta) = (0.3, 0.3)$. Figure A.4 displays the results after 25,000 iterations. Notably, the first 500 iterations are not valid because the Markov chain has not achieved the equilibrium distribution. These must be considered in the burn-in period. The histograms of the sampled values approximate the parameters distribution that we are inferring.

A.7 Variance Reduction

The Monte Carlo method consists of sampling a distribution and averaging these values to approximate the mean of this distribution. The main drawback of this method is the low rate of convergence, on the order $1/\sqrt{N}$. Variance reduction methods are procedures used to increase the precision of the estimates of the Monte Carlo method. The main procedures are common random numbers, antithetic variates, control variates, importance sampling and stratified sampling.

A.7.1 Antithetic variates

We estimate $E(g(X))$ using the Monte Carlo method. Assume Y_1 and Y_2 as random variables with the identical distribution $g(X)$. This indicates that $E(g(X)) = E(Y_1) = E(Y_2) = E\left(\frac{Y_1+Y_2}{2}\right)$ and $\text{Var}\left(\frac{Y_1+Y_2}{2}\right) = \frac{\text{Var}(Y_1) + \text{Cov}(Y_1, Y_2)}{2}$. If Y_1 and Y_2 are negatively correlated, then the variance of N simulated values from $\frac{Y_1+Y_2}{2}$ is smaller than the variance of $2 \times N$ simulated values from Y_1 or Y_2 .

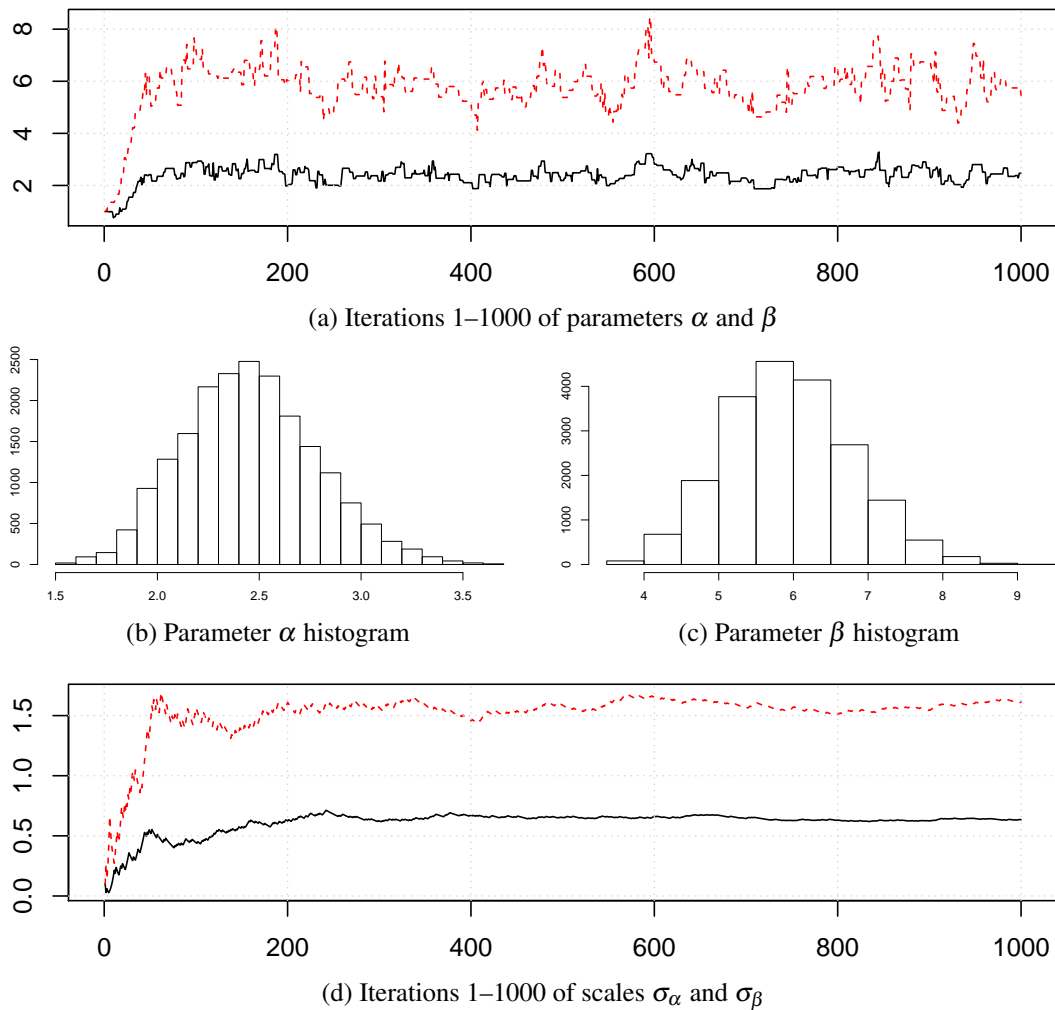


Figure A.4: Metropolis-Hastings algorithm example

Example A.6 In this example extracted from Wikipedia, we show how the antithetical method works and how it is used by CCruncher. We would like to estimate the following integral using the Monte Carlo method:

$$I = \int_0^1 \frac{1}{1+x} dx$$

In this case the random variable is $U \sim \text{Uniform}[0, 1]$, and its variate is $1 - U$. Listing A.1 computes the integral value using plain Monte Carlo, Monte Carlo with antithetic method, and Monte Carlo with antithetic variates but only to reduce the number of simulated values as CCruncher does.

	Sims	N	Mean	Variance
Exact = $\ln(2)$	-	-	0.693147	-
Monte Carlo	10000	10000	0.695685	0.019624
Antithetic	5000	5000	0.693404	0.000593
CCruncher	5000	10000	0.693099	0.019508

```

f <- function(x) { 1/(1+x) }

result = matrix(nrow=4, ncol=4)
colnames(result) = c("Sims", "N", "Mean", "Variance")
rownames(result) = 1:4

rownames(result)[1] = "Exact";
result[1,1] = 0
result[1,2] = 0
result[1,3] = integrate(f, lower=0, upper=1)$value
result[1,4] = 0

u = runif(10000)
rownames(result)[2] = "Monte_Carlo";
result[2,1] = length(u)
result[2,2] = length(u)
result[2,3] = mean(f(u))
result[2,4] = var(f(u))

u = runif(5000)
rownames(result)[3] = "Antithetic";
result[3,1] = length(u)
result[3,2] = length(u)
result[3,3] = mean((f(u)+f(1-u))/2)
result[3,4] = var((f(u)+f(1-u))/2)

u = runif(5000)
u = c(u, 1-u)
rownames(result)[4] = "CCruncher";
result[4,1] = length(u)/2
result[4,2] = length(u)
result[4,3] = mean(f(u))
result[4,4] = var(f(u))

result

```

A.7.2 Latin Hypercube Sampling (LHS)

This variance reduction technique belongs to the stratified sampling family. It consists of dividing the total probability range $[0,1]$ in N equispaced segments. Then we create a sample of size N picking a random value from each segment using a uniform distribution and applying the inverse transform sampling. Finally we shuffle the sample to obtain the desired distribution random values. When we have two independent random variables, we repeat the above procedure for each of them, obtaining the Latin Square effect: the generated random points are in a reticle in which each row and column has only one point. The procedure for three or more independent variables is identical, and it is called Latin Hypercube Sampling [4].

When the random variables are dependent, the previous schema does not work, and it is reformulated recurring to copulas and the rank statistic [31]. In this case, the number of bins affects the quality of the generated values, the larger the better.

Example A.7 We compare the estimated density of $N(0,1)$ using a sample of 100 simulated values without using LHS and a sample of 100 simulated values using LHS. Listing A.2 exposes the R script, and figure A.5 displays the results.

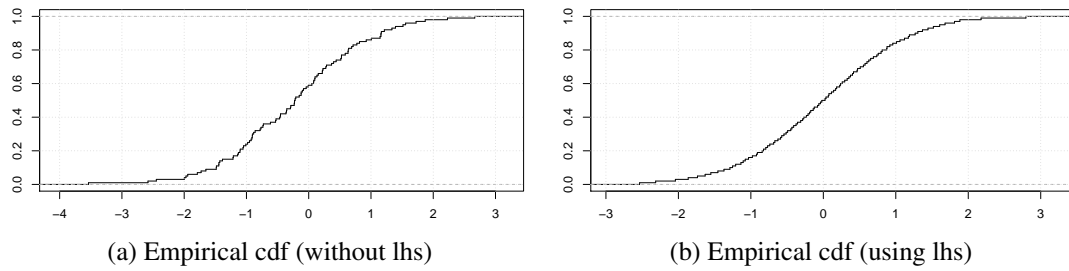


Figure A.5: Latin Hypercube Sampling example (1-dim)

Listing A.2: Latin Hypercube Sampling example (R script)

```

rlhs <- function(nblocks, icdf, ...)
{
  ret = rep(NA, nblocks)
  for(i in 1:nblocks) {
    u = runif(1, min=(i-1)/nblocks, max=i/nblocks)
    ret[i] = icdf(u, ...)
  }
  ret = sample(ret)
  return(ret)
}

x1 = rnorm(100, mean=0, sd=1)
cdf1 = ecdf(x1)
plot(cdf1, verticals=TRUE, pch=-1, ylab="prob", main="");

x2 = rlhs(100, qnorm, mean=0, sd=1)
cdf2 = ecdf(x2)
plot(cdf2, verticals=TRUE, pch=-1, ylab="prob", main="");

```

Bibliography

- [1] Carlo Acerbi and Dirk Tasche. Expected Shortfall: A Natural Coherent Alternative to Value at Risk. *Economic Notes*, 31(2):379–388, 2002.
- [2] Theodore W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley series in probability and mathematical statistics: Probability and mathematical statistics. Wiley, 1984.
- [3] Christian Bluhm, Ludger Overbeck, and Christoph Wagner. *An Introduction to Credit Risk Modeling*. Financial Mathematics series. Chapman & Hall/CRC, 1st edition, September 2002.
- [4] Phelim Boyle, Mark Broadie, and Paul Glasserman. Monte Carlo methods for security pricing. *Journal of Economic Dynamics and Control*, 21(8-9):1267–1321, June 1997.
- [5] Tim J. Brereton, Dirk P. Kroese, and Joshua C. Chan. Monte Carlo Methods for Portfolio Credit Risk. Technical report, Australian National University, College of Business and Economics, School of Economics, 2012.
- [6] Delphine Cassart, Carlos Castro, Ronny Langendries, and Thomas Alderweireld. Confidence Sets for Asset Correlation, February 2007.
- [7] Peter Crosbie and Jeff Bohn. Modeling Default Risk. Technical report, Moody’s KMV, December 2003.
- [8] Michel Crouhy, Dan Galai, and Robert Mark. A comparative analysis of current credit risk models. *Journal of Banking & Finance*, 24(1-2):59–117, January 2000.
- [9] Stefano Demarta and Alexander J. McNeil. The t Copula and Related Copulas. *International Statistical Review*, 73:111–129, 2005.
- [10] Paul Demey, Jean-Frédéric Jouanin, Céline Roget, and Thierry Roncalli. Maximum likelihood estimate of default correlations. *Risk*, 17:104–108, November 2004.
- [11] Paul Embrechts, Alexander J. McNeil, and Daniel Straumann. Correlation and Dependence in Risk Management: Properties and Pitfalls. In *Risk Management: Value at Risk and Beyond*, pages 176–223. Cambridge University Press, 2002.
- [12] European Central Bank. *The use of portfolio credit risk models in Central Banks*, number 64 in Occasional Paper Series, July 2007.
- [13] Rüdiger Frey and Alexander J. McNeil. Dependent defaults in models of portfolio credit risk. *Journal of Risk*, 6:59–92, October 2003.
- [14] Rüdiger Frey, Alexander J. McNeil, and Mark A. Nyfeler. Copulas and credit models. *Risk*, 14:111–114, October 2001.
- [15] Juan Carlos García Céspedes and David García Martín. The two-factor model for credit risk: A comparison with the BIS II one-factor model. Risklab seminar, January 2002.
- [16] Paul H. Garthwaite, Yanan Fan, and Scott A. Sisson. Adaptive Optimal Scaling of Metropolis-Hastings Algorithms Using the Robbins-Monro Process. *ArXiv e-prints*, June 2010.

- [17] Michael B. Gordy and Erik Heitfield. Estimating Default Correlations from Short Panels of Credit Rating Performance Data. Technical report, Federal Reserve Board, January 2002.
- [18] Christoff Gössl. Predictions based on certain uncertainties - a Bayesian credit portfolio approach. Working paper, July 2005. Hypo Vereinsbank AG.
- [19] Greg M. Gupton, Christopher C. Finger, and Mickey Bhatia. CreditMetricsTM – Technical Document. Technical report, RiskMetrics Group, 1997.
- [20] Stefan R. Jaschke. Quantile-VaR is the Wrong Measure to Quantify Market Risk for Regulatory Purposes, July 2001. EFA 2002 Berlin Meetings Discussion Paper.
- [21] Philippe Jorion. *Value at Risk*. McGraw-Hill, 1997.
- [22] John F. Kenney and E.S. Keeping. *Mathematics of statistics*, volume 2. Van Nostrand, 2nd edition, 1951.
- [23] Samuel Kotz and Saralees Nadarajah. *Multivariate t Distributions and Their Applications*. Cambridge University Press, 2004.
- [24] Alexander Kreinin and Marina Sidelnikova. Regularization Algorithms for Transition Matrices. *Algo Research Quarterly*, 4, March 2001.
- [25] David X. Li. On Default Correlation: A Copula Function Approach. *The Journal of Fixed Income*, 9(4):43–54, March 2000.
- [26] Kete Long. The fallacy of an overly simplified asymptotic single-risk-factor model. *Journal of Risk Model Validation*, 5(4):27–48, 2012.
- [27] Alexander J. McNeil, Rüdiger Frey, and Paul Embrechts. *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton University Press, Princeton, N.J., 2005.
- [28] Krishan Nagpal and Reza Bahar. Measuring default correlation. *Risk*, 14(3):129–132, March 2001.
- [29] Ioannis Ntzoufras. *Bayesian Modeling Using WinBUGS*. Wiley Series in Computational Statistics. Wiley, Hoboken, 2009.
- [30] Basel Committee on Banking Supervision. International Convergence of Capital Measurements and Capital Standards: A Revised Framework (Comprehensive Version). Technical report, Bank for International Settlements, June 2006.
- [31] Natalie Packham and Wolfgang M. Schmidt. Latin hypercube sampling with dependence and applications in finance. Working paper 15, Frankfurt School of Finance and Management, Centre for Practical Quantitative Finance (CPQF), October 2008.
- [32] Lutz Schloegl and Dominic O’Kane. A note on the large homogeneous portfolio approximation with the Student-t copula. *Finance and Stochastics*, 9(4):577–584, 2005.
- [33] Abe Sklar. *Fonctions de répartition à n dimensions et leurs marges*. Publications de l’Institut de Statistique de l’Université de Paris 8, 1959.
- [34] Nikola A. Tarashev. Measuring portfolio credit risk correctly: Why parameter uncertainty matters. *Journal of Banking & Finance*, 34(9):2065–2076, 2010.
- [35] Gerard Torrent-Gironella and Josep Fortiana. Simulation of High-Dimensional t-Student Copulas with a Given Block Correlation Matrix, June 2011. ASTIN Colloquium Madrid.

-
- [36] Oldrich Vasicek. Probability of Loss on Loan Portfolio. Technical report, KMV Corporation, February 1987.
- [37] Rand R. Wilcox. *Introduction to Robust Estimation and Hypothesis Testing*. Statistical Modeling and Decision Science. Academic Press, 2nd edition, December 2004.
- [38] Tom Wilde. CreditRisk+: A Credit Risk Management Framework. Technical report, Credit Suisse, 1997.