Collateral Damage

Creating a credit loss model incorporating a dependency between defaults and LGDs

Note: in this public version of the report, confidential information has been removed.

A master thesis project carried out at Rabobank Netherlands, Risk Management

Date

March 1st, 2016

Author

Tim de Wit
University of Twente

Supervisory committee

Martin van Jole
Rabobank

Reinoud Joosten
University of Twente

Manicka Pijnenburg
Rabobank

Berend Roorda
University of Twente

SUMMARY

This is a report of a study carried out at Rabobank Netherlands, Risk Management in order to investigate a possible dependency between defaults on loans and the size of losses resulting from these defaults. The goal of the study was to find evidence of such a dependency, and to design a way to model loss distributions that incorporates it. Each section of the report describes a stage of the project: first, the literature research, then acquiring a suitable dataset and choosing a modelling method, evaluating of parameter estimation methods, performing data analysis and finally performing a credit loss simulation.

We started our study by carrying out a literature review on the subject of PD-LGD dependency and modelling methods designed to incorporate it. We found several studies that linked LGD to macroeconomic variables similar to those that influence PD in the literature. We also found four different categories of modelling methods designed to model a dependency, differing in level of complexity and detail. These methods are listed in Section 3.2. The correct choice of modelling method depends on the quality of available data and the desired level of accuracy for loss distribution estimations. We used a dataset acquired from the Global Credit Data (GCD) consortium, because this was the best dataset available to us in terms of number of registered defaults and losses. Because our dataset is anonymised, we chose a relatively uncomplicated two-factor model (Witzany, A Two-Factor Model for PD and LGD Correlation, 2009) The model has three different parameters: one for defaults correlation, one for loss correlation, and one for the correlation between defaults and losses. The parameters are called *p*, *q* and *ω*, respectively.

After choosing our credit loss model, we described several different methods that can be used to estimate parameters. The two-factor model has three parameters: two that measure the amount of correlation in defaults and losses between assets, and one that governs the dependency between defaults and losses. We generated different sample datasets using known parameters and made estimates from those to test our estimation methods. The methods tested included the method of moments, maximum likelihood and Bayesian estimation. Our methods are described in detail in Section 4.3 of the report. Ultimately, we concluded that moments estimators based on the variance of default frequencies and average losses worked best to estimate parameters. We performed our data analysis using these estimators on our GCD dataset.

From our data analysis, we found that losses from defaults are most likely influenced in part by macroeconomic variables. We ran a Monte Carlo simulation to approximate a loss distribution using this fact and found that this systemic effect on losses increases the risk capital needed for credit portfolios such as the one we used by 15%. We did not find a correlation between losses and defaults in our dataset: the estimated size of the effect was close to zero. We did find a 'lagged' correlation, where the default frequency of one year correlated quite strongly with loss rates in the previous year. This effect does not have any effect on the loss distribution and therefore also not on economic capital levels. This is because the economic conditions from the previous year are known before the current year and used in PD predictions. This means that there is no shock effect requiring the allocation of extra capital.

The main findings from our study are the methods we described and tested, and the systemic effect on losses we found in our dataset. We recommend that more datasets from different credit portfolios are made suitable for data analyses like the one performed during our study, and that the influence of the economy on both defaults and LGDs are studied further to check our results and to look for dependencies in different credit markets.

ACKNOWLEDGEMENTS

Many people deserve acknowledgement for their contribution to this thesis report. First of all, the supervisory committee: Martin van Jole, Manicka Pijnenburg and Berend Roorda all contributed significantly with their feedback and suggestions. I would like to thank them for their continuous guidance and help.

Several colleagues at Rabobank helped me to access data sources which proved essential to this study. Their names, in alphabetical order, are Martin van Buren, Claire Kouwenhoven-Gentil, Marjon Ruijter and Mart Stokkers. Their taking the time to help me is much appreciated.

The Capital Planning and Modelling team and the other colleagues at Integrated Risk made writing this thesis easier by providing a friendly and professional atmosphere. This deserves an acknowledgement, too.

Finally, I thank Rabobank for providing me with the opportunity and the means to carry out my research, as well as a great experience.

TABLE OF CONTENTS

GLOSSARY

This is a list of abbreviations and technical terms that are used in this report, and their definition.

Term	Definition
Collateral	Something of value that is transferred to the bank in the event that a client defaults on his loan (see: Default), and which can then be sold to recover part of the outstanding balance.
Default	Different definitions are used, but generally, a person or company is in default if they are more than 90 days behind on payments and/or there is no realistic expectation of receiving a due payment from them.
Downturn/ Downturn LGD	A downturn is a period in which economic conditions worsen. This usually leads to an increase in defaults and higher losses on defaults. The downturn LGD is a concept used in regulations on financial institutions, defined as the average LGD in a downturn period experienced by the bank. See: LGD
EAD	Exposure at Default. The percentage of the amount originally loaned to a counterparty that has not been paid back at the time of default.
EC	Economic Capital. This is the capital that a bank holds to cover losses on loans. The Economic Capital is estimated in such a way that it can cover losses in a worst-case scenario.
EL	Expected Loss. The percentage of money loaned out that a bank expects to lose on a credit portfolio.
LGD	Loss Given Default. The percentage of a loaned amount that is lost in the event that the counterparty defaults. One minus the Recovery Rate (RR).
ODF	Observed Default Frequency. This is the percentage of firms in a population that have defaulted in a given year.
PD	Probability of Default. The probability (or estimated probability), conditional on available information, that a client will default on their loan within the next year.
RC	Regulatory Capital. The amount of capital a bank has to hold as a requirement from the financial regulator.
RR	Recovery Rate. The percentage of a loaned amount that can be recovered in the event of a default, usually by selling the collateral. One minus the Loss Given Default (LGD).
UL	Unexpected Loss. The loss in a worst-case scenario that the bank is protected from by its Economic Capital.

1. Research setup

1.1. Introduction

Rabobank is one of the three large banks in the Netherlands. As such, they are a major player in the Dutch credit markets. Modelling the bank's future credit losses and calculating the correct Regulatory and Economic Capital levels is one of the tasks of the Capital Planning and Modelling team, part of the Risk Management department of Rabobank. In accordance with Basel II guidelines, they use the Vašíček model to calculate the loss distribution of their credit portfolios. This model divides the risk for each counterparty into a systemic factor, common for all parties, and an idiosyncratic factor, unique to each one. This ensures that there is a certain correlation between Probabilities of Default (PDs) in scenarios, reflecting the cyclical nature of the economy. In the current regulatory modelling setup, the Loss Given Default (LGD) is neither correlated among counterparties nor with the default event. Usually, it is a fixed number to be multiplied with the indicator function of a default event. Banks are encouraged to make conservative estimates of the LGD, however, partly because there is some evidence that suggests PD and LGD are dependent (Basel Committee on Banking Supervision, 2005). The Capital Planning and Modelling team is interested in the effect such a dependency might have on the modelled probability distribution of credit losses. This master thesis report is intended to investigate the nature of the PD-LGD dependency and how it could affect credit loss distribution and Economic Capital.

The goal of the Capital Planning and Modelling team for this project is to find a way to model credit losses incorporating a dependency between PD and LGD. The main question is in what way the PD and LGD should affect each other in this model: whether a dependency between the two should be systemic or idiosyncratic or both and whether the correlation should be fixed or not, for example. Once a suitable modelling framework has been chosen, the team wants to use it to model credit losses on one of the bank's credit portfolios. They are interested in the impact the correlation has on the Economic Capital calculation: how strongly does having a dependency increase the capital buffers indicated by the model?

In this report, we describe the research project designed to answer the Capital Planning and Modelling's question. Each section of the report is dedicated to a stage of the project. In the final section, we draw conclusions from our observations and do recommendations for future actions.

1.2. Research questions and Methods

It is useful for direction to distil the questions we want to answer into one main research question, and then to outline the knowledge needed to answer it into several sub-questions. These can then be answered in order to find the solution to the main question. The main question to be answered is the following one:

"How can we create a credit loss model that incorporates a dependency between PD and LGD, which we can use for capital calculations on Rabobank credit portfolios?"

Different kinds of knowledge are needed to answer the question, which makes it necessary to do several different kinds of research. We must find out what is known about the dependency between PD and LGD, and what modelling methods exist that incorporate it. We must think of a way to evaluate potential solutions, and we must investigate the impact of implementing a solution. To make this large research process easier to oversee, we can divide the project into several phases, in which we must answer one or more sub-research questions. The four phases of this research process are literature research, normative research, parameter estimation and implementation/evaluation. In the first phase, we gather and compile knowledge from scientific literature on the subject of PD-LGD dependency, and modelling methods that incorporate it. In the second phase, we do a requirements analysis for the credit loss model we want to create. The third phase of the project involves analysing historic default data and estimating the parameters to be used in our model. In the final phase, we implement our modelling solution and evaluate the impact of a PD-LGD dependency on capital estimates.

The questions to be answered, in order, are as follows:

1. What is known about the nature of the dependency between PD and LGD? ***(Literature)***
2. What kind of modelling methods exist that are designed to incorporate a PD-LGD dependency? ***(Literature)***
3. Which requirements can be identified for a credit loss model for EC calculations***? (Normative)***
4. Which modelling method is most suited to the needs of the Capital Planning and Modelling team? ***(Normative)***
5. Which parameters should be used when modelling PD and LGD variables, and what should be their value? ***(Estimation of parameters)***
6. What is the effect of using a PD-LGD correlated model on Economic Capital recommendations? ***(Implementation/Evaluation)***

The first two questions are answered by way of literature research. We compile a review of the available knowledge on the dependency between PD and LGD: the empirical evidence that indicates that there is a dependency and estimates on the magnitude of the correlation. We also make a list of modelling methods for the LGD that include a correlation with the PD, including their advantages and disadvantages and the parameters that need to be estimated. These are the alternatives from which we choose a modelling method in Section 3.

The next two questions, three and four, are answered through an analysis of the available data and the needs of the Capital Planning and Modelling team. We determine the level of detail that is required from a credit loss model for capital calculations, and we choose a way to attain this level using the available data. With the selection of a modelling method, the normative stage of the project is concluded.

We answer the fifth question through data analysis and the modelling of the credit default process. We try to emulate the real behaviour of this process through the choice of parameters and estimation of their value by an approximation method suited to our chosen modelling method. Then, we implement the model and estimate the probability distribution of credit losses.

The last question is answered by comparing our results from question five with results from conventional credit loss models. We analyse the impact of the dependency between PD and LGD on the credit loss probability distribution and on capital levels.

Each of the four research phases makes up one section of the report. In the final section, we discuss our findings and draw conclusions about their implications for practice and further research.

2. Literature

This section of the report describes our literature research into the dependency between PD and LGD. The first part of the section is a review on the evidence available in the literature on a dependency between PD and LGD, and the magnitude of this correlation. The second part is a list of published credit risk modelling methods that incorporate a correlation or dependency between PD and LGD.

2.1. literature on PD-LGD dependency

2.1.1. Introduction: Credit Loss

Credit loss is defined as the loss a bank takes on its loans. When banks issue loans to persons or companies, they anticipate that some of those loans will not be paid back in full. Due to a variety of reasons (unemployment, bankruptcy), some counterparties will default on their loans. To compensate losses from these defaults, banks raise the price of loans dependent on the risk of default posed by a counterparty. The higher the estimated risk for a counterparty, the higher the interest that counterparty has to pay. The Expected Loss is the expected value of the amount of money the bank will lose on a portfolio of loans.

To estimate the Expected Loss, banks create models to estimate two parameters: the PD and the LGD. The PD is the Probability of Default, which is the probability that a client of the bank will default on his loan in a certain period, usually a year. A default is taken to mean that the client is more than 90 days behind on their interest payments. The LGD is the Loss Given Default. This is the percentage of the money loaned to the client that the bank will lose if the client defaults. For loans with a collateral, this loss tends to be low, as the proceeds from the sale of the collateral can cover a large part of the Exposure at Default (EAD). This is the percentage of the loaned amount that has not been paid back at the time of default. For loans without a collateral, the loss tends to be higher, as the client may not have anything of value to be sold.

The loss on a credit portfolio is uncertain, since it is not known in advance whether clients will pay back their loans or not. For this reason, banks hold risk capital to compensate for any losses. Financial regulators have strict guidelines on the amount of capital that has to be held for certain classes of loans. The amount of capital the bank has to hold under these guidelines is called the Regulatory Capital (RC). Banks will often choose to keep more capital than that, though, because they want to be on the safe side. A bank will define the worst-case scenario they want to be prepared for (say, an event that happens once every 10,000 years) and hold an amount of risk capital sufficient to survive this scenario. This amount is called the Economic Capital (EC).

2.1.2. Credit Risk Capital

Banks are required to calculate their regulatory capital (RC) buffers for credit portfolios as the unexpected loss (UL) minus the expected loss (EL) according to the Basel II approach (Basel Committee on Banking Supervision, 2006):

$$RC = UL - EL = DR_{99.9\%} \times LGD \times EAD \tag{2.1}$$

Here, $DR_{99.9\%}$ represents the default rate at the quantile 99.9%, which is multiplied with the LGD and the exposure at default (EAD). This framework assumes that the average LGD and EAD are fixed and independent from the default rate. The DR is calculated as a function of PD and the asset correlation ρ (which is set by the regulation for each asset class). The basis for this regulatory formula is the assumption that the events of default are driven by normally distributed variables, which are composed of a single normally distributed common factor and individual independent normally distributed idiosyncratic factors (Vašíček, 1987). These variables can be represented in the following way:

$$X_i = Y\sqrt{\rho} + Z_i\sqrt{1-\rho} \tag{2.2}$$

Where {Y,Z_1,Z_2,…} ~ N(0,1) are independent and, consequently, {X_i} is a set of standard normal variables with pairwise correlations of ρ. The event of a default of counterparty *i* is then equal to the event that $N^{-1}(X_i)<PD_i$.

In the described regulatory framework, the LGD is assumed to be independent from default rates. Banks may use a fixed percentage as the average LGD when calculating credit loss, but should adjust this percentage to reflect economic downturn conditions (Basel Committee on Banking Supervision, 2005). Traditionally, most other credit risk models have also assumed an LGD independent from systemic factors and therefore, from default rates (Altman, Default Recovery Rates and LGD in Credit Risk Modeling and Practice: An Updated Review of the Literature and Empirical Evidence, 2006). However, many recent papers have presented evidence that the losses from defaults are in fact volatile, and that they tend to go up when the default rate rises (Altman, Default Recovery Rates and LGD in Credit Risk Modeling and Practice: An Updated Review of the Literature and Empirical Evidence, 2006).

2.1.3. Evidence of a PD-LGD correlation in public data

The idea that the default rate and the average LGD could be correlated was pioneered by Frye (Repressing Recoveries, 2000), which examined data on bank loans, corporate bonds and sovereign bonds from the Moody's Default Risk Service database and found that recovery rates ($RR = 1 - LGD$) were significantly lower in high-default periods. The data used is from the period of 1983 to 1998, which contained two years of economic downturn: 1990 and 1991. During these years, the average LGD on a Moody's portfolio rose by 45%. On bank loans, Frye suggests that the LGD nearly doubles when a depressed state occurs. This suggests that the LGD moves together with many of the same economic variables that the PD does. Importantly, though, the actual PD of a client did not seem to influence the level of systematic risk. Both high- and low-quality loans saw approximately the same increase in LGD levels during depressed periods.

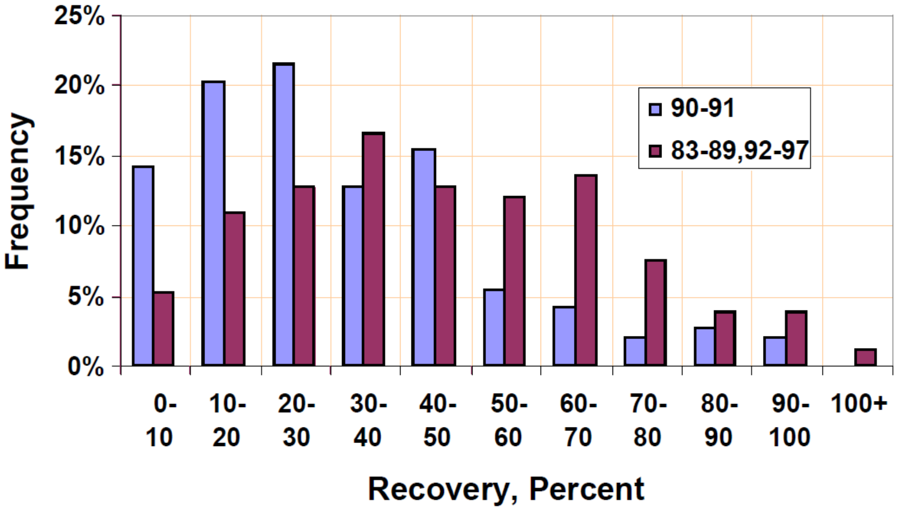


FIGURE 2.1. RECOVERY IN HIGH- AND LOW-DEFAULT YEARS (FRYE, DEPRESSING RECOVERIES, 2000)

A later article (Frye, A False Sense of Security, 2005), using the same dataset with the addition of 1998 through 2001, confirmed earlier results for separate industries and separate loan quality categories, all of which showed a significant PD-LGD correlation. A study on corporate bond defaults between 1982 and 2001 (Altman, Brady, Resti, & Sironi, 2005) found that using a linear regression model, 51% of variance in LGD rates could be explained by the default rate. Using logistic and power models, they could explain over 60% of the variance from the default rate. This is strongly indicative of the notion that PD and LGD are influenced by the same factors, at least in the corporate credit segment. A different study hypothesised that the correlation could be explained by the *fire-sale effect* (Acharya, Bharath, & Srinivasan, 2007): in distressed industries, even healthier companies are less liquid than usual and cannot afford to spend as much money on defaulted competitors' assets. Due to a higher number of defaults, there is also a higher supply of assets from defaulted companies.

This will lead to low sales prices for these assets, especially in highly asset-specific industries such as energy and telecommunications. Acharya, et al. (2007) found a decrease in recovery rate of 50% in these industries during distress periods, using data from the S&P Credit Pro database.

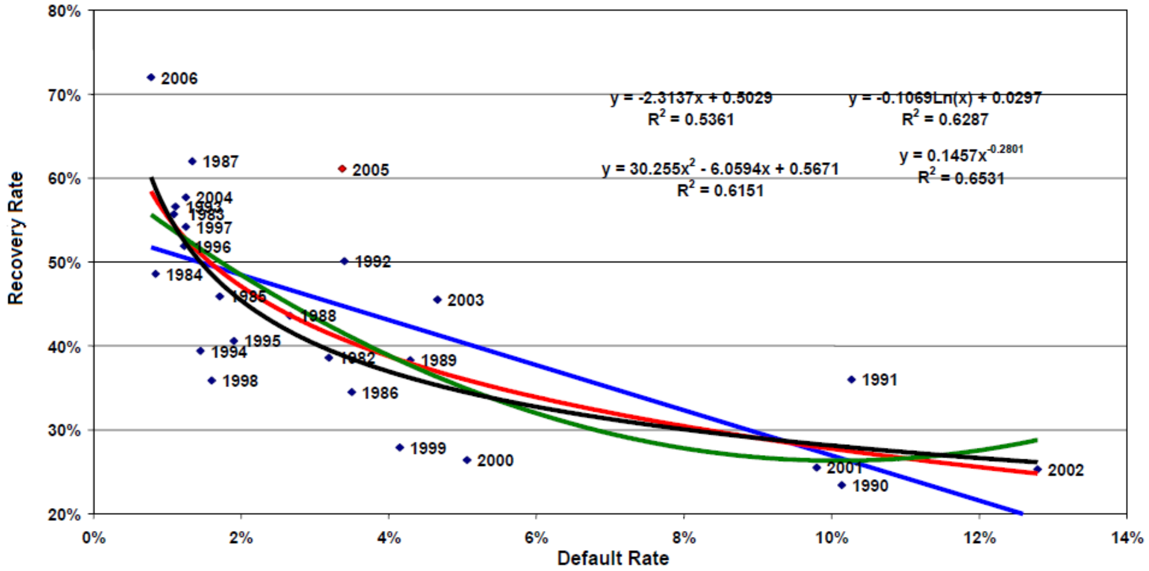


FIGURE 2.2. UNIVARIATE MODELS, RESULTS OF REGRESSIONS CARRIED OUT BETWEEN THE RECOVERY RATE AND THE DEFAULT RATE (ALTMAN, BRADY, RESTI, & SIRONI, 2005)

2.1.4. Evidence from banking data

More general evidence for a PD-LGD dependency in credit losses was presented (Caselli, Gatti, & Querci, 2008), which analysed data on defaults from five Italian banks representing over 70% of the credit market in Italy. This article examined 11,649 defaults of both consumers and small or medium-sized enterprises (SMEs) looking for explanatory variables for the LGD rates and analysing the distribution of LGD. They found that the LGD over the entire market had a bimodal distribution, with a high probability of the LGD being either 1 or 0.

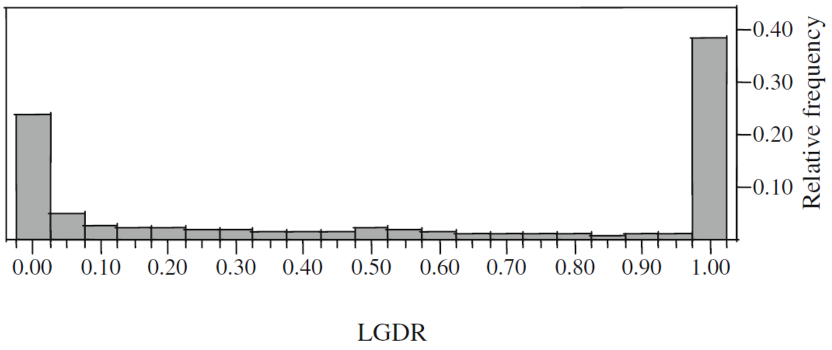


FIGURE 2.3. HISTOGRAM OF THE LGD RATE IN THE ITALIAN CREDIT MARKET (CASELLI, GATTI, & QUERCI, 2008)

When split into different loan types, however, the credit LGD distributions are not so bimodal and usually tend to a median close to either 0 or 1, with relatively few LGDs on the other side (Caselli, Gatti, & Querci, 2008). Real estate-secured loans to households, for example, tend to have a very low LGD, with a mean of 0.15 and a median of 0.02. Credit card loans, on the other hand, have a median LGD of 1.00 and an average of 0.79. In loans to SMEs, this effect is less prominent, but still present.

Caselli, Gatti, & Querci (2008) also tried to find macroeconomic variables with which to predict LGD rates, and found that for consumer loans, the default rate, along with household consumption and the unemployment rate, formed the predictor with the lowest error. For SME loans, the best model incorporated the GDP growth rate and the number of employed people. A later study using banking data from the Czech Republic confirmed that unemployment and consumption are good predictors of consumers' LGD rate (Belyaev, Belyaeva, Konečný, Seidler, & Vojtek, 2012). The authors did not test the default rate as a predicting factor, choosing to predict both loss and default using macroeconomic variables. In corporate loans, they found that GDP growth is a good predictor of the LGD rate.

A recent study used default data from the Global Credit Data Consortium, a collaboration of many banks sharing default data, to investigate cyclicality in credit losses (Keijsers, Diris, & Kole, 2015). Using stylised sets of macroeconomic variables in time series, they found that both LGD and PD move together with factors representing economic conditions. They also confirmed that the bimodality of the LGD persists in recent banking data (June 2014).

2.1.5. Conclusion

While it has been accepted for a long time that default rates are cyclical and dependent on the current state of the economy (Duffie, Saita, & Wang, 2007), the dependency of the LGD on macroeconomic factors and therefore, its co-dependency with the default rate, has long been ignored in credit risk management practice (Altman, Default Recovery Rates and LGD in Credit Risk Modeling and Practice: An Updated Review of the Literature and Empirical Evidence, 2006). However, evidence from both bond and retail banking data (as shown in previous paragraphs) has indicated that PD and LGD are co-dependent. This means that credit loss models which ignore this correlation will underestimate credit losses (Witzany, 2013; Caselli, Gatti, & Querci, 2008). A simulation of loan portfolios has shown that a 30% increase in Economic Capital could be needed to compensate for this effect (Altman, Brady, Resti, & Sironi, 2005), but other publications using different parameters have different estimates, both higher and lower (Altman, Default Recovery Rates and LGD in Credit Risk Modeling and Practice: An Updated Review of the Literature and Empirical Evidence, 2006). Since the LGD distribution differs highly per asset class (Caselli, Gatti, & Querci, 2008), it is hard to make general statements about the impact of a PD-LGD correlation on appropriate capital levels. This impact will depend highly on the specific properties of the loan portfolio considered.

2.2. Modelling methods

This part of the section is a list of published modelling methods that deal with a dependency between PD and LGD and the ways that they work. There are two main categories: models based on abstract risk factors and models based on concrete risk factors. Models in the first category are usually some form of variation on the Basel framework (Basel Committee on Banking Supervision, 2006), which involves the defaults of counterparties as a set of Bernoulli variables determined by a common risk factor and many idiosyncratic risk factors. Models in the second category make predictions of PD and LGD using information on the counterparties, on both individual and macroeconomic levels. They typically use a regression based on this information, along with a mathematical transformation, to make predictions.

2.2.1. Single Factor Models

The credit capital model by Frye et al. (2000) is, according to the authors, the first credit model to incorporate systematic LGD risk. It works using one common risk factor, on which both default and LGD depend. Each counterparty *i* is assumed to have an exposure of \$1.00. After one year, the value of their collateral is a random number determined by three parameters:

$$C_i = \mu_i(1 + \sigma_i Y_i) \tag{1.3}$$

$$Y_i = q_i X + \sqrt{1 - q_i^2}\,\zeta_i \tag{2.4}$$

Where X and {ζ_i} are independent and standard-normally distributed. {μ_i} and {σ_i} are fixed variables that reflect individual expectations and volatilities. This means that the Collateral C_i is a normally distributed variable. The defaults of counterparties, meanwhile, are also determined by the common risk factor X:

$$A_i = p_i X + \sqrt{1 - p_i^2}\, \xi_i \tag{2.5}$$

$$D_i = \begin{cases} 1, & if\ \ A_i > N^{-1}(1 - PD_i) \\ 0, & otherwise \end{cases} \tag{2.6}$$

Where {D_i} are the default indicators of counterparties, {PD_i} are the individual probabilities of default and {ξ_i} is a series of mutually independent standard-normally distributed variables. The actual credit loss from the portfolio, in this model, is as follows:

$$Loss = \sum_i D_i * \max(0, 1 - \ C_i) \tag{2.7}$$

The publication of this model, which rendered unexpected loss estimates over two times higher than models like CreditMetrics, prompted a lot of research into the dependency between PD and LGD (Altman, Default Recovery Rates and LGD in Credit Risk Modeling and Practice: An Updated Review of the Literature and Empirical Evidence, 2006). However, it was also criticised, initially because the normality of the Collateral allows for a negative loss given default. An alternative Single Factor Model was published three years later with a lognormally distributed LGD (Pykhtin, 2003). This model is similar to the model from Frye (2000): defaults are governed according to Equations 2.5 and 2.6, and the loss is determined by Equation 2.7. The difference is in the LGD, which can be written as follows:

$$LGD_i = \exp(\mu_i + \sigma_i Y_i) \tag{2.8}$$

$$Y_i = q_i X + r_i \xi_i + \sqrt{1 - q_i^2 - r_i^2}\, \zeta_i \tag{2.9}$$

Where $\{X, \xi_i, \zeta_i\} \sim \mathrm{N}(0,1)$ and μ_i and σ_i are individual parameters for each borrower *i*. As we can see from Equation 2.9, individual PD and LGD both depend on the common risk factor X, which determines the state of the economy. The individual LGD then also depends on the idiosyncratic risk factor ξ_i, which determines the PD along with X, causing an extra correlation between the individual PD and the LGD. Finally, the individual factor ζ_i influences only the LGD. Because of the exponential function in Equation 2.8, the LGD can no longer be negative in this model.

The central assumption of single-factor credit models is that both the default rate and the LGD interact with the state of the economy in the same way through some correlation effect, which is modelled by the variable X and a correlation factor. In diversified portfolios, this causes the default rate and the average LGD to correlate almost perfectly with one another (Witzany, A Two-Factor Model for PD and LGD Correlation, 2009). However, analysis of real default data has shown that this one explanatory factor is not very realistic and that defaults and LGD may be influenced by different components of the economic cycle (Altman, Brady, Resti, & Sironi, 2005).

A solution to amend the problem of the PD-LGD correlation was suggested in Syrkin & Shirazy (2013): instead of simulating defaults with a Gaussian variable, they suggest simulating loss rate, or D x LGD, directly. In their example, they use a binary Boolean LGD, resulting in a Vašíček distribution of loss. However, if a continuous LGD variable is used, the resulting loss distribution can no longer be analytically determined. An LGD distribution must then be estimated from which to draw losses. The authors call their solution “an effective and flexible tool for loss estimations from the practical standpoint in particular”. In their conclusion, however, they say that a study of how to relate inputs to existing empirical data, is needed to give their model justification.

2.2.2. Two-Factor or n-factor Model

As an alternative to the Single Risk Factor models, researchers have published models with two or more common risk factors to model the defaults and LGDs of credit portfolios. These two factors are then correlated to reflect the fact that they are in part based on the same economic cycle; only on different aspects of it. One factor determines the defaults, usually according to the Basel framework, while the other factor determines the losses using some distribution.

Jiři Witzany (2009) criticises the one-factor models suggested in Frye (2000) and Pykhtin & Shirazy (2013) and proposes to extend their model to amend the high correlation between PD and LGD. The "natural choice" for such an extension (Witzany, A Two-Factor Model for PD and LGD Correlation, 2009) is defined in the following way:

$$A_i = p_i X + \sqrt{1 - p_i^2}\xi_i \tag{2.10}$$

$$B_i = q_i\left(\omega X + \sqrt{1-\omega^2}Y\right) + \sqrt{1 - q_i^2}\zeta_i \tag{2.11}$$

Where X, Y, {ξ_i}, {ζ_i}~N(0,1) are independent. {p_i} and {q_i} are correlation parameters depending on asset class and ω is a fixed parameter on [-1,1] determining the correlation between the PD and LGD systematic factors. Defaults are determined by Equation 2.6, just like in the earlier one-factor models, while the LGD is determined by a probability distribution:

$$LGD_i = F_{LGD}^{-1}\big(N(B_i)\big) \tag{2.12}$$

Where F_{LGD}^{-1} is the inverse of some distribution function. Witzany (2009) suggests fitting a beta distribution from default data or using a kernel-smoothed empirical distribution (where a large set of historical observations is smoothed and then used to approximate a distribution). The study uses the *ksdensity* function in Matlab to approximate a cumulative distribution function for LGDs. Using historic default data from a portfolio of unsecured retail loans by a Czech bank, it is estimated that ω should probably be around the 0.1-0.2 mark, but there is enough data to say this with confidence. One advantage of this two-factor model is that it is quite easy to implement and does not require data with too much detail to fit it to a real portfolio. However, the stylised Gaussian common factors are probably not completely realistic in emulating economic cyclicality.

Eckert, Jakob, & Fischer (2015) extends the two-factor model by Witzany to a 4-factor model, splitting up the LGD into three different parts to reflect the unexposed amount, the secured amount and the unsecured amount of the loan. The three parts of the loan are determined by correlated Gaussians (as in Equation 2.11) and are drawn from beta distributions (similar to Equation 2.12). This model provides the benefit of slightly more realism, but it requires estimating 12 covariates and doubles the number of variables, which makes it convoluted. It also requires data on exposures at default and collateral value to make the estimations, which makes it harder to implement.

2.2.3. Concrete Factor Models

Rather than simulating the effects of the economic cycle with one or more abstract variables, it is also possible to use actual macro-economic and account-level parameters to model PD and LGD. If we do this, the dependency between PD and LGD does not stem from correlations between risk factors, but from the way they are derived from the same parameters. Models that fall into this category use macroeconomic parameters like LGD growth, consumption rates and unemployment, which are good predictors of overall default rates and average LGD (Belyaev, Belyaeva, Konečný, Seidler, & Vojtek, 2012), along with counterparty-specific parameters like Income-to-Loan and Loan-to-Value, which are good predictors of individual default and LGD respectively (Chava, Stefanescu, & Turnbull, 2011).

The credit loss model by Chava, Stefanescy and Turnbull (2011) uses two different methods for determining PD and LGD. However, both methods are based on a 1*K vector of variables $\mathbf{X}_i(t)$ assigned to each counterparty *i* at time *t*. The variables used include the return on the S&P 500, credit spread, treasury yield, term spread and the log of total defaulted debt on the macro level, and firm size, asset return, debt to assets, and volatility on the firm level. The elements of the matrix are adjusted so that they fit a standard-normal distribution. They then use an exponential variable to govern default events:

$$PD_i = 1 - \exp(-\exp(\boldsymbol{X}_i\boldsymbol{\beta})) \tag{2.13}$$

Where **β** is a K*1 vector of coefficients determined by regression. The recovery rate, which is equal to one minus the LGD, is estimated by a Logit or Probit transform:

$$R_i = 1 - LGD_i = N(\boldsymbol{X}_i\boldsymbol{\beta}_r) \tag{2.14}$$

$$R_i = 1 - LGD_i = \frac{1}{1 + \exp(\boldsymbol{X}_i\boldsymbol{\beta}_r)} \tag{2.15}$$

Where $\boldsymbol{\beta}_r$ is a vector of regression coefficients and N is the cumulative distribution function of the normal distribution. When comparing the model to actual loss data, the study finds that the model performs well in predicting defaults and recovery rates. Most other models based on concrete factors use similar methods based on regression, followed by some transformation, to determine PDs and LGDs (Belyaev, Belyaeva, Konečný, Seidler, & Vojtek, 2012). However, some research has suggested that LGDs might be better predicted by regression trees or neural network algorithms (Qi & Zhao, 2011).

After individual PD's and LGD's have been determined for a portfolio, a loss distribution has to be estimated. This is often done by Monte Carlo simulation (Meng, Levy, Kaplin, Wang, & Hu, 2010), which requires some way of correlating defaults and LGD's across lenders. This can be done on a portfolio level and on an idiosyncratic level. One proposed method entails introducing a random 'frailty' factor F, drawn from a beta distribution, that influences defaults across the portfolio (Chava, Stefanescu, & Turnbull, 2011):

$$PD_i = 1 - \exp(-\mathrm{F}\exp(\boldsymbol{X}_i\boldsymbol{\beta})) \tag{2.16}$$

Defaults can then be drawn using a Bernoulli process, while individual LGDs are kept fixed.

Meng, et al. (2010) suggest adapting the model to incorporate a common risk factor across idiosyncratic risk factors. Recall that in Witzany's two-factor model (Equations 2.10 and 2.11), A_i was the risk driver for defaults and B_i for losses. Using Meng's model, the drivers are defined as follows (Meng, Levy, Kaplin, Wang, & Hu, 2010):

$$A_i = p_i\boldsymbol{X}_i\boldsymbol{\beta} + \sqrt{1 - p_i^2}\xi_i \tag{2.17}$$

$$B_i = q_i\boldsymbol{X}_i\boldsymbol{\beta}_l + \sqrt{1 - q_i^2}\left(\omega\xi_i + \sqrt{1 - \omega^2}\zeta_i\right) \tag{2.18}$$

Where $\boldsymbol{\beta}_l$ is a vector of regression coefficients for the LGD and $\{\zeta_i\}$ and $\{\xi_i\}$ are individual risk factors. All variables are standard-normally distributed, which makes A_i and B_i standard-normally distributed as well. The defaults are then determined as in Equation 2.6, while LGD's are determined as in Equation 2.12. This model is similar to both the abstract two-factor model and concrete models, and it contains correlation both on portfolio and individual levels.

The fact that concrete risk factor models use actual data to make predictions on defaults and recoveries, gives this class of models legitimacy. They are also useful because they make loss predictions on the individual level, rather than the portfolio level, which means they could also be used to assess the risk of one particular lender.

However, the need for large amounts of data on both macroeconomic and individual levels is also a major disadvantage of these modelling methods.

2.2.4. LGD Distributions

Many of the authors cited in this section have either drawn losses given default from, or fitted them to, various distribution functions. In this section, we name the used probability distributions for drawing LGDs and any relevant points on their suitability.

- **Normal distribution** – the normal distribution is the go-to option for data analysts and modellers in many disciplines, because it is a ubiquitous distribution. It has been used for fitting the LGD by Rösch and Scheule (2009), as well as by Keijsers, Diris, & Kole (2015) while analysing banking data. It was also used to render LGDs (Frye, et al., 2000) in a single-factor credit loss model. Keijsers, Diris, & Kole (2015) used a combination of two normal distributions, one for good loans and one for bad loans, leading to a bimodal distribution. A drawback of the normal distribution is that it cannot be restricted to a closed interval, and if used to render LGDs some of them will be negative or greater than one.
- **Lognormal distribution** – As a response to the possible negativity of normally distributed LGDs, one can use a lognormal distribution to model LGDs (Pykhtin, 2003). Variables drawn from this distribution will always be greater than 0, but they can still be larger than 1. Due to insights into the bimodality of the LGD, however, the lognormal distribution has not been used in recent publications.
- **Beta distribution** – The beta distribution is a family of distributions defined on the [0, 1] interval, which can have many different shapes. Its domain makes it useful for random behaviour of percentages and proportions, such as the LGD. The beta distribution was suggested as an analytical alternative to an empirical distribution by Witzany (2009). By adjusting parameters, it can be made to have one or two modes.

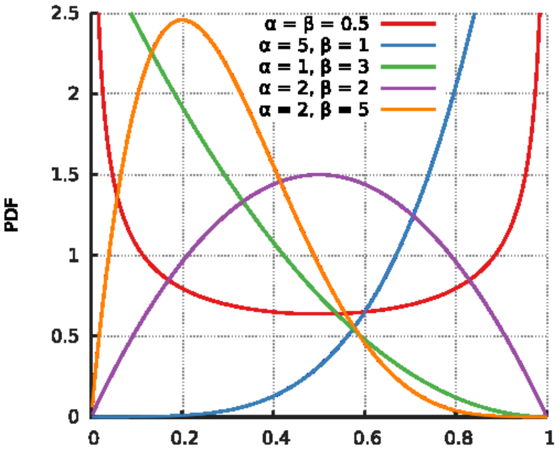


FIGURE 2.4. VARIETIES IN DENSITY FUNCTIONS OF THE BETA DISTRIBUTION (WIKIPEDIA, 2014)

- **Empirical distribution –** The empirical distribution is a collection of historical instances of a random variable from which one can draw. When the distribution of a variable is unknown or complex and a lot of historical data is available, this can be a good choice of distribution. The only assumption one has to make when using an empirical distribution is that the variable being drawn has a constant unchanging distribution. This makes the empirical distribution 'lighter' on assumptions than analytical distributions.
- **Bernoulli distribution & Beta distribution mixed –** Noting that historical data on LGDs contain a large percentage of extreme outcomes 0 and 1, Calabrese (2014) suggests using a mixture of a Bernoulli and a Beta distribution to model them:

$$F_{LGD|S}(y) = \begin{cases} p_0^s & y = 0 \\ p_0^s + (1 - p_0^s - p_1^s)F_{B|S}(y) & y \in (0,1) \\ 1 & y = 1 \end{cases} \tag{2.19}$$

Where p_0^s is the probability of LGD being 0, given a certain state of the economic cycle, $p_1^s$is the probability of LGD being 1 given that state, and $F_{B|S}$(y) is a Beta distribution function. Probabilities and the distribution should be fitted by maximum likelihood estimation to historical data. This distribution, while convoluted, can provide variables distributed most similarly to real data (as seen in Figure 2.3) out of all suggested solutions.

2.2.5. Conclusion

Credit loss models incorporating a dependency between the default rate and the loss given default come, essentially, in two different varieties: models based on abstract risk factors and models based on concrete risk factors. Models using concrete factors make predictions about default probabilities and loss using both relevant macroeconomic variables and individual variables. Models using abstract factors try to simulate observed behavior of the loss on an entire portfolio without trying to make accurate predictions on the individual level. While using concrete factors gives a credit loss model more legitimacy, it requires a lot of data on both macroeconomic and individual levels, as well as an extensive conceptual model of the default process and the factors that influence it. Models using abstract factors require less data to be calibrated, but their results may not be as realistic as those of models using concrete factors. The choice of a modelling method for a simulation study, therefore, will depend greatly upon the level of detail of available data, and on the level of realism the study is meant to achieve. In the next section, we determine which modelling method is best considering the available data and our specific needs.

3. Choice of modelling method

This section of the report describes the decision process which was used to decide on a modelling method for our impact study on the dependency between PD and LGD. The decision involved the level of detail in the available loss data, the data requirements of the selected modelling methods, and the needs of the Capital Planning & Modelling team for this study. We analyse the quality and depth of the available data and determine for which models it is suited. Then, we select the most suitable method based on this, and other relevant factors.

3.1. Available Data

The quality and the level of detail of data are a major factor in the decision process, as they determine which model parameters can be estimated. In this section, we describe the sources of credit loss data available to Rabobank. One of these sources is Global Credit Data (GCD), a consortium of banks which pools credit data from all its participants. The other sources are internal to Rabobank.

The rest of this section is confidential and not publically available. The chosen data source is the GCD dataset.

3.2. Properties of Modelling Methods

The various models described in Section 2 have varying needs in terms of data depth and accuracy, as well as varying levels of realism and applicability. While choosing a modelling method to proceed with, we need to find a balance between the realism a model offers and the requirements it has in terms of data. In this section, we describe the data requirements of each model listed in Section 2.2, as well as the realism it offers.

3.2.1. Data Requirements

In Section 2.2, we described four different basic models which can be used for credit loss modelling: the one-factor abstract model published by Frye, et al. (2000), the two-factor model by Witzany (2009), the macroeconomic-factor model with a frailty variable (Chava, Stefanescu, & Turnbull, 2011) and the macroeconomic-factor model with idiosyncratic correlation (Meng, Levy, Kaplin, Wang, & Hu, 2010). Each factor has number of parameters that need to be estimated to use it for simulations. These parameters are listed in Table 3.1.

Model	Parameters
Frye, et al. (2000)	p_i, q_i
Witzany (2009)	p_i, q_i, ω
Chava, et al. (2011)	$\boldsymbol{\beta}$, $\boldsymbol{\beta}_r$, $F_F(x)$
Meng, et al. (2010)	$\boldsymbol{\beta}$, $\boldsymbol{\beta}_l$, p_i, q_i, ω

TABLE 3.1. MODELLING METHODS AND PARAMETERS TO BE ESTIMATED

We can see from the table that generally, adding realism to a model has the consequence of adding more parameters that need to be estimated. Whereas a single-risk-factor model only requires the estimation of two correlations, using the most complex model from Meng, et al. (2010) requires estimating two vectors of regression factors, as well as three correlation factors.

Each of the parameters listed in Table 3.1 requires a certain kind of dataset to estimate: asset correlation parameters p_i and q_i each require a collection of observations of defaults and LGDs, respectively, from a homogenous portfolio of lenders. From these data, asset correlations can be estimated using, for example, the sample covariance between observations (Düllman, Küll, & Kunisch, 2008).

In order to also estimate the parameter ω (as defined by Witzany (2009)), we would need to observe the common risk factors in both PD and LGD across different time periods, and estimate a correlation between the two. This requires a dataset with many observations over a long time period in order to be accurate. Since we cannot observe the common risk factors at given points in time directly, we need to estimate them from data, which requires many observations in each period. In order to then estimate the correlation from the common risk factor estimates, many of these estimates are needed, which means we will need PD and LGD data over a longer time period.

For models including macroeconomic factors, estimating parameters requires not only data on PD and LGD, but also on the relevant factors which we want to include in our regressions. These factors need to be transformed in such a way that they fit a standard normal distribution. Next to the macroeconomic and firm-specific factors, the models also require default and LGD observations over a longer time period. In the case of Meng, et al. (2010), where we need to estimate an idiosyncratic correlation factor ω, we need to be able to identify individual lenders across the two sets of observations (defaults and LGDs). We need to be able to do this in order to divide the variance in PD and LGD into a common part and an idiosyncratic part. If we only need to estimate a Frailty factor, like in the model by Chava, et al. (2011), this identity field is not necessary, because this model does not identify correlation between PD and LGD on an idiosyncratic level. However, we still need sufficient data on firm-specific factors to make PD and LGD estimations on the account level.

3.2.2. COMPLEXITY AND REALISM

While each complication added to a model increases its data requirement, it can also potentially increase its realism and usefulness. In this section, we discuss the way in which complexities added by each of the four models from Table 3.1 increase their usefulness.

The least complex model is the one described in Frye, et al. (2000). It assumes that the economic factors that affect lenders' PDs are the same ones that affect their LGDs, as the model employs a single common risk factor. This last assumption is problematic, because it causes a perfect correlation between the common risk factors. Given a very large number of loans, this causes the default frequency and the average LGD to be completely dependent as idiosyncratic risks are cancelled out. This is of course not realistic, which is why the model leads to capital levels that are too high.

The model described by Witzany (2009) is an attempt to improve on the complete correlation of the single risk factor model by taking away the assumption that economic factors influencing PD and LGD are exactly the same. Instead, they are assumed to be different, correlated factors. This means that an ODF of quantile x won't be automatically accompanied by the loss rate of the same quantile, which makes the model more realistic. For example, in home mortgages, the LGD is determined by the market value of housing, while PD is more related to consumers' buying power and unemployment rates. Both of these different factors are in turn influenced by the economic cycle, causing them to be correlated.

A next step in complexity is determining the PD and LGD by a vector of macroeconomic and client-specific variables, all of which can then be simulated together. This method makes the model more directly relatable to real scenarios and gives it more legitimacy. Finally, correlating the idiosyncratic risk factors as suggested by Meng, et al. (2010) will reflect the fact that firms with a raised risk of default tend to also have decreased collateral value (Meng, Levy, Kaplin, Wang, & Hu, 2010). In this way, each added complexity in our described models increases the legitimacy and realism of the resulting loss simulations. However, these complexities also each require additional data to make reliable parameter estimates. Without sufficient data, the use of a more complex model does not add value.

3.3. Conclusion

As our goal is to create a model that reflects reality as well as possible, we should choose a modelling method that allows us to do that within our data constraints. The logical solution is to choose the most complex model for which we can reliably estimate parameters using the data available. Since the best dataset we have is an anonymised collection of defaults, we do not have the necessary firm-specific data to estimate a regression model for LGD and PD based on them. We only have a collection of default frequencies and a collection of observed LGDs. The best solution for our situation is therefore to use the two-factor model: it uses a probability distribution driven by Gaussian variables instead of a regression model, so we can estimate parameters without knowing much about individual clients. This will likely require some assumptions to be made about homogeneity of the loan portfolio. Due to the short time period and the global nature of the dataset, we must also have reservations about the reliability of our results and their applicability to real credit portfolios. We describe the definitions and parametrisation of the two-factor model based on the Global Credit Data on large corporate entities in the next section of the report.

4. Estimation methods

In this section of the report, we describe the definitions and parametrisations of our credit loss model based on the two-factor framework described by Witzany (2009). First, we define our model and pay attention to the assumptions we need to make. We also choose a probability distribution to use for the LGD. Then, we test several estimation methods for our model parameters and choose the most efficient one. In Section 5, we apply the chosen estimation method to our data and describe the results.

4.1. Definition and assumptions

As described in Section 2.2.2, the two-factor model for credit loss modelling is defined by the following equations, slightly adjusted from Equations 2.10 and 2.11:

$$A_{i,t} = pX_t + \sqrt{1-p^2}\xi_{i,t} \tag{4.1}$$

$$B_{i,t} = q\left(\omega X_t + \sqrt{1-\omega^2}Y_t\right) + \sqrt{1-q^2}\zeta_{i,t} \tag{4.2}$$

Where ω, p and q are correlation parameters and all X, Y, $\{\xi_i\}$ and $\{\zeta_i\}$ are i.i.d. N(0,1) random variables. During any given period *t*, instances of $A_{i,t}$ and $B_{i,t}$ are generated for each client *i* to determine whether or not the client defaults in that period and, if he defaults, what the bank's loss will be on his loan. For our estimation, we will use periods corresponding to calendar years, because our data on defaults is aggregated into yearly frequencies. We will pay no attention to the specific date in a year on which a client has gone into default and pretend that all default event outcomes in a given year were determined at one moment during that year. The defaults and LGDs are then defined as follows:

$$D_i = \begin{cases} 1, & if\ \ A_i > N^{-1}(1-PD_i) \\ 0, & otherwise \end{cases} \tag{4.3}$$

$$LGD_{i,t} = F_{LGD}^{-1}\left(N(B_{i,t})\right) \tag{4.4}$$

Since our data is anonymised, we can only make estimations under the assumption of a homogenous portfolio of lenders. For this reason, the indices on correlation parameters p_i and q_i have been removed. For our analysis, we will assume that all clients in our portfolio have the same asset correlation factor. We also cannot observe client ratings or other identifying properties in our data, so we have to assume that all clients have the same PD and the same expected LGD while estimating our parameters. This is a strong assumption, since our database contains defaults from many different rating classes and countries. Given more information, we would want to create several pools of similarly rated or located entities we could more legitimately assume to be homogenous. Further on in the report, we describe the way to extend our model to incorporate different asset classes.

By making the assumption of a homogenous portfolio, we can observe the common risk factors of each year by performing transformations on our dataset. Provided that we have enough observations, the conditional mean of all values of $A_{i,t}$ in a given year should be equal to X_t. Similarly, $B_{i,t}$ has an expected value of $q\big(\omega X_t + \sqrt{1-\omega^2}Y_t\big) = qZ_t$. Due to the strong law of large numbers, for large data samples, the observed default frequency in a year (ODF_t) and the portfolio's mean LGD (LGD_t) can be expressed as a function of X_t and Z_t.

$$ODF_t \approx g(X_t) = N\left(\frac{pX_t + N^{-1}(PD)}{\sqrt{1-p^2}}\right) \tag{4.5}$$

$$LGD_t \approx h(Z_t) = \int_{-\infty}^{\infty} F_{LGD}^{-1}\left(N\left(qZ_t + \sqrt{1-q^2}w\right)\right)\varphi(w)dw \tag{4.6}$$

Using these equations, we can find an approximate value of X_t from the ODF, while the value of Z_t can be approximated by performing transformations on the LGD observations and then taking the mean. To perform these transformations, however, we need the values of *p* and *q*, so these will need to be estimated first. Then, we can approximate the time series $\{X_t\}$ and $\{Z_t\}$ and estimate the correlation parameter ω. Using the estimation method of moments, we can use the sample covariance of $\{X_t\}$ and $\{Z_t\}$ as an estimator for ω. This works because the expected value of this covariance is equal to ω^2.

4.2. LGD DISTRIBUTION

The first step in observing the common risk factors is to choose an appropriate probability distribution to approximate that of the LGD. We use this distribution in two ways: firstly, we need the distribution to transform LGD observations into a uniform distribution on the unit interval in order to observe instances of Z_t while estimating the value of ω. Secondly, we use the distribution to generate LGD values when simulating credit losses. Especially in this last case, it is important to have a distribution which can be quickly evaluated, mainly because of the large number of LGDs that need to be generated. This means that we must find an analytical probability distribution to fit to our data.

If we take another look at the histogram in Figure 3.1, we see that our LGD data is mostly restricted to the unit interval, with concentrations of observations on both ends of the interval. These properties make the beta distribution look like a good fit: it is restricted to the (0,1) interval and it can be parametrised to be bimodal with concentrations at 0 and 1, like our LGD data. However, we need to make a few adjustments to our data before it can be properly fitted: since the beta distribution is only supported on the open unit interval, all observations equal or smaller than 0 and all observations equal or greater than 1 need to be changed to values within this interval. Respectively, they should be adjusted to 0+ε and 1-ε, where ε is a small number.

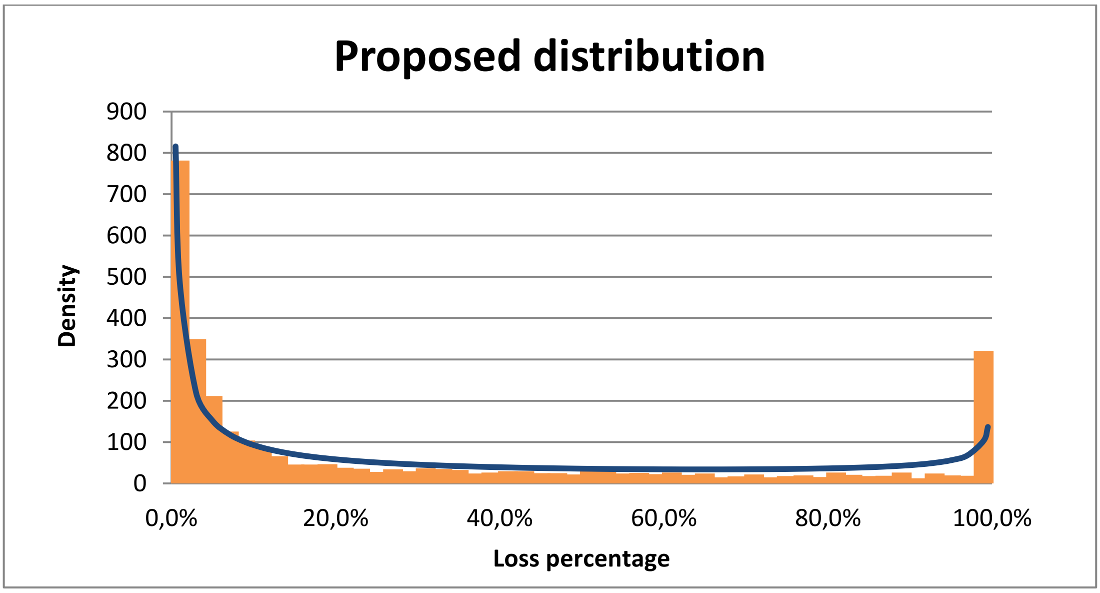


FIGURE 4.1. THE DENSITY FUNCTION OF THE PROPOSED BETA DISTRIBUTION LAID OVER LGD DATA

To estimate the parameters of our beta distribution, we obtain maximum likelihood estimators for the parameters of the beta distribution, *a* and *b*, by using the MATLAB function *fitdist*. As input data, we use a vector of LGD observations from the GCD dataset, adjusted to fit the open unit interval. To find a good value for ε, we perform a grid-search with many different values and choose the one for which the fittest distribution has a variance closest to our sample variance. This ensures that the likelihood of extreme events for our distribution is similar to the actual likelihood. The script used to find ε can be found in Appendix A. The value we find for ε is thirty basis points, or 0.003. The resulting beta parameters are [a, b] = [0.2625, 0.5998], which renders a bimodal distribution that seems to fit the shape of the data, as seen in Figure 4.1.

4.3. Description of estimation methods and comparison

Now that we have determined which probability distribution to use for the LGD, we can estimate the correlation parameters of our model. In this section and the following, we describe the estimation methods we can use for this and we compare their efficiency at generating estimates. We do this by generating several different sets of testing data with known parameters, and performing parameter estimation on them. First, we describe the properties of our datasets. Then, we describe the estimation methods that are tested in this report. Finally, in Section 4.4, we describe the testing experiments and elaborate on their results.

We use several different kinds of testing data: one type ('realistic') is roughly the size of the Global Credit Data we have in terms of number of years and number of defaults, while the other ones are different. We generate a large number of each type and perform estimations on them, in order to test the influence of several data properties on the quality of estimators. Next to the realistic case, we have an ideal case, with a long dataset and lots of loans per year, a small portfolio set with a small number of customers, a mixed portfolio, with different 'buckets' of similar clients, and a risky portfolio with clients that are more likely to default. The properties of the different types of testing data are shown in Table 4.1. The Matlab script used for generating the data can be found in Appendix B.

	"Realistic"	**"Ideal"**	**"Small"**	**"Mixed"**	**"Risky"**
p	0.2	0.2	0.2	0.2	0.2
q	0.2	0.2	0.2	0.2	0.2
ω	0.2	0.2	0.2	0.2	0.2
#years	7	30	30	30	30
#loans	100,000	100,000	10,000	3 × 30,000	100,000
PD	0.8%	0.8%	0.8%	0.5%; 1.0%; 2.0%	4.0%

TABLE 4.1. PROPERTIES OF TESTING DATA

We run each test of estimators with all datasets to see if any dataset properties have effects on the efficiency of any specific estimator. An estimator is tested by generating many different datasets of each type, and estimating parameters from each of them. The estimations from each iteration are saved and put together to create a distribution of parameter estimates for each combination of estimator and dataset type. The mean and variance of these distributions can be used as performance measures for the estimator used, given the type of data. We use parameter values of 0.2, because this is a relatively large value, making it easily detected in sample data, while it still leaves the largest part of the risk drivers to be determined by idiosyncratic variables. This makes our simulations more efficient, while still keeping our testing data somewhat realistic.

There are several methods which are used to estimate correlation parameters from data. Two often-used methods for asset correlation are the method of moments and maximum likelihood (Düllman, Küll, & Kunisch, 2008). In the context of PD-LGD correlation, one other methods was used by researchers: Markov Chain Monte Carlo (MCMC) (Witzany, A Two-Factor Model for PD and LGD Correlation, 2009). This method is an iterative algorithm which can be used to estimate a range of parameters with separate likelihood functions. All three of these methods may be used to estimate parameters in our model. In this section, we describe these estimation methods. In the following section, we compare the quality of their rendered estimates.

4.3.1. Method of moments

The method of moments is an estimation method which involves deriving the expected value of a moment of the random variables. This moment must be dependent on the parameters we want to estimate. After that, we calculate the mean observed value of the derived moment and use it to estimate our parameter. For example, if we want to estimate the μ of a series of normal random variables, we can take the first moment of these variables at E(X) = μ. A method of moments estimator for μ would then be the sample's average value of X.

Likewise, if we want to estimate the variance σ of these variables, we would use the second moment of the series: $E(X^2 - E(X)^2) = \sigma$. Our estimator would then be the average value of $X - \bar{X}$.

For the three parameters we are estimating, we can use several different method of moments estimators. This section is split into three parts, corresponding with the three parameters we need to estimate.

4.3.1.1. p ESTIMATORS

Since we only observe defaults as a binary variable, the moment estimator of *p* has to involve several default indicators. As our moment, we use the product of two default indicators in the same year *t*:

$$D_{i,t} = I(\{pX_t + \sqrt{1-p^2}\xi_{i,t} < N^{-1}(PD)\}) \tag{4.7}$$

$$\begin{aligned} E(D_{i,t}D_{j,t}) &= E\left(E(D_{i,t}D_{j,t}|X_t)\right) \\ &= E\left(P(D_{i,t} = 1 \,\&\, D_{j,t} = 1|X_t)\right) \\ &= N_2(N^{-1}(PD), N^{-1}(PD), p^2) \end{aligned} \tag{4.8}$$

Where N_2 is the bivariate standard normal cumulative distribution. This equality holds because the two default drivers are correlated normal variables with variance one and mean zero. Now that we know the expected value of $D_{i,t}D_{j,t}$, we can find the average value of this variable, over all *t* and *i*, in the following way:

$$\frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t(N_t-1)}\sum_{i}\sum_{j\neq i} D_{i,t}D_{j,t} = \sum_{t=1}^{T}\frac{D_t(D_t-1)}{N_t(N_t-1)} \tag{4.9}$$

Where D_t is the number of defaults in year *t* and N_t is the number of outstanding loans in that year. We find the method of moments estimator for *p* by adjusting the value of *p* in Equation 4.8 such that the expected value of $D_{i,t}D_{j,t}$ given *p* matches the observed average from Equation 4.9. After that, we perform Bessel's correction (Larsen & Marx, 2012) to counteract a bias resulting from Jensen's inequality: especially for small values of T, the estimate of *p* tends to be lower than the actual value. This is because in small samples, the sample mean tends to deviate from the actual expected value to fit the observations, lowering the average distance from the mean. We observe that the estimator becomes less biased if we multiply it by $T/(T-1)$.

4.3.1.2. q ESTIMATORS

Estimating the value of *q* can be done in the same way as *p*, by setting some threshold for the LGD and counting the amount of LGDs that pass this threshold in the same way we count defaults. However, since we have much fewer observations of the LGD than of the ODF, a more efficient method is preferred. We will describe two such methods for estimating *q*, and test both of them at the end of this section. The first method uses the strong law of large numbers to make an estimate based on the variance of B_t, while the second one does not.

We can use our estimated probability distribution to observe approximate values of the LGD risk-driving variables $\{B_{i,t}\}$, which allows us to use a moments estimator, the variance of the common risk-driving variables. We use the strong law of large numbers for this estimate, because we assume that the average value of $B_{i,t}$ in year *t* is approximately equal to its expected value (Equation 4.10). We then observe the sample variance of yearly average risk-driving variables, B_t, and use it to estimate their actual variance:

$$\frac{1}{D_t}\sum_{i} N^{-1}\left(F_{LGD}(LGD_{i,t})\right) \approx B_t \tag{4.10}$$

$$E\left(var(B_t)\right) = E\left((qZ_t)^2 - E\left((qZ_t)\right)^2\right)$$

$$= E(q^2Z_t^2) = q^2E(Z_t^2)$$

$$= q^2 \tag{4.11}$$

$$\hat{q} = \sqrt{var(\{B_t\})} \tag{4.12}$$

Where var({X}) is the unbiased sample variance of the set of variables {X} and, in Equation 4.12, the set {B_t} is to be interpreted as the set of approximations of the actual values of B_t calculated using Equation 4.10. Because this estimator uses the strong law of large numbers, it may not work very well if the data quantity is low. We expect this estimator to need both a large number of years and a large number of clients to work well.

Another method to estimate *q*, without using the strong law of large numbers, is to use the products of two observed loss rates in the same year as a moment. Since LGDs in the same year are correlated with a factor q^2, the expected value of this moment contains a *q* and allows us to estimate the parameter value. Since our chosen LGD distribution is quite complex, however, calculating the expected value is computing power-intensive and involves a double numeric integration:

$$E\left(E\left(LGD_{i,t} * LGD_{j,t} \middle| Z_t\right)\right) =$$

$$= E\left(F_{LGD}^{-1}\left(N\left(qZ_t + \sqrt{1-q^2}\zeta_{i,t}\right)\right) * F_{LGD}^{-1}\left(N\left(qZ_t + \sqrt{1-q^2}\zeta_{j,t}\right)\right)\right)$$

$$= \iint\limits_{-\infty}^{\infty} F_{LGD}^{-1}\left(N(u)\right) * F_{LGD}^{-1}\left(N(v)\right)\varphi_2(u, v, q^2)\, du\, dv \tag{4.13}$$

Where φ_2 is the bivariate normal density. We can then measure the actual average value of $LGD_{i,t}$ * $LGD_{j,t}$ by evaluating the following summation:

$$\frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{D_t(D_t-1)}\sum_{i=1}^{D_t}\sum_{j=1}^{D_t} LGD_{i,t} * LGD_{j,t}\right) \tag{4.14}$$

Of source, since we can only observe loss rates in the event of a default, we can only use the defaulted lenders as observations. Once we have measured the value of the summation, we can adjust the value of *q* in Equation 4.13 until the expected value matches the measured value. The value of *q* for which this is true is our estimate. Since this estimator does not require the strong law of large numbers in order to be derived, we expect that it will work better than the previous *q* estimator when analysing small or short datasets. This conjecture will be tested in Section 4.4.

4.3.1.3. *ω* Estimators

Similarly to the previous section on *q* estimators, in this section we describe two different estimators for *ω*: one that uses the strong law of large numbers and one that does not. The first estimator, which is based on observing common factors and using their sample covariance as an estimate, uses the law. The second one is based on measuring the average value of the product of two observations. It does not require the strong law of large numbers. We compare the two methods in Section 4.4.

Once we have estimates for both *p* and *q*, using the strong law of large numbers, we can use these estimates to observe the values of {Z_t} and {X_t}. The sample covariance between these two series is a method of moments estimator for *ω*. We approximate the values of X_t and Z_t in the following way:

$$\hat{X}_t \approx \frac{N^{-1}(ODF_t)\sqrt{1-\hat{p}^2} - N^{-1}(\,\overline{ODF}\,)}{\hat{p}} \tag{4.15}$$

$$\hat{Z}_t \approx \frac{1}{D_t}\sum_i N^{-1}\left(F_{LGD}\left(LGD_{i,t}\right)\right) \tag{4.16}$$

$$\omega \approx \sqrt{covar\left(\{\hat{X}_t\},\{\hat{Z}_t\}\right)} \tag{4.17}$$

Since this estimator uses the strong law, we expect it to need a relatively large numbers of observations to work properly.

ω Could also be estimated by measuring the average value of the product of one lender's loss rate in year *t* and another lender's default indicator in the same year. The expected value of this product contains both *ω* and *q*, so we need our previously estimated value of *q* for it to work. The expected value of the product is as follows:

$$E\left(E\left(D_{i,t} * D_{j,t} * LGD_{j,t} \middle| Z_t\right)\right) =$$

$$= E\left(D_{i,t} * D_{j,t} * F_{LGD}^{-1}\left(N\left(qZ_t + \sqrt{1-q^2}\zeta_{j,t}\right)\right)\right)$$

$$= \iint_{-\infty}^{\infty} N\left(\frac{N^{-1}(PD_i)+pu}{\sqrt{1-p^2}}\right) N\left(\frac{N^{-1}(PD_i)+pu}{\sqrt{1-p^2}}\right) F_{LGD}^{-1}\left(N(v)\right)\varphi_2(u,v,q\omega)\,du\,dv \tag{4.18}$$

By adjusting the value of *qω* in Equation 4.18 until the value of the integral matches a measured value, we can find an estimated value for *qω*. We can then divide this estimate by our estimate for *q* to find an estimate for *ω*. We find our measurement by taking the following average:

$$\frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{n_t(n_t-1)}\sum_{i=1}^{n_t}\sum_{\substack{j=1\\ j\neq i}}^{n_t} D_{j,t} * D_{i,t} * LGD_{i,t}\right) =$$

$$= \frac{1}{T}\sum_{t=1}^{T}\left(\frac{(D_t-1)}{n_t(n_t-1)}\sum_{i=1}^{n_t} D_{i,t} * LGD_{i,t}\right) \tag{4.19}$$

Since this estimator does not use the law of large numbers, we expect it to work better than the previously described *ω* estimator when using small data samples.

This concludes the section on the method of moments. In Section 4.4, we compare the efficiency of these estimators with others using generated data.

4.3.2. Maximum likelihood & Likelihood functions

A different method for estimating parameters is the maximum likelihood estimator. This estimate is generated by maximizing the likelihood of observing the sample data, over the value of the parameter:

$$\hat{\theta} = \arg\max_{\theta} P(data|\theta) \tag{4.20}$$

If a process is governed by several parameters, this may complicate the process of finding an analytical maximum to the likelihood function. For example, in our process, the LGD is governed by both factors *q* and *ω*, which means likelihood functions will contain both of them. The asset default correlation *p*, however, can be found independently of the other two factors. A maximum likelihood estimate for *p* can be calculated as follows (Düllman, Küll, & Kunisch, 2008):

$$\hat{p} = \sqrt{\frac{\frac{m_2}{T} - \frac{m_1^2}{T^2}}{1 + \frac{m_2}{T} - \frac{m_1^2}{T^2}}};\ m_1 = \sum_{t=1}^{T} N^{-1}(ODF_t);\ m_2 = \sum_{t=1}^{T} \left(N^{-1}(ODF_t)\right)^2 \qquad (4.21)$$

Where T is equal to the number of years for which we have data. Once again, we perform Bessel's correction and multiply our estimate by $T/(T-1)$ to remove the bias. While this means that the estimator is no longer a maximum likelihood estimator, it makes it better for our purposes, as we have a very short dataset. We will still refer to this *p* estimator as the maximum likelihood estimator for the readability of this section. The quality of this estimator will be compared with the quality of the method of moments estimator for *p* in Section 4.4.

It is also possible to create a likelihood function involving all parameters at once and maximise this function. The following likelihood function, based on estimating X_t and Y_t using ODFs and average LGD's, can be derived (Witzany, A Two-Factor Model for PD and LGD Correlation, 2009):

$$\begin{aligned} L(\{ODF_t, LGD_t\}|p, q, \omega) &= \prod_{t=1}^{T} L(ODF_t)\, L(LGD_t|ODF_t) \\ &= \prod_{t=1}^{T} \frac{\varphi(X_t)}{\frac{\partial g}{\partial x}(X_t)} \prod_{t=1}^{T} \frac{\varphi(Y_t)}{\sqrt{1-\omega^2}\frac{\partial h}{\partial x}\left(\omega X_t + \sqrt{1-\omega^2} Y_t\right)} \qquad (4.22) \end{aligned}$$

Where *g(x)* and *h(x)* are the functions described, respectively, in Equations 4.5 and 4.6, and φ is the normal density function. Witzany (2009) obtains this likelihood function by multiplying Equations 4.5 and 4.6 with the normal distribution function, then differentiating and applying the chain rule to obtain the density functions. The likelihoods of the ODF's make use of the strong law of large numbers, since the observed ODF is used to imply X_t. Since all observations of X_t and Y_t are independent, the functions can then be multiplied to get the overall likelihood function.

Witzany's likelihood function can be most easily maximized iteratively, using the following algorithm:

0. Assign a vector of initial values $\{p^0, q^0, w^0\}$. Then, set $j = 1$.
1. Obtain p^j by maximizing likelihood: $p^j = \arg\max_p L(\{ODF_t, LGD_t\}|p, q^{j-1}, w^{j-1})$
2. Obtain q^j by maximizing likelihood: $q^j = \arg\max_q L(\{ODF_t, LGD_t\}|p^j, q, w^{j-1})$
3. Obtain w^j by maximizing likelihood: $w^j = \arg\max_w L(\{ODF_t, LGD_t\}|p^j, q^j, w)$
4. Check the absolute value δ of the difference between $\{p^{j-1}, q^{j-1}, w^{j-1}\}$ and $\{p^j, q^j, w^j\}$
5. If δ < ε, where ε is a very small number, return $\{p^j, q^j, w^j\}$. Otherwise, set $j = j + 1$ and return to step 1.

The function, however, is computationally demanding and potentially unstable, as there may be local maxima on which the function can get stuck. Furthermore, it only renders point estimates, which may not be informative enough for research purposes if there is uncertainty about the quality and size of the dataset. Because of these reasons, Witzany (2009) recommends using a different method of estimation for the model: Markov Chain Monte Carlo.

4.3.3. Markov Chain Monte Carlo

Markov Chain Monte Carlo is a category of Bayesian sampling methods which sample parameter values in an iterative process. This is especially useful when the parameters have complex multivariate densities, because

MCMC methods break these densities down into univariate densities, which are more manageable. The samples drawn by a MCMC process form a Markov Chain, as each sample is generated by adding a random number to the last sample. After a large number of iterations, the samples generated will resemble samples from the multivariate density that is intended.

The Markov Chain Monte Carlo method used by Witzany to estimate parameters is called Gibbs sampling (Geman & Geman, 1984). This method can be used to generate samples of parameters using the data. In a generic case, if we want to estimate a set of parameters {θ_i; i=1,2,…,k}, the procedure of generating samples would be as follows (Witzany, A Two-Factor Model for PD and LGD Correlation, 2009):

0. Assign a vector of initial values $\{\theta_i^0\}$. Then, set j = 1.
1. Sample $\theta_1^j \sim D\left(\theta_1 \middle| \theta_2^{j-1}, \dots, \theta_k^{j-1}\right)$
2. Sample $\theta_2^j \sim D\left(\theta_2 \middle| \theta_1^{j}, \dots, \theta_k^{j-1}\right)$

   …

k. Sample $\theta_k^j \sim D\left(\theta_k \middle| \theta_1^{j}, \dots, \theta_{k-1}^{j}\right)$. Then, set $j = j + 1$ and return to step 1.

The conditional distributions $D(\theta_l|\theta_1, \dots, \theta_{l-1}, \theta_{l+1}, \dots, \theta_k)$ together fully define the multivariate distribution of the set of parameters.

For estimating a set of parameters based on a set of sample data, the conditional probabilities of the parameters can be characterised by the likelihood function of the data conditional on the parameters. This function is divided by a normalizing factor *C* so that the total volume under the likelihood plane is equal to one:

$$D\left(\theta_1 \middle| \theta_2^{j-1}, \dots, \theta_k^{j-1}\right) = \frac{L\left(data \middle| \theta_1, \theta_2^{j-1}, \dots, \theta_k^{j-1}\right)}{C} \tag{4.23}$$

To find *C*, however, we need to integrate the likelihood function over its supported domain. Because our likelihood function cannot be analytically integrated, this makes the Gibbs sampling algorithm incredibly inefficient: for each set of *k* samples, we would need to numerically integrate our likelihood function *k* times. Because the likelihood function itself also contains a numeric integration, we would need to perform a very large number of numeric integrations. To counteract this, we use the Metropolis-Hastings algorithm (Hastings, 1970). This algorithm works as follows: for each sampling step in the procedure, instead of actually sampling from the conditional density D, we sample a value for θ_1^j from some arbitrary density (in Equation 4.22, a normal density was inserted). Then, we choose to accept the new sample with probability R:

$$\theta_1^j = \theta_1^{j-1} + M; M \sim N(0, c) \tag{4.24}$$

$$R = \frac{L\left(data \middle| \theta_1^j, \theta_2^{j-1}, \dots, \theta_k^{j-1}\right)}{L\left(data \middle| \theta_1^{j-1}, \theta_2^{j-1}, \dots, \theta_k^{j-1}\right)} \tag{4.25}$$

If we do not accept the new value of the parameter, the parameter keeps its previous value. In this way, as *j* grows large, new samples of the parameters will converge to the actual conditional density of the parameters given the data. The speed of the convergence depends on the chosen density and the starting values for the parameters. If the starting value is close to the actual expected value of the parameter, convergence is instant. If the starting value is further away from it, the amount of iterations before convergence depends on the variance in the chosen probability density. In Figure 4.2, a convergence process is shown where correlation factor *p* is estimated using generated data. The real value of *p* used for data generation is 0.1.

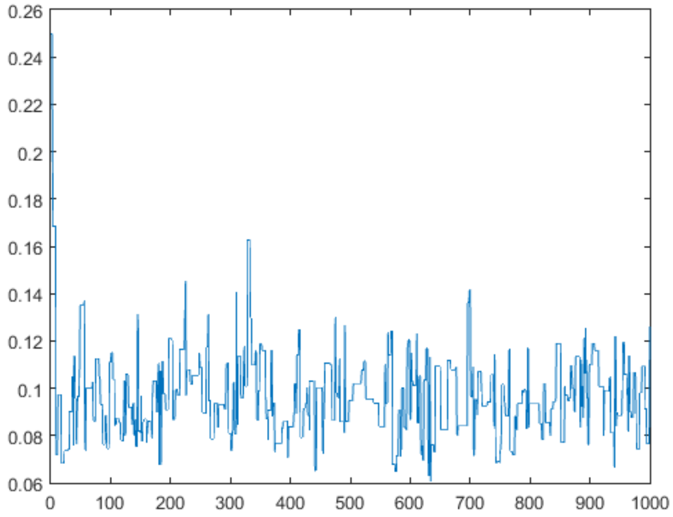


FIGURE 4.2. A SERIES OF P SAMPLES FROM AN MCMC PROCESS. THE SAMPLES CONVERGE AROUND 0.1.

Once a large number of samples has been generated, this set of samples provides a good approximation of the conditional density of the parameter to be estimated. The larger the set of samples generated, the more accurate the approximation will be. In Figure 4.3, we plot a histogram of the sample values shown in Figure 4.2. We can see that a set of 1000 samples provides a relatively good indication of the density's shape and mode, but the histogram is not very smooth yet. Just for identifying the mean, however, this number of samples is sufficient: the mean of the samples is 0.098, which is only a 2% deviation from the actual value.

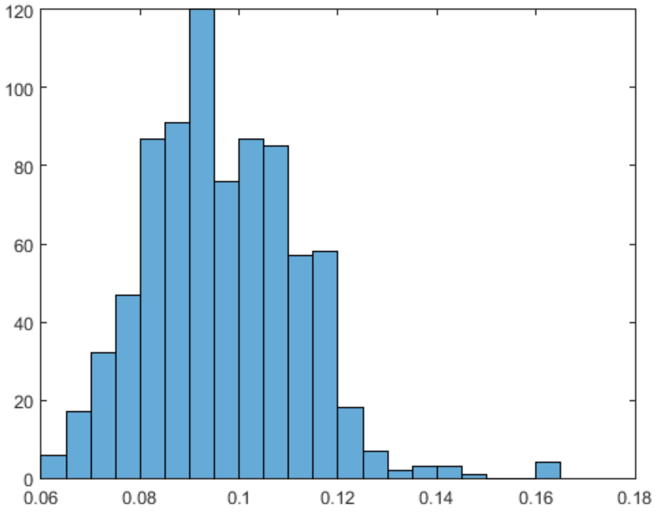


FIGURE 4.3. HISTOGRAM OF SAMPLES SHOWING THE APPROXIMATE CONDITIONAL DENSITY OF P

The accuracy of the MCMC method, like any other method, is dependent on the size of the data sample. A larger dataset will give a more concentrated density and provide a better estimate. We test the quality of MCMC against our moments and maximum likelihood estimators in the next section.

4.4. Comparison of estimation methods

To compare the efficiency of our estimation methods, we define several sets of data and apply the estimation methods on these sets to estimate the parameters that were used. We test the estimators by generating a large number of similarly sized sets of data and estimating parameters from each of these sets. Then, we compile descriptive statistics on estimates from all methods and compare them to each other. The two desirable qualities for estimators are unbiasedness, so a mean estimate equal to the actual value of the parameter, and a low variance of estimates. We use datasets with different properties to test our estimators' performance for different kinds of loan portfolios. These datasets were previously described in Section 4.3, but we show them here again for readability:

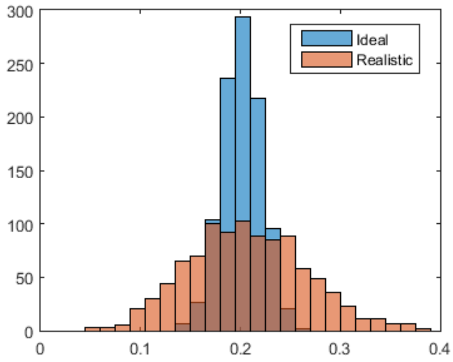
	"Realistic"	**"Ideal"**	**"Small"**	**"Mixed"**	**"Risky"**
p	0.2	0.2	0.2	0.2	0.2
q	0.2	0.2	0.2	0.2	0.2
ω	0.2	0.2	0.2	0.2	0.2
#years	7	30	30	30	30
#loans	100,000	100,000	500	3 × 30,000	100,000
PD	0.8%	0.8%	0.8%	0.5%; 1.0%; 2.0%	4.0%

TABLE 4.1. PROPERTIES OF TESTING DATA (REPEATED FROM SECTION 4.3)

The different sets are intended to test estimators on different kinds of portfolios in order to investigate whether sudden changes to the portfolio affect their accuracy differently.

4.4.1. Estimators for PD Asset correlation

The first test we run is a comparison of estimators for *p*: the moments estimator versus the maximum likelihood estimator. We generate all the datasets described in Section 4.3 one thousand times and perform both the moments estimator and the maximum likelihood estimator for *p* on each dataset. The 'Mixed' dataset is generated in three parts, and estimates are generated for each part separately. The mean of the three estimates for *p* is then used. The means and standard deviations of the sets of estimates are shown in Table 4.2. Histograms of the estimates generated by the two methods for both the realistic and the ideal dataset are shown in Figures 4.4 and 4.5.



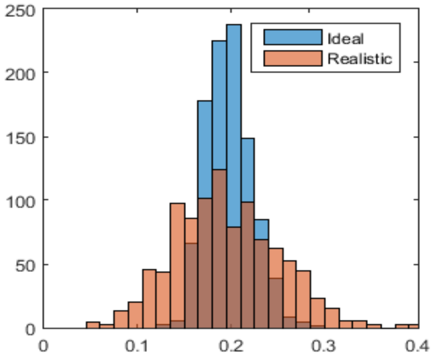


FIGURE 4.4. (LEFT) HISTOGRAM OF MAX. LIKELIHOOD ESTIMATES OF P, FOR BOTH IDEAL AND REALISTIC DATASETS
FIGURE 4.5. (RIGHT) HISTOGRAM OF MOMENTS ESTIMATES OF P, FOR BOTH IDEAL AND REALISTIC DATASETS

Estimation method		**"Realistic"**	**"Ideal"**	**"Small"**	**"Mixed"**	**"Risky"**
Moments	Mean of $\{\hat{p}\}$	0.1949	0.1980	0.1954	0.1991	0.2011
	$\sigma(\{\hat{p}\})$	0.0563	0.0238	0.0433	0.0303	0.0292
Maximum Likelihood	Mean of $\{\hat{p}\}$	0.2071	0.2015	0.2502	0.2040	0.2026
	$\sigma(\{\hat{p}\})$	0.0585	0.0194	0.0342	0.0257	0.0263

TABLE 4.2. DESCRIPTIVE STATISTICS OF P ESTIMATES FROM TESTING DATA

Obviously, the ‘ideal’ dataset of thirty years’ length produces estimates that are more accurate than the ‘realistic’ dataset. Standard deviations of estimates using the ideal sets are twice to three times smaller than those of estimates from realistic datasets. Even with the realistic dataset, however, most *p* estimates are quite close to the actual value of 0.2. Despite correcting for a bias, both estimators still have a slight deviation when using the realistic dataset, but this deviation is a lot smaller than the bias of the uncorrected estimators. Interestingly, while the maximum likelihood estimator seems to be better when estimating parameters based on long datasets, the method of moments estimator performs better for the short datasets. The moment estimates are more concentrated around their mean, which is closer to the actual value. For our data analysis, we will therefore use the method of moments estimator. The scripts and functions used to generate the estimates used for this section can be found in Appendices C and D.

4.4.2. Quality of Moments Estimators

In Section 4.3.1, we described two categories of moments estimators: those that require the law of large numbers to be derived, and those that do not. In this subsection, we compare the two categories by quality to see which one is most suited to our needs. We generate our testing datasets 500 times and perform estimates of *q* and *ω* using both methods. The MATLAB scripts used for this are shown in Appendix J. For the *ω* estimations, we assume that all parameters other than *ω* are known. When applying the estimator to real data, we would estimate these other parameters first and use the estimates, so the *ω* estimates generated here are more accurate than they would be if implemented. The results of the test are shown in Table 4.3.

Estimation method			**“Realistic”**	**“Ideal”**	**“Small”**	**“Mixed”**	**“Risky”**
Using SLLN	q	Mean of $\{\hat{q}\}$	0.2047	0.2019	0.2601	0.1995	0.2003
		$\sigma(\{\hat{q}\})$	0.0536	0.0247	0.0348	0.0448	0.0264
	ω	Mean of $\{\widehat{\omega}\}$	0.1939	0.1880	0.0194	0.1799	0.1800
		$\sigma(\{\widehat{\omega}\})$	0.1514	0.0828	0.0929	0.1046	0.0998
Without SLLN	q	Mean of $\{\hat{q}\}$	0.2330	0.2065	0.6553	0.2051	0.1954
		$\sigma(\{\hat{q}\})$	0.2630	0.1799	0.2996	0.1798	0.1809
	ω	Mean of $\{\widehat{\omega}\}$	0.2373	0.1952	0.1779	0.1946	0.1237
		$\sigma(\{\widehat{\omega}\})$	0.4014	0.3544	0.3475	0.3500	0.2812

TABLE 4.3. DESCRIPTIVE STATISTICS OF Q AND W ESTIMATES FROM TESTING DATA

Clearly, the moments estimators derived without the strong law of large numbers do not live up to our expectations when tested: because these estimators use fewer assumptions, we expected them to be more accurate. However, their variance is much higher than those of the variance-based estimators that use the strong law. The mean of the estimates is accurate in most cases, but nearly all of the individual estimates are inaccurate to the point that they are unusable. Using a ‘small’ dataset does seem to affect the estimators using the strong law more than it does those that don’t use it, but the *q* estimates using the strong law are still more accurate. For *ω* estimates, both categories are very unreliable when using the small dataset.

The reason that the estimators without the SLLN perform so poorly seems to be that the parameters *q* and *ω* have very little impact on the expected value of our measured moments. The impact is so small that the variance of the measurement drives the estimate off-point by tens of percentage points. This fact is illustrated in Figure 4.6, where we see a line indicating the expected value of the measured moment in relation to the value of *q*, pictured on the horizontal axis. The observed moments, scattered horizontally, correspond with wildly differing values of *q*, leading to estimates from 0.01 up to 0.70. The moment as it was conceptualised seems to be too variable in relation to the slope coefficient of its expected value with respect to *q*.

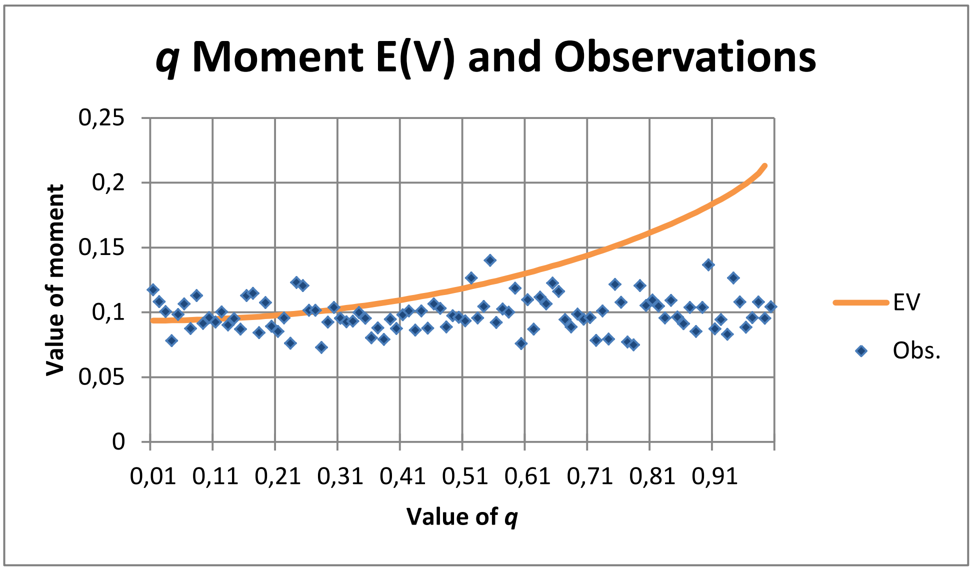


FIGURE 4.6: LINE SHOWING THE EXPECTED VALUE OF A MOMENT IN RELATION TO THE ACTUAL VALUE OF Q, AND A SCATTERING OF OBSERVED VALUES OF THE MOMENT.

In order to improve the moment estimator that does not use the strong law of large numbers, we tried measuring several other moments, including the averages values of $\sqrt{LGD_{i,t}} * \sqrt{LGD_{j,t}}$ and $ln(LGD_{i,t}) * ln(LGD_{j,t})$. These measurements lead to improvements in the accuracy of estimations, because their expected value has a higher slope with respect to q, but none of the operations we tried improved the estimates so much that they became better than our other moments estimator. Because of this, we decided to use the moments estimators derived using the law of large numbers for our comparison of methods.

4.4.3. Moments, Likelihood and MCMC Estimators

The Markov Chain Monte Carlo method (MCMC), as suggested by Witzany, is useful for estimating multiple parameters at once while also getting an impression of their conditional distributions, rather than just a point estimate. However, since our model has sufficiently few parameters to be estimated independently, we can also maximize the likelihood function, or simply use method of moments estimators. We perform a similar simulation experiment to the one conducted in Section 4.4.1 to find out which method provides more accurate estimates. The MATLAB code used for this experiment can be found in Appendices E and F.

The datasets used for this experiment are the same ones as described previously in this section. Once again, we generate each set five hundred times and this time, we estimate all three parameters using the methods as described in Section 4.3. The Markov Chain Monte Carlo algorithm was iterated 500 times for each generated dataset, after which the first 100 samples for each parameter were discarded to allow for time to converge. The mean of the remaining samples was taken as the MCMC estimate for the dataset. The mixed dataset was generated in separate parts, after which estimates were done on each bucket individually. The mean of the three estimates obtained was then used for the table of results. The means and standard deviations of the estimates obtained are shown in Table 4.4.

Estimation method	parameter		"Realistic"	"Ideal"	"Small"	"Mixed"	"Risky"
Moments	p	Mean of $\{\hat{p}\}$	0.1939	0.1993	0.1869	0.1986	0.1994
		σ($\{\hat{p}\}$)	0.0552	0.0242	0.0574	0.0303	0.0293
	q	Mean of $\{\hat{q}\}$	0.2047	0.2019	0.2601	0.1995	0.2003
		σ($\{\hat{q}\}$)	0.0536	0.0247	0.0348	0.0448	0.0264
	ω	Mean of $\{\widehat{\omega}\}$	0.1939	0.1880	0.0194	0.1799	0.1800
		σ($\{\widehat{\omega}\}$)	0.1514	0.0828	0.0929	0.1046	0.0998
Markov Chain Monte Carlo	p	Mean of $\{\hat{p}\}$	0.2087	0.1923	0.0505	0.1809	0.1817
		σ($\{\hat{p}\}$)	0.0503	0.0589	0.3153	0.0823	0.0842
	q	Mean of $\{\hat{q}\}$	0.2343	0.2029	0.1193	0.1978	0.1933
		σ($\{\hat{q}\}$)	0.0617	0.0464	0.3913	0.0813	0.0637
	ω	Mean of $\{\widehat{\omega}\}$	0.1346	0.1743	0.0245	0.1523	0.1661
		σ($\{\widehat{\omega}\}$)	0.2714	0.1333	0.3072	0.1699	0.1740
Maximum Likelihood	p	Mean of $\{\hat{p}\}$	0.1937	0.1937	0.2411	0.1958	0.1937
		σ($\{\hat{p}\}$)	0.0247	0.0247	0.0255	0.0237	0.0247
	q	Mean of $\{\hat{q}\}$	0.1947	0.1947	0.7378	0.2065	0.1947
		σ($\{\hat{q}\}$)	0.0235	0.0235	0.1944	0.0280	0.0235
	ω	Mean of $\{\widehat{\omega}\}$	0.2175	0.2175	0.1194	0.2090	0.2175
		σ($\{\widehat{\omega}\}$)	0.1419	0.1419	0.1130	0.1388	0.1419

TABLE 4.4. DESCRIPTIVE STATISTICS OF PARAMETER ESTIMATES FROM TESTING DATA

The different properties of the various datasets we used seem to affect the estimation methods in different ways. For our considerations, we use the 'Ideal' dataset as a base scenario and investigate the effects of mutations to the properties of the data on the quality of estimates.

We see that given data over a shorter time period, as seen in the 'Realistic' datasets, all the parameter estimates become less accurate and more volatile. Especially estimates of ω, which are only based on as many datapoints as there are years, are affected by this, with standard deviations growing by from 50% (moments estimators) up to 75% (MCMC). Estimates of p and q are less affected, but still become less accurate. The method of moments estimators seem to be least affected by the shortage of years, while MCMC and Maximum Likelihood get larger inaccuracies. The effect of a dataset over a shorter time period is illustrated in Figures 4.7 through 4.9, which show histograms of estimates of ω from generated datasets of both types, gained using the three different estimation methods. The long datasets provide more accurate and less volatile estimates.

Note that there are large peaks at 0 for Moments and Max. Likelihood estimates. This is because these methods, as implemented in this case, do not allow for a negative value of ω, replacing it by 0. The moments estimator for ω is strictly positive because it is the square root of a covariance, and the maximum likelihood estimator is found through a search of the unit interval, making it also strictly positive. Because short datasets are more likely to show a negative sample correlation, replacements by zero are very prevalent in our 'Realistic' dataset tests. As the scope of this report is limited to a positive correlation between PD and LGD, this difference between the methods will not influence the outcome of our research.

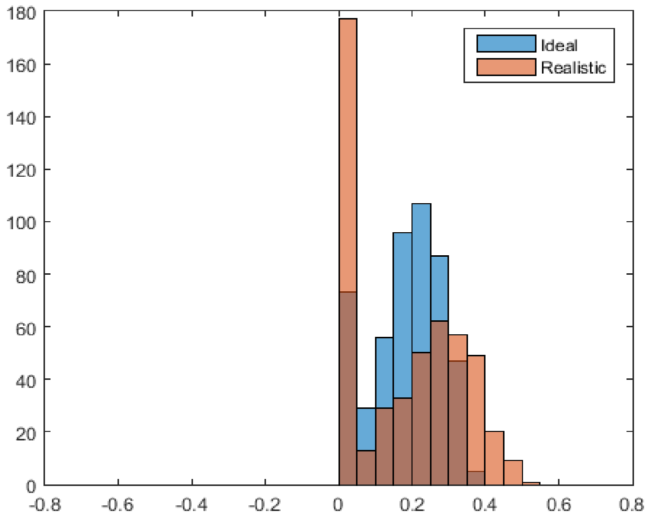


FIGURE 4.7. COMPARATIVE HISTOGRAM OF W ESTIMATES USING A MOMENT ESTIMATOR

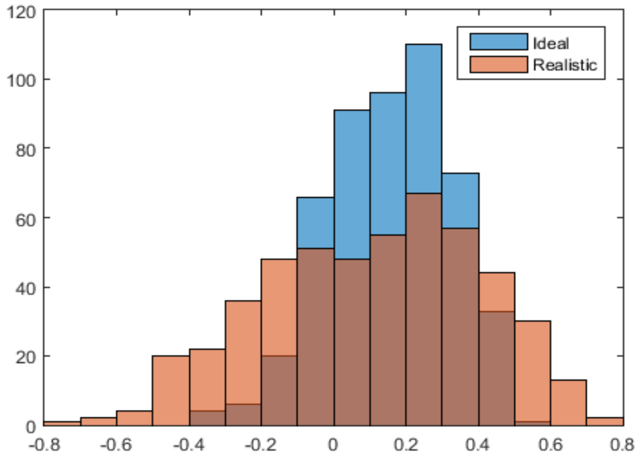


FIGURE 4.8. COMPARATIVE HISTOGRAM OF W ESTIMATES USING MARKOV CHAIN MONTE CARLO

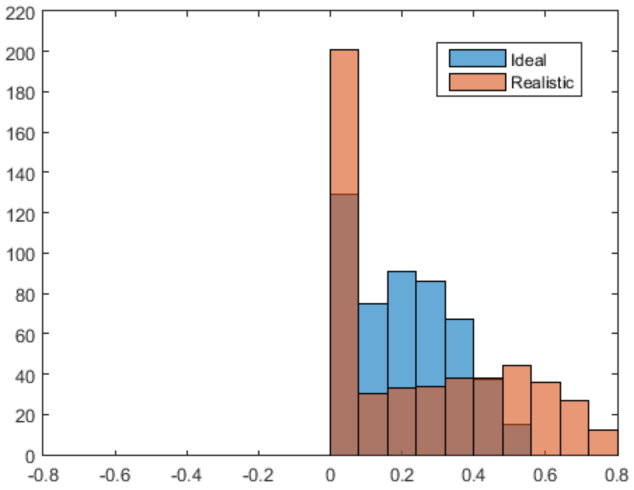


FIGURE 4.9. COMPARATIVE HISTOGRAM OF W ESTIMATES USING MAXIMUM LIKELIHOOD

A lower number of defaults, as seen in the ‘Small’ datasets, doesn’t seem to affect estimates of *p* too much in moments and likelihood estimators. MCMC shows a very small *p*. It appears that a few defaults per year is still enough to make accurate estimates of asset correlation and average PD. This is probably because we only use the ODF each year as observations and not the individual defaults. Estimates of *q* and *w*, however, are more affected, because each individual default counts as an observation and is used as a datapoint in estimates. We see that *q* estimates from all estimation methods are much less accurate, and method of moments estimates also become more volatile. MCMC estimators greatly underestimate *q*, Moments estimators overestimate it and the likelihood function optimisation produces extremely high estimates. The same effects are seen in ω estimates, though less extreme, as especially MCMC wildly underestimates *w*. 500 lenders in a portfolio is apparently not sufficient to make reliable estimates of *q* and *w*. When there are more defaults each year, as in the ‘Risky’ datasets, we see the opposite effects as all estimates become more accurate and Moments estimates also become less volatile.

The division of the dataset into three different ‘buckets’, as seen in the ‘Mixed’ dataset, does not seem to affect the quality of estimates much. Estimates made using this dataset are of roughly the same accuracy and volatility as those made using the ‘Ideal’ datasets. A negative effect on estimate quality may be being compensated by the slightly increased amount of defaults in the dataset, but judging from the lack of change in quality, this negative effect must be quite small.

For our current two-factor model and our dataset from Global Credit Dara, it appears that the method of moments estimators will likely provide more accurate estimates with a lower variance than the methods of Markov Chain Monte Carlo and Maximum Likelihood. For the ‘realistic’ dataset, especially the estimates of ω are a lot less volatile when using the method of moments estimators. Since this parameter is the focus of this research, that is a conclusive advantage. For datasets with fewer years, the p and q estimates of all methods are of comparable quality. While Markov Chain Monte Carlo is a good method to estimate parameters of large and complex models for which moments estimation is not possible, it seems that for simpler models, moments estimators are at least as good. For our own data analysis, we will therefore use the method of moments estimation.

4.5. Conclusion

In this section of our report, we have described a methodology for estimating the simple two-factor model for credit losses based on the kind of data that is available to us – not sorted into buckets and anonymised. When using a simple model with a small set of data, the most efficient estimation method is the one using moments estimators. Given a longer dataset, maximum likelihood becomes a less variable estimator. Bayesian estimation through MCMC becomes interesting once models become too complex for moments or likelihood estimates to be calculated due to interdependencies or complex likelihood functions.

We also investigated how estimators would perform being used on other kinds of datasets. We investigated small amounts of lenders, larger probabilities of defaults, and portfolios made up of several buckets. For the last case, we assumed that for each client, their respective bucket was known to us. Each of these data properties affects estimators in different ways, which are described in section 4.4.2.

In the next section, we use our methodology to perform analysis on our data and try to find a connection between the PD and the LGD.

5. Simulation and results

In this section, we apply the selected methodology described in Section 4 to the Global Credit Data to estimate our model parameters and look for a dependency between PD and LGD. Then, we run a Monte Carlo simulation to approximate a loss distribution for a portfolio of loans to large corporate entities. We also generate a loss distribution without incorporating a dependency between PD and LGD, and compare the results.

5.1. Parameter Estimation

5.1.1. GCD Data Analysis

We use the method of moments estimators described in Section 4.3 to generate estimates for our model parameters. To be able to process the Global Credit Data, we separate our LGD observations into columns, with each column representing a year from our data period. Then, we add zeros to each column to serve as non-default observations. The number of zeros in each column is adjusted such that the frequency of positive column elements matches the ODF from the Global Credit Data. We generate, essentially, a column of observations of $D_{i,t}$ * $LGD_{i,t}$ over one year. This way of representing the data ensures that we can mostly use the same code that we used previously, on generated data, to analyse the Global Credit Data. The only difference is in the storage of the LGD data: because not all years have the same number of loans, we use a cell array to store the columns of observations, rather than a matrix. The MATLAB script used for the data analysis we describe in the following section can be found in Appendix G.

Using the script from Appendix G, we analysed LGD data from GCD over the seven years for which reliable data was available on both ODF and LGD: the years 2007 through 2013. LGD data were available in sufficient numbers from 2006 on, but no ODF was available for the first year. To reiterate the meaning of the parameters: *p* is the parameter for correlation in default events, *q* is the parameter for correlation in LGDs, and *ω* is the parameter for correlation between PD and LGD. Upon performing our analysis, we get the following estimates for *p*, *q* and *ω*:

Parameter	p	q	ω
Estimate based on GCD data	0.14	0.18	0.00*

TABLE 5.1. PARAMETER ESTIMATES USING METHOD OF MOMENTS ESTIMATORS DESCRIBED IN SECTION 4.3.1
(*) – THIS ESTIMATE SET TO 0.00 BECAUSE THE SAMPLE COVARIANCE BETWEEN X AND Z IS NEGATIVE.

Firstly, there appears to be evidence of a correlation between the LGDs of lenders that is similar in size to the correlation of default events. This fact alone has an impact on capital calculations, as it shows us that multiplying an extreme default frequency with the average LGD does not produce a loss estimate for the same extreme case. However, the PD-LGD correlation which we are searching for in this report cannot seem to be found. In fact, we find a slightly negative covariance which causes us to set the *ω* estimate to 0 (because as stated before, our method of moments does not allow for negative parameters). For a confirmation of these results, we also used our other two estimation methods on the GCD Data. These methods also had to be slightly adjusted to use data in cell format, in the same way shown in Appendix G.

Parameter	p	q	ω
Max. Likelihood estimate	0.12	0.20	0.06
MCMC: 5% quantile	0.09	0.16	-0.64
MCMC: mean of samples	0.16	0.26	-0.08
MCMC: 95% quantile	0.25	0.45	0.50

TABLE 5.2. PARAMETER ESTIMATES USING MCMC AND MAX. LIKELIHOOD ESTIMATORS

In Table 5.2, we show the parameter estimates generated by MCMC and Max. Likelihood estimators, as well as a 90% confidence interval for the parameters generated using MCMC. We see that *p* estimates of all three estimation methods are very close, indicating a high level of confidence. *q* estimates are slightly less concentrated, but all three methods are still close, and the confidence interval is fairly tight. For ω, we get a confirmation of our uncertainty, as estimates are all around the zero point and the confidence interval is very wide. The shape of the conditional distribution of ω, formed by samples from the MCMC algorithm, can be seen in Figure 5.1. The distribution appears to be very wide, with a mode slightly left of the zero point, in the negative part of the graph. Clearly, there is not enough evidence to indicate a dependency between PD and LGD in the form we investigated. However, since we saw in Figure 3.2 (repeated in the next section) that LGDs peaked one year before the ODF, in 2008 rather than 2009, we have reason to test if there may be a 'lagged' dependency.

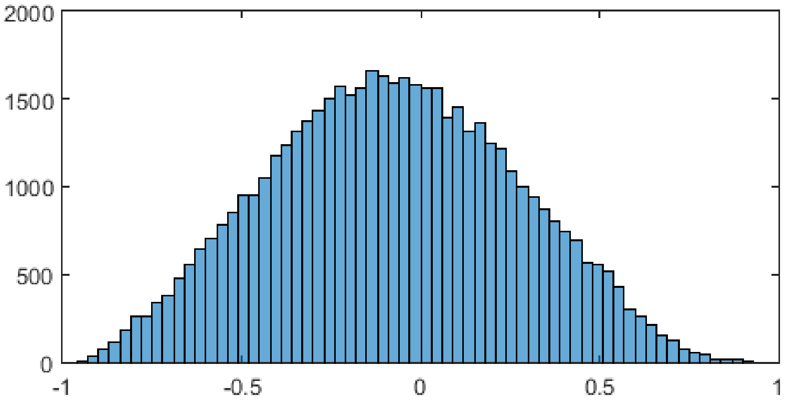


FIGURE 5.1. HISTOGRAM OF W-SAMPLES FROM 50.000 ITERATIONS OF THE MCMC ALGORITHM

5.1.2. Lagging dependency

A 'lagged' dependency is what we call a dependency between the LGD of one year and the PD of the next year. This dependency would indicate that the economic conditions that influence both PD and LGD, influence LGD earlier or more quickly than they do the PD. There is an intuitive explanation for this effect, if it exists: LGD levels are directly influenced by the market prices of assets like real estate, stock holdings and other fixed assets, which govern the amount of money that can be recovered by selling them. In economic downturns, prices for such assets often go down quickly as investments are delayed or stopped and demand drops, raising LGD levels. Meanwhile, firms that cease to be profitable as a result of economic conditions can often subsist on their assets for a while before going into default. Therefore we may expect to see default frequencies rise later than LGD levels. In our limited data sample, we did observe an LGD peak in the year before an ODF peak, which gives us reason to investigate further.

To investigate a possible lagged dependency between PD and LGD, we follow the same process described earlier, this time using the same default frequencies, but with LGDs from 2006 through 2012. ODFs are calculated using the regular 2007-2013 dataset, after which the 2006-2012 LGDs are used for calculations on the variance of the LGD. The process is shown in Appendix H. Using this method, we get the following estimates for the model parameters:

Parameter	**p**	**q**	**ω**
Estimate based on GCD data	0.14	0.15	0.29

TABLE 5.3. PARAMETER ESTIMATES USING METHOD OF MOMENTS ESTIMATORS ON LAGGED DATA

p And q estimates for the lagged dataset are very similar to the previous results, which is to be expected. However, we do see a large difference in ω compared to the previous test: there now appears to be quite a large correlation between the PD and LGD realisations. Once again, we apply the other two estimation methods to confirm that their results are similar.

Parameter	p	q	ω
Max. Likelihood estimate	0.12	0.16	0.51
MCMC: 5% quantile	0.09	0.12	-0.24
MCMC: mean of samples	0.14	0.20	0.34
MCMC: 95% quantile	0.22	0.33	0.75

TABLE 5.4. PARAMETER ESTIMATES USING MCMC AND MAX. LIKELIHOOD ON LAGGED GCD DATA

Once again, p and q results are similar to the earlier results from the normal GCD data, seen in Table 5.2. We are mostly interested in the ω column, which is distinctly different. Both MCMC and Maximum Likelihood estimates are higher than their equivalents in Table 5.2, which confirms the higher ω shown in Table 5.3. The 90% confidence interval for ω obtained by MCMC simulation, is narrower than the one found earlier, and positioned more on the positive side. The shape of the conditional distribution, shown in Figure 5.2, is also different, showing a mode at a higher ω than the expected value. This explains the fact that the Max. Likelihood estimate is much higher than the other two estimates. 85% of the samples obtained is positive, giving us 85% confidence that there is a positive lagged correlation between defaults- and LGD-driving factors. This is not enough to be conventionally significant, but it is an indication that there may be a correlation, and a cause for further research.

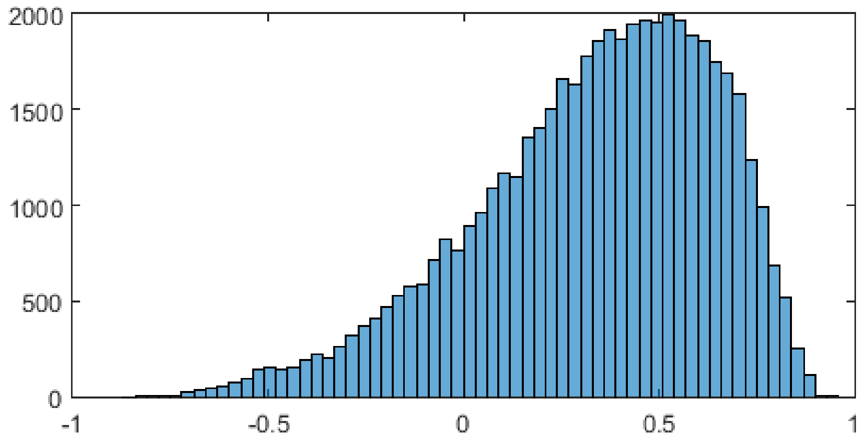


FIGURE 5.2. HISTOGRAM OF W-SAMPLES FROM 50.000 ITERATIONS OF THE MCMC ALGORITHM (LAGGED)

Even though we cannot conclude that the dependency between defaults and LGD exists, our estimation provides enough reason to simulate the effects of such a dependency on capital. In our simulations incorporating the dependency, we will use the method of moments estimates from Table 5.3, since these were deemed the most suitable in Section 4.4, and other results have given us no reason to adjust our method.

5.2. Simulation

In this section, we describe our credit loss simulation process and its results. We do calculations using three different credit models: the regular Vašíček model, currently used for Regulatory Capital, with a fixed LGD, an adjusted model with a stochastic LGD, but without a dependency between defaults and LGD, and finally a two-factor model as described by Witzany (2009), including a dependency between defaults and LGD. In the first subsection, we describe these models and our simulation processes, and in the second subsection we discuss the results of the simulation.

5.2.1. Process

In Section 5.1, we saw that defaults and LGDs of the same year do not appear to be correlated, but that there was a correlation between defaults in one year and the LGDs of the previous year of our data. We adjust our model to reflect this mechanism and investigate the effects on credit losses. According to the capital planning methodology used by Rabobank, market conditions of the previous year should be taken into account when determining clients' probabilities of default (PDs). This is because Rabobank uses point-in-time (PIT) PDs, which are adjusted each time capital calculations are made. Any negative effect of economic conditions on a client's creditworthiness should be picked up by the PD model and taken into account. This makes using a lagged dependency not possible within the current framework, which means that the simulations done in this section are mainly useful as an impact study of the effect on credit losses caused by a PD-LGD dependency. If the dependency is confirmed to exist, this may also have an impact on the way clients' PDs are estimated.

Our simulation study works using the same modelling method described in Section 2.2.1 and repeated in 4.1, with a small difference: we now correlate X_t with Y_{t-1} instead of Y_t. This means we have to slightly adjust our model definition:

$$A_{i,t} = p\left(\omega Y_{t-1} + \sqrt{1-\omega^2}X_t\right) + \sqrt{1-p^2}\xi_{i,t} \tag{5.1}$$

$$B_{i,t} = qY_t + \sqrt{1-q^2}\zeta_{i,t} \tag{5.2}$$

The difference with the previous definition is that Y_t is now determined independently, while X_t is dependent on Y_{t-1}. For our experiments, we adjust the values of *p*, *q* and *ω* between 0 and the values shown in Table 5.3.

We investigate the effect of the correlation on losses by performing a Monte Carlo simulation. Each iteration starts by generating values of X_t and Y_t for two years. While generating these values, we assign a value of 0 to $Y_{0.}$ so that X_1 is simply normally distributed $N\big(0.(1-\omega)\big)$. Then, we generate individual risk-driving factors $A_{i,t}$ and $B_{i,t}$ for the second year and calculate the portfolio loss for that year. This loss is stored in an array, after which the next iteration begins. This way, we generate a large number of realisations of losses to determine the probability distribution of the loss. We perform the Monte Carlo simulation three times with different parameter values. The three experiments' input values are shown in Table 5.5. The code used to perform the experiments is shown in Appendix I.

Experiment	**p**	**q**	**ω**
D-LGD and LGD correlation	0.14	0.15	0.29
LGD correlation	0.14	0.15	0
No extra correlations	0.14	0	0

TABLE 5.5. PARAMETER ESTIMATES USING METHOD OF MOMENTS ESTIMATORS ON LAGGED DATA

Note that due to our assumption of a stationary portfolio and the absence of lender information, our estimates of the parameter values may be slightly higher than usual. This is because our estimators are based on the deviation of yearly average losses and default frequencies from our expectations. Since yearly expectations, which use more of the available information, are generally closer to realised figures, the average deviation from the expectation would be lower if we used them, leading to lower estimated variance. This is why the assumption of a stationary portfolio could lead to a higher estimated correlation parameter. If, in the future, appropriate data becomes available, the parameter estimation should be confirmed using predicted PDs and LGDs.

5.2.2. Results

In this section, we post the results of our Monte Carlo simulation and discuss their implications. For our simulation, we generated each of the three scenarios two million times and saved the aggregated loss percentages on the portfolio. The resulting arrays were sorted in order to generate approximate loss distributions given the sets of parameters. We made histograms of the losses to show the shape of the density functions given the different sets of parameters. These histograms are shown in Figure 5.3.

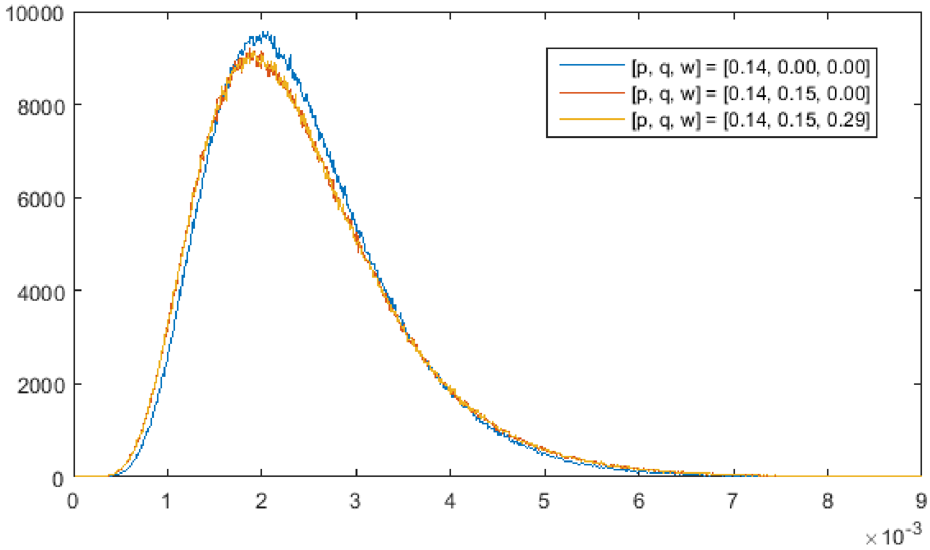


FIGURE 5.3. DENSITIES OF LOSSES GIVEN THREE DIFFERENT SETS OF PARAMETERS

We can see from the shape of the densities in Figure 5.3 that the introduction of a lagged correlation between X and Y does not seem to affect the loss distribution: the yellow and red lines have the same shape. This makes sense, because the factor Y_{t-1} only affects the loss in year *t* through its correlation with X_t. Since it is otherwise not involved in determining the loss, the effect is the same as it would be if X_t were a simple standard normal variable. Realistically, in capital calculations, information from year *t-1* would be used in determining the PD and expected loss, making a lagged correlation spurious. After all, if there were a market downturn in year *t-1*, this would raise PD estimations for year *t*. Clearly, a correlation between PD and LGD is only interesting for capital calculations and modelling if the effect occurs immediately. If macroeconomic factors affect PD and LGD both at different times, there is no shock effect that requires the allocation of extra capital.

The densities' shapes in Figure 5.3 show that while there is no effect from the value of *w*, the introduction of a systemic correlation of LGDs does affect the shape of the loss distribution. A *q* of 0.15 instead of 0.00 makes extreme aggregated losses, both high and low, more likely to occur than usual. This will likely raise the amount of capital required for higher adequacy levels. We compare several extreme right-side quantiles of the three generated distributions in Table 5.6. Note that the expected loss for all three distributions is 0.248%.

Quantile	0.5000	0.9000	0.9900	0.9990	0.9995	0.9999
p, q, w	0.225%	0.381%	0.572%	0.759%	0.815%	0.949%
p, q, 0	0.225%	0.381%	0.573%	0.760%	0.816%	0.955%
p, 0, 0	0.228%	0.372%	0.541%	0.702%	0.747%	0.860%

TABLE 5.6. LOSS PERCENTAGES AT DIFFERENT QUANTILES, GIVEN CERTAIN MODEL PARAMETERS

We see that the median loss is lower when the process governing LGDs has a systematic component. We can also see this in Figure 5.3, where the modes of the distributions with q = 0.15 are situated more to the left. At higher quantiles, the losses on the portfolio modelled with positive q are higher than those modelled with zero q. At the confidence level of 99,99%, used by Rabobank for EC calculations, the EC using q = 0.15 is around 0.71% of the outstanding balance on the portfolio, while EC with q = 0.00 is 0.61%. This means that the systemic nature of LGD's as modelled here raises EC by 15%. At the confidence level required by the regulator, 99.9%, capital levels are raised by 14%, from 0.50% of the outstanding balance to 57%.

5.3. Conclusion

In this section, we performed parameter estimations on our Global Credit Data and used the resulting parameters to do credit loss simulations. We used these simulations to approximate loss distributions for several values of the parameters. We saw that the processes governing defaults and LGD's both had systematic components, which we estimated at p = 0.14 and q = 0.15. Since we had to make the assumption of a stable homogenous portfolio, our estimates for p and q may be overestimated. We found no convincing evidence of a correlation between the two processes in the same year. However, we did find a large lagged correlation between the two processes, which indicates that PD and LGD may be influenced by the same macroeconomic factors, at different speeds.

We performed loss simulations using three models: one with only p positive, one with p and q positive, and one with p and q positive and a lagged correlation between PD and LGD. The last two models returned identical loss distributions. We found that introducing a systematic component into the process governing LGD's leads to an increase in Economic Capital. In the case of the combined GCD portfolio, the size of the effect as estimated would lead to an increase of 15% in EC at 99.99% confidence level.

6. Conclusion

In this final section of the report, we summarise the findings from our research, draw conclusions and make recommendations for further research or actions on part of stakeholders. Initially, as described in Section 1, we set out to answer our main research question:

"How can we create a credit loss model that incorporates a dependency between PD and LGD, which we can use for capital calculations on Rabobank credit portfolios?"

To this end, we performed literature research on the topic to find modelling methods and evidence for the PD-LGD dependency, we looked for suitable datasets, we selected a modelling method and a set of parameter estimators and finally, we created our model and used it to generate a loss distribution on a portfolio of loans. During this process, we discovered several pieces of knowledge that are relevant stakeholders.

Firstly, we found that most loan portfolios at Rabobank do not have the required quality of loss registration to perform the analysis needed for our research. Assessing a dependency between defaults and losses given defaults requires loss registration at the individual lender level, across several years. Definitions of default and loss rate should be consistent over these years to make for a useful dataset.

After acquiring the GCD dataset, we selected a suitable credit loss model: the two-factor model (Witzany, A Two-Factor Model for PD and LGD Correlation, 2009), described in Section 2.2. This model allows for a dependency between defaults and losses at the portfolio level and the usage of any probability distribution for LGD. Given an anonymised dataset like ours, or any dataset which has no detailed information on lenders other than PD and expected LGD, we concluded that a two-factor model is the most efficient choice for generating a loss distribution. The model has three different parameters: one for defaults correlation, one for loss correlation, and one for the correlation between defaults and losses. The parameters are called p, q and ω, respectively. If detailed lender characteristics are available, a more complex model may become useful.

We described several methods that can be used for parameter estimation on the two-factor model and tested them using different generated datasets. We found that moment estimators, which are based on measuring the variance of the data-implied common factors each year, are the most efficient and accurate for most kinds of datasets, including the one we used. In order for estimations of q and ω to be reliable, there should be at least a dozen default observations per year. This is the case because we need to measure the average value of the LGD in order to imply the value of the common factor Y_t for each year. If there are fewer observations, we found that estimations become exceedingly variable and inaccurate.

By estimating parameters using our selected method, we found a value for the correlation between losses, suggesting that losses upon default are correlated at a level similar to that of defaults. This indicates that losses upon default are probably sensitive to macroeconomic variables, which has implications for economic capital. We found that the correlation suggested by our data, if implemented in capital calculations, will lead to a 15% increase in EC at the 99.99% confidence level.

We did not find any evidence for a dependency between defaults and losses in the same year, with the estimated value of the correlation parameter close to zero. We tested for a lagged dependency, with default events following LGD from the previous year, and found a positive correlation, though not at a statistically significant level. This indicates that defaults and losses upon default may be affected by the same macroeconomic variables at different times, but further research is needed to confirm this. In any case, a lagged dependency will not have any impact on EC, as information from previous years should be incorporated in PD estimations.

In conclusion, we described the complete process required to create a credit loss model which incorporates a dependency between defaults and LGDs, and performed data analysis to parametrise our model. Although we ultimately did not find the dependency we were searching for, we did perform useful analyses of various estimators and their respective levels of accuracy, we made inquiries about internal data quality at Rabobank, and we found evidence for a systematic effect in determining LGDs which could affect EC by a significant margin.

6.1. Recommendations

Based on the findings of our research, in this section, we do the following recommendations for further research and future actions:

- Our research should be replicated using a dataset which has not been anonymised, and which has information on PD's and expected loss rates on the individual level. Such data allows us to compare the expected default frequencies and loss rates with the realised values. In this way, the effects of our assumption of the homogeneity of our portfolio can be measured and mitigated.
- While our research suggests that defaults and losses react to macroeconomic variables at different times, this may not apply to different credit markets. Asset classes such as retail credit, SME and home equity loans may show different dependencies between losses and defaults. For a complete credit risk management approach involving a stochastic LGD and a PD-LGD dependency, our research approach should be expanded into these different credit classes.
- In future regulation, a systemic effect in loss determination should be acknowledged and incorporated into credit risk capital requirements. This could be done using the same methods we used in this report, fitting a distribution to LGDs and using simulation, but this requires much more data than banks are currently required to have. In fact, for smaller portfolios, banks are not likely to ever have enough default data to perform the analyses done in this report. An easier solution might be to establish guidelines for using a *downturn LGD* (Basel Committee on Banking Supervision, 2005) on an appropriate level of extremeness. These guidelines should relate to the size of systemic effects and correlations for each asset class.

7. References

Acharya, V. V., Bharath, S. T., & Srinivasan, A. (2007). Does Industry-wide Distress Affect Defaulted Firms? - Evidence from Creditor Recoveries. *Journal of Financial Economics, 85*(3), pp. 787-821.

Altman, E. I. (2006). *Default Recovery Rates and LGD in Credit Risk Modeling and Practice: An Updated Review of the Literature and Empirical Evidence.* New York: New York University, Stern School of Business.

Altman, E. I., Brady, B., Resti, A., & Sironi, A. (2005). The Link between Default and Recovery Rates: Theory, Empirical Evidence, and Implications. *Journal of Business, 78*(6).

Basel Committee on Banking Supervision. (2005). *Guidance on Paragraph 468 of the Framework Document.* Basel: Bank for International Settlements.

Basel Committee on Banking Supervision. (2006). *International Convergence of Capital Measurements and Capital Satandards.* Basel: Bank for International Settlements.

Belyaev, K., Belyaeva, A., Konečný, T., Seidler, J., & Vojtek, M. (2012). *Macroeconomic Factors as Drivers of LGD Prediction: Empirical Evidence from the Czech Republic.* Prague: Czech National Bank Working Paper Series 12.

Calabrese, R. (2014). Downturn Loss Given Default: Mixture distribution estimation. *European Journal of Operational research*, pp. 271-277.

Caselli, S., Gatti, S., & Querci, F. (2008). The Sensitivity of the Loss Given Default Rate to Systematic Risk: New Empirical Evidence on Bank Loans. *Journal of Financial Services Research*, 34, pp. 1-34.

Chava, S., Stefanescu, C., & Turnbull, S. (2011). Modeling the Loss Distribution. *Management Science,* pp. 1267-1287.

Duffie, D., Saita, L., & Wang, K. (2007). Multi-period corporate default prediction with stochastic covariates. *Journal of Financial Economics*, pp. 635-665.

Düllman, K., Küll, J., & Kunisch, M. (2008). Estimating asset correlations from stock prices or default rates - which method is superior? *Deutsche Bundesbank Discussion Paper Series 2: Banking and Financial Studies*.

Eckert, J., Jakob, K., & Fischer, M. (2015). *A Credit Portfolio Framework under Dependent Risk Parameters PD, LGD and EAD.* Erlangen-Nürnberg: Friedrich-Alexander-Universität.

Frye, J. (2000). Depressing Recoveries. *Risk Magazine - 13.11*, pp. 108-111.

Frye, J. (2005). A False Sense of Security. In E. I. Altman, A. Resti, & A. Sironi, *Recovery Rates and Loss Given Default* (p. Ch. 10). London: Risk Books.

Frye, J., Ashley, L., Bliss, R., Cahill, R., Calem, P., Foss, M., . . . Resiak, M. (2000). Collateral damage: A source of systematic credit risk. *Risk Magazine, 13.4,* pp. 91-94.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6), pp. 721-741.

Global Credit Data. (2015). *Home*. Retrieved 10 12, 2015, from Global Credit Data: http://www.globalcreditdata.org/

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1), pp. 97-109.

Keijsers, B., Diris, B., & Kole, E. (2015). *Cyclicality in Losses on Bank Loans.* Rotterdam: Tinbergen Institute.

Larsen, R. J., & Marx, M. L. (2012). *Introduction to Mathematical Statistics and its Applications (5th edition);* pp. 316*.* Boston: Pearson Higher Education, Inc.

Meng, Q., Levy, A., Kaplin, A., Wang, Y., & Hu, Z. (2010). *Implications of PD-LGD Correlation in a Portfolio Setting.* New York: Moody's Analytics.

Pykhtin, M. (2003). Unexpected recovery risk. *Risk magazine, 16*.8, pp. 74-79.

Qi, M., & Zhao, X. (2011). Comparison of modeling methods for Loss Given Default. *Journal of Banking and Finance*, pp.2842-2855

Rösch, D., & Scheule, H. (2009). Credit portfolio loss forecasts for economic downturns. *Financial Markets, Institutions & Instruments, 18*.*1,* pp. 1-26.

Syrkin, M., & Shirazi, A. (2013, March 28). *Modeling Intra and Inter Correlations in Credit Default Losses.* Retrieved September 22, 2015, from Social Science Research Network: http://ssrn.com/abstract=2241137

Vašíček, O. A. (1987). *Probability of Loss on Loan Portfolio.* San Francisco: KMV Corporation.

Wikipedia. (2014, November 14). *Probability density function for the Beta distribution.* Retrieved September 23, 2015, from Wikipedia: https://commons.wikimedia.org/wiki/File:Beta_distribution_pdf.svg

Witzany, J. (2009, September 22). *A Two-Factor Model for PD and LGD Correlation.* Retrieved September 14, 2015, from http://ssrn.com/abstract=1476305

Witzany, J. (2013). *Estimating Default and Recovery Rate Correlations - Working Paper 3.* Prague: Charles University of Economics.

8. Appendices

Appendix A: MATLAB script to find value for Epsilon

This is a MATLAB script that finds a value of epsilon, the variable we use to correct zero and one LGD's to fall within the open unit interval (see Section 4.2) We choose epsilon so that the fitted beta distribution has the same variance as the observed LGD data. It uses a grid-search algorithm to find the correct value.

```
%This script finds a value of epsilon for which the beta distribution
%estimate has the same variance as observations.

%Since our LGD Observations are in percentage points, we need to divide by
%100 first.

load('LGDObs.mat');
LGDObs = LGDObs./100;

%These are the positions of the zeros and ones in the dataset.
Zeros = 1:502;
Ones = 3825:4101;

%We create a grid of epsilons that we can search for the correct value
Epsilons = 0.00001:0.00001:0.01;
A = length(Epsilons);
Variances = zeros(1,A);

%For each possible value of epsilon, we generate a probability distribution
%and evaluate its variance. The variances are stored in a vector.
for i=1:A
    LGDObs(Zeros) = ones(size(Zeros))*Epsilons(i);
    LGDObs(Ones) = ones(size(Ones))-Epsilons(i);

    LGDist = fitdist(LGDObs, 'beta');
    Variances(i) = var(LGDist);
end

%We find the variance closest to the wanted variance, and find the
%corresponding Epsilon. This is our result.
Differences = abs(Variances - 0.1137);
EpsInd = Differences == min(Differences);

Epsilon = Epsilons(EpsInd);
```

APPENDIX B: MATLAB FUNCTION FOR GENERATING TESTING DATA

This function generates a matrix of defaults and a matrix of LGDs according to the parameters specified in the inputs.

```
function [Defaults, LGDs] = GenerateData(Rho_PD, Rho_LGD, Omega, nYears,
nLoans, PD)

%This function generates PD and LGD data according to a Two-Factor model.

%The input variables for this function are:
    %*Rho_PD: the asset correlation for PD. AKA p^2
    %*Rho_LGD: the asset correlation for LGD. AKA q^2
    %*Omega: the correlation between PD and LGD factors. AKA w^2
    %*nYears: the number of years of data required
    %*nLoans: the number of loans at the beginning of each year
    %*PD: the probability that a loan will be defaulted on in a given year

%Each loan is identical to the next in this simulation. The output of the
%function is two nLoansxnYears matrices containing default indicators and
%LGDs.

load('LGDBetaDist.mat'); %Load the beta distr. for LGDs

%Here we generate common risk factors for PD (row 1) and LGD(row 2)
Commons = randn(2,nYears);
Commons(2,:) = sqrt(Omega)*Commons(1,:) + sqrt(1-Omega)*Commons(2,:);

%Now we generate the idiosyncratic risk factors and adjust them using
%common factors and correlations.
PDFactors = randn(nLoans, nYears);
LGDFactors = randn(nLoans, nYears);


%Correlate the risk factors using the common factors
for i = 1:nYears

    PDFactors(:,i) = sqrt(1-Rho_PD)*PDFactors(:,i) +
sqrt(Rho_PD)*Commons(1,i);
    LGDFactors(:,i) = sqrt(1-Rho_LGD)*LGDFactors(:,i) +
sqrt(Rho_LGD)*Commons(2,i);
end

%Create empty matrices for the observed defaults and LGDs
Defaults = [];
LGDs = zeros(nLoans, nYears);

%Now transform the risk factors into default indicators and LGD rates.
Defaults = floor(normcdf(PDFactors)+PD);
A = find(Defaults == 1); %we only generate LGDs for defaulted loans.
LGDs(A) = icdf(LGDist,normcdf(LGDFactors(A)));
```

APPENDIX C: MATLAB SCRIPT FOR GENERATING P ESTIMATES

This script is used to generate estimates of p using two different estimators. It references two functions that actually estimate the parameter value. These functions can be found in Appendix C.

```
%SCRIPT: TestEstimatorsPD.m
%Parameter Estimation using two different estimators. This script generates
%estimates of p (or sqrt(Rho)) using two different estimators. Each
%estimator is calculated N times. The length of the dataset is nYears.

MomentRhos = [];
MLRhos = [];

N=1000;
nYears = 7;

%We use MATLAB's parallel computing loop instead of a normal loop.
parfor a = 1:N
    [Defaults, LGDs] = GenerateData(0.2^2,0.2^2,0.2^2,nYears,100000,0.008);

    %Here we calculate the time series of averages from Defaults and LGDs.
    ODFs = mean(Defaults);
    NumDefs = sum(Defaults);

    PDEst = mean(ODFs); %Estimate a global (TTC) PD

    %We estimate Rho1 (p^2) using two different functions.

    MomentRhos(a) = MomentRho(ODFs,PDEst,Defaults);
    MLRhos(a) = MLRho(ODFs,PDEst);

end

%Now, we convert the Rho values into values of p:
MLRhos = sqrt(MLRhos);
MomentRhos = sqrt(MomentRhos);

%We save our data.
save('Rho1Test', 'MomentRhos', 'MLRhos');

%Now, we create a histogram of both sets of estimates
hold on
h1 = histogram(MomentRhos);
h2 = histogram(MLRhos);
h1.BinWidth = h2.BinWidth;
h1.DisplayName = 'Moments';
h2.DisplayName = 'MaxLikelihood';
hold off
```

APPENDIX D: TWO MATLAB FUNCTIONS TO ESTIMATE P

This appendix contains two MATLAB functions: one that creates moments estimates of p^2 and one that creates maximum likelihood estimates. First, the method of moments function:

```
function RhoEst = MomentRho(ODFs,PDEst,Defaults)

%This function estimates the asset correlation (Rho or p^2) based on
%default frequencies per year. It uses a method of moments estimator which
%is derived in our report.

%This function now supports different amounts of loans each year. All
%spaces in Defaults with value -1 will be ignored.

DiDj = [];
S = size(ODFs);

for t = 1:S(2)
    D = Defaults(:,t);
    NN = size(D(D~=-1));
    N = NN(1);
    X = ODFs(t)*N; %N is the number of loans each year. Adjust accordingly!
    DiDj(t) = X*(X-1)/(N*(N-1));
end

M2 = mean(DiDj);
P1 = norminv(PDEst);

%We find the correct value of Rho through a grid search: we create a vector
%of Rho values for corresponding bivariate distributions.
RhoVector = [0:0.001:1];

%Here, we generate the covariance matrices for our RhoVector.
sigmas = ones(2,2,1001);
for i = 1:1000
    sigmas(:,:,i) = [1 RhoVector(i); RhoVector(i) 1];
end

%We calculate the expected value of our moments estimator for each of the
%Rho values in our grid.
mu = [0 0];
DistValues = [];
for i = 1:1000
    sigma = sigmas(:,:,i);
    DistValues(i) = mvncdf([P1 P1], mu, sigma);
end

%We find the value of Rho for which the expected value of our estimator is
%closest to the actual value.
Differences = abs(DistValues - M2);
RhoInd = Differences == min(Differences);

%Finally, we choose the corresponding Rho and apply a Bessel correction.
RhoEst = RhoVector(RhoInd)*((S(2))/(S(2)-1))^2;
```

Now, the maximum likelihood estimator:

```
function RhoEst = MLRho(ODFs,PDEst)

%This function estimates the asset correlation based on default rates. It
%is based in the methodology described in Dullmann (2008), p.11
%This estimate is a Maximum Likelihood estimator.

T = size(ODFs);
T = T(2);

m_1 = sum(norminv(ODFs));
m_2 = sum(norminv(ODFs).^2);

%Note that we apply a Bessel correction at the end.
RhoEst = (((m_2/T)-((m_1)^2)/(T^2))/(1 + (m_2/T) - ((m_1)^2)/(T^2)))*(T/(T-
1))^2;
```

APPENDIX E: MATLAB SCRIPT TO COMPARE MCMC AND MOMENTS ESTIMATORS

This script generates a set of testing data and estimates its parameters using both method of moments estimators and MCMC estimators. It references functions described in previous appendices, as well as WitzanyLikelihood, a likelihood function, and OptWitzLikelihood, a function that optimises the likelihood function. These functions are shown in Appendix F. As shown, the script is configured to generate 'realistic' datasets, with a length of 7 years.

```
%Parameter Estimation.

%This script is based on the following model (and will generally use the
%same conventions):

%Default driver A(i) = sqrt(Rho_PD)*X + sqrt(1-Rho_PD)*Xi(i)
%LGD Driver B(i) = sqrt(Rho_LGD)*(sqrt(Omega)*X +
%sqrt(1-Omega)*Y)+sqrt(1-Rho_LGD)*Zeta(i)

N = 1000; %Number of estimates to be made;
MCMC_N = 500; %Number of MCMC iterations to be done (>>100);
nYears = 7; %Number of years in each dataset.
nLoans = 30000; %Number of loans each year.
PD0 = 0.008; %PD each year.
P = ProgressBar(N); %Start a Progress Bar

%We define the likelihood function to be the one described by Witzany
L = @WitzanyLikelihood;
load('LGDBetaDist.mat'); %Load the beta distr. for LGDs

%We create empty arrays to hold the estimates.
p_EstMom = [];
q_EstMom = [];
w_EstMom = [];

p_EstMCMC = [];
q_EstMCMC = [];
w_EstMCMC = [];

p_EstML = [];
q_EstML = [];
w_EstML = [];

%We set default starting points for the MCMC process.
r1_0 = 0.5;
r2_0 = 0.5;
w_0 = 0.5;

parfor a = 1:N
    [Defaults, LGDs] = GenerateData(0.2^2,0.2^2,0.2^2,nYears,100000,0.008);
    %This script generates two matrices: Defaults and LGDs.

    %First, we perform the moments estimating process.
    %Here we calculate the time series of averages from Defaults and LGDs.
    ODFs = mean(Defaults);
    NumDefs = sum(Defaults);
    LGDRs = sum(LGDs)./sum(LGDs~=0);
    LossRates = mean(LGDs);

    PDEst = mean(ODFs); %Estimate a global TTC PD
    DistEst = fitdist(nonzeros(LGDs), 'beta'); %Estimate a TTC LGD Dist.

    %First, we estimate Rho1. We use a moment estimator.

    Rho1Est = MomentRho(ODFs,PDEst,Defaults);

    %Now, we estimate the Rho2. For this, we try to observe the LGD risk
    %factors by performing a transformation.

    SL = size(LGDs);
    LGDFactorVars = [];
    LGDCommonFactors = [];
    for i = 1:SL(2)
        LGDFactorVars(i) = var(norminv(cdf(DistEst,nonzeros(LGDs(:,i)))));
        LGDCommonFactors(i) = 
mean(norminv(cdf(DistEst,nonzeros(LGDs(:,i)))));
    end

    %Now, for Omega, we will try to observe the common factors for PD and 
LGD
    %using our previously estimated PD and LGD parameters.

    %We take squareroots of the Rho values to get p and q values.
    p_EstMom(a) = sqrt(Rho1Est);
    q_EstMom(a) = sqrt(var(LGDCommonFactors));

    PDCommonFactors = Basel(Rho1Est,ODFs,PDEst,1);
    CovarMatrix = cov(PDCommonFactors, LGDCommonFactors);
    Omega2Est = max(0,CovarMatrix(1,2));

    w_EstMom(a) = sqrt(Omega2Est);

    %Now, we perform the MCMC estimation process.
    %We create empty arrays.
    p = zeros(MCMC_N,1);
    q = zeros(MCMC_N,1);
    w = zeros(MCMC_N,1);

    %We set their first values to the null values set earlier.
    p(1) = r1_0;
    q(1) = r2_0;
    w(1) = w_0;

    for i = 2:MCMC_N
        p(i) = p(i-1) + norminv(rand(),0,0.1);
        if (p(i))^2 < 1
            R = min(L(ODFs, LGDs, p(i), q(i-1), w(i-1), DistEst, PDEst) / 
L(ODFs, LGDs, p(i-1), q(i-1), w(i-1), DistEst, PDEst),1);
            u = rand();
            if u > R
                p(i) = p(i-1);
            end
        else
            p(i) = p(i-1);
        end

        q(i) = q(i-1) + norminv(rand(),0,0.1);
        if (q(i))^2 < 1
```

```
            R = min(L(ODFs, LGDs, p(i), q(i), w(i-1), DistEst, PDEst) / 
L(ODFs, LGDs, p(i), q(i-1), w(i-1), DistEst, PDEst),1);
            u = rand();
            if u > R
                q(i) = q(i-1);
            end
        else
            q(i) = q(i-1);
        end
        
        w(i) = w(i-1) + norminv(rand(),0,0.1);
        if (w(i))^2<1
            R = min(L(ODFs, LGDs, p(i), q(i), w(i), DistEst, PDEst) / 
L(ODFs, LGDs, p(i), q(i), w(i-1), DistEst, PDEst),1);
            u = rand();
            if u > R
                w(i) = w(i-1);
            end
        else
            w(i) = w(i-1);
        end
        
    end
    
    %The mean of the samplings once converged will be the estimate.
    p_EstMCMC(a) = mean(p(100:MCMC_N));
    q_EstMCMC(a) = mean(q(100:MCMC_N));
    w_EstMCMC(a) = mean(w(100:MCMC_N));
    
    %Optimize the likelihood function to find the ML parameters;
    [p_EstML(a), q_EstML(a), w_EstML(a)] = OptWitzLikelihood(ODFs, LGDs, 
DistEst, PDEst);
    
    %Update the Progress Bar
    P.progress;
    
end

P.stop; %Stop the Progress Bar

%Save the parameter estimates.
save('MomvsMCMCRealistic.mat','p_EstMCMC', 'q_EstMCMC', 'w_EstMCMC', 
'p_EstMom', 'q_EstMom', 'w_EstMom');
```

APPENDIX F: WITZANY'S LIKELIHOOD FUNCTION

```
function L = WitzanyLikelihood(ODFs, LGDs, p, q, w, Dist, PD0)

%This function calculates the likelihood function for a set of observed PDs
%and LGDs, given estimates for parameters and the function h(x) =
%E(G(Z)|x). G is the LGD distribution. The process is described in
%Witzany(2009).

%This function refers to g_dif and h_dif. These are the derivatives of g
%and h as mentioned in Witzany(2009).

A = size(ODFs);
B = size(LGDs);
T = A(2);

if T ~= B(2)
    error('Number of ODFs and LGDRs not equal!')
end

%PDLs will be  an array of likelihoods for the observed PDs, given
estimates
%for Rho1 and PD0. Since we are assuming an ARMA(0,0) process, u1(t) is
%equal to x(t). First, we must evaluate x(t) and u(t) for all t.

X = (norminv(ODFs)*sqrt(1-p^2)-norminv(mean(ODFs)))/p;
U1 = X;  %Must be adjusted if we assume ARMA(p,q) with p,q~=0
Beta_0 = 1; %This must also be adjusted in the aforementioned case.

PDLs = normpdf(U1) ./ g_dif(X,p,PD0)*Beta_0;

%LGDLs will be an array of likelihoods for the observed LGDs, given
%estimates for Rho2 and the probability distribution Dist. Once again, u(t)
%will be equal for now. First, we observe the common factors and store them
%in array Y.

SL = size(LGDs);
Y = [];

parfor i = 1:SL(2)
    Y(i) = mean(norminv(cdf(Dist,nonzeros(LGDs(:,i)))));
end

%now we normalize Y so that the (expected) variance becomes 1.
Y = Y*(1/q);

%The calculated values are not the values of the driver: we need to
% subtract and divide to correct for x and u.

Y_noX = (Y - w*X) / sqrt(1-w^2);
U2 = Y_noX; %To be adjusted in case p,q~=0

LGDLs = normpdf(U2) ./ (sqrt(1-w^2)*Beta_0*h_dif(Y,q,Dist));
Ls = [LGDLs PDLs];
L = prod(Ls);
```

G_DIF AND H_DIF

This function calls on two other functions: h_dif and g_dif. These are the derivatives of functions g and h described in Equations 4.5 and 4.6. These functions are pasted here.

```
function G = g_dif(x,r1,PD0)

G = (normcdf((r1*(x+0.001) + norminv(PD0))/sqrt(1-r1^2))-normcdf((r1*x +
norminv(PD0))/sqrt(1-r1^2)))/0.001;


function H = h_dif(y,r2,Dist)

%This function approximates the derivative of h(x) as described in Witzany
%(2009).

S = size(y);
H = [];

parfor i = 1:S(2)

    Ws = norminv(0.001:0.001:0.999);
    Ds = normpdf(Ws);
    Ds = Ds / sum(Ds);

    A1 = Ws*sqrt(1-r2^2) + r2*y(i);
    A2 = Ws*sqrt(1-r2^2) + r2*(y(i)+0.001);

    A1 = normcdf(A1);
    A2 = normcdf(A2);

    A1 = icdf(Dist,A1);
    A2 = icdf(Dist,A2);

    A1 = A1 .* Ds;
    A2 = A2 .* Ds;

    H(i) = (sum(A2) - sum(A1))/0.001;
end
```

Optimisation

The Witzany likelihood function is optimized using another function, OptWitzLikelihood. This function optimizes the likelihood function iteratively, looking for a maximum from the starting values. A central assumption here is that our search spaces (denoted by m) are sufficiently small to make function values at the edges of an interval close to the maximum value within that interval. For m=0.1, this seems to be the case.

```
function [p, q, w] = OptWitzLikelihood(ODFs, LGDs, Dist, PD0)

%This function optimises Witzany's likelihood function for a given set of
%data, a probability distribution for the LGD and an estimated probability
%of default. It optimizes the function using an iterative algrithm. We
%optimize for each parameter separately, in turn through grid searches.
L = @WitzanyLikelihood; %We set L to be Witzany's Likelihood function.

%We create empty arrays to store the consecutive estimates for each
%parameter.
p = [];
q = [];
w = [];

%We set starting values for the parameters.
p(1) = 0.5;
q(1) = 0.5;
w(1) = 0.5;

%We create parameters for a while-loop. k is the difference between
%parameters on step i.
k = 1;
i = 2;

%We iterate until k is equal to zero.
while k>0.01
    %First, we do a search to find the most likely value of p.
    m = 0.1;
    p(i) = 0.5;
    while m>=0.001
        P = p(i)-4*m:m:p(i)+4*m;
        PLs = zeros(size(P));
        parfor j=1:length(PLs)
            PLs(j) = L(ODFs, LGDs, P(j), q(i-1), w(i-1), Dist, PD0);
        end
        PInd = PLs == max(PLs);
        p(i) = mean(P(PInd));
        m = m/10;
    end

    %Then, we find q.
    m = 0.1;
    q(i) = 0.5;
    while m>=0.001
        Q = q(i)-4*m:m:q(i)+4*m;
        QLs = zeros(size(Q));
        parfor j=1:length(QLs)
            QLs(j) = L(ODFs, LGDs, p(i), Q(j), w(i-1), Dist, PD0);
        end
        QInd = QLs == max(QLs);
        q(i) = mean(Q(QInd));
        m = m/10;
    end
    %Finally, we find w.
    m = 0.1;
    w(i) = 0.5;
    while m>=0.001
        W = w(i)-4*m:m:w(i)+4*m;
        WLs = zeros(size(W));
        parfor j=1:length(WLs)
            WLs(j) = L(ODFs, LGDs, p(i), q(i), W(j), Dist, PD0);
        end
        WInd = WLs == max(WLs);
        w(i) = mean(W(WInd));
        m = m/10;
    end

    %We calculate the differences between steps. If there is a difference,
    %we start over.
    k = abs(p(i)-p(i-1))+abs(q(i)-q(i-1))+abs(w(i)-w(i-1));
    i = i +1;
end

%We assign the final estimates as the return values.
p = p(i-1);
q = q(i-1);
w = w(i-1);
```

Appendix G: MATLAB script to analyse Global Credit Data

This is the script we use to generate parameter estimates from Global Credit Data. It loads a cell array of observations and then performs analysis year by year. MomentRhoData is a function called by this script that is mostly identical to the MomentRho script described in Appendix D. It is adjusted to read cells rather than matrices. Note that the only difference between the versions described here and the originals, is the way data is loaded In the loops that create the ODFs and LGDCommonFactors and LGDFactorVars arrays. These loops now call cells instead of parts of matrices.

```
%Parameter Estimation.

%This script is based on the following model (and will generally use the
%same conventions):

%Default driver A(i) = sqrt(Rho_PD)*X + sqrt(1-Rho_PD)*Xi(i)
%LGD Driver B(i) = sqrt(Rho_LGD)*(sqrt(Omega)*X +
%sqrt(1-Omega)*Y)+sqrt(1-Rho_LGD)*Zeta(i)

%First, we load our data (a cell of LGD arrays. A 0 means no default).
load('LGDDefaultsCell.mat');
AllLGDs =
[LGDCell{1,1};LGDCell{1,2};LGDCell{1,3};LGDCell{1,4};LGDCell{1,5};LGDCell{1
,6};LGDCell{1,7}];
AllLGDs = nonzeros(AllLGDs);

%Here we calculate the time series of averages from Defaults and LGDs. We
%also record the lengths of the vectors to create a weighted average PD
%later.

nYears = 7; %7 is the number of years in our dataset.

ODFs = zeros(1,nYears);
Weights = zeros(1,nYears);
Total = 0;

%We calculate the yearly ODFs and record the amount of observations each
%year. This has to be done separately for each year, because the data is
%stored in cells rather than in a matrix.

for i = 1:nYears
    D = LGDCell{1,i};
    ODFs(i) = length(nonzeros(D))/length(D);
    Weights(i) = length(D);
    Total = Total + length(D);
end

PDEst = sum(ODFs .* Weights)/Total; %Enter the weighted average PD.
DistEst = makedist('beta',0.2124,0.4367); %Enter our chosen distribution.

%First, we estimate Rho1. We know that the number of defaults in a
%year should be distributed according to the Basel formula. We use a
%maximum likelihood estimator.

Rho1Est = MomentRhoData(ODFs,PDEst,LGDCell);

%Now, we estimate the Rho2. For this, we try to observe the LGD risk
%factors by performing a transformation.

LGDFactorVars = zeros(1,7);
LGDCommonFactors = [];
```

```
for i = 1:7
    D = LGDCell{1,i};
    LGDFactorVars(i) = var(norminv(cdf(DistEst,nonzeros(D))));
    LGDCommonFactors(i) = mean(norminv(cdf(DistEst,nonzeros(D))));
end

%Now, for Omega, we will try to observe the common factors for PD and LGD
%using our previously estimated PD and LGD parameters.

p_Est = sqrt(Rho1Est);
q_Est = sqrt(var(LGDCommonFactors));

%Initially, we estimate Omega^2. We take the square root to get our
%estimate for omega.

PDCommonFactors = Basel(Rho1Est,ODFs,PDEst,1);
CovarMatrix = cov(PDCommonFactors, LGDCommonFactors);
Omega2Est = CovarMatrix(1,2);

w_Est = sqrt(Omega2Est);
```

Moment Rho Estimator

This function is MomentRho from Appendix D, adjusted to deal with cell arrays of observations.

```
function RhoEst = MomentRhoData(ODFs,PDEst,LGDCell)

%This function estimates the asset correlation (Rho or p^2) based on
%default frequencies per year. It uses a method of moments estimator which
%is derived in our report.

%This function now supports different amounts of loans each year. All
%spaces in Defaults with value -1 will be ignored.

DiDj = [];
S = size(ODFs);

for t = 1:S(2)
    D = LGDCell{1,t};
    N = length(D);
    X = ODFs(t)*N; %N is the number of loans each year. Adjust accordingly!
    DiDj(t) = X*(X-1)/(N*(N-1));
end

M2 = mean(DiDj);
P1 = norminv(PDEst);

%We find the correct value of Rho through a grid search: we create a vector
%of Rho values for corresponding bivariate distributions.
RhoVector = [0:0.001:1];

%Here, we generate the covariance matrices for our RhoVector.
sigmas = ones(2,2,1001);
for i = 1:1000
    sigmas(:,:,i) = [1 RhoVector(i); RhoVector(i) 1];
end

%We calculate the expected value of our moments estimator for each of the
%Rho values in our grid.
mu = [0 0];
DistValues = [];
for i = 1:1000
    sigma = sigmas(:,:,i);
    DistValues(i) = mvncdf([P1 P1], mu, sigma);
end

%We find the value of Rho for which the expected value of our estimator is
%closest to the actual value.
Differences = abs(DistValues - M2);
RhoInd = Differences == min(Differences);

%Finally, we choose the corresponding Rho and apply a Bessel correction.
RhoEst = RhoVector(RhoInd)*(S(2)/(S(2)-1))^2;
```

Appendix H: Lagged data analysis

To perform data analysis on lagged data, we need to adjust the way the data is read. For the moments estimators, this bit was coded using a separate cell for the LGDs from 2006 called 'LGD2006' and saved in the file 'LGD2006 Cell.mat'. In this appendix, we show both the script used for moments estimations and the piece of script used to adjust the data for entry into MCMC and Max. Likelihood functions. First, the script for moments.

```
%Parameter Estimation on lagged data.

%This script is based on the following model (and will generally use the
%same conventions):

%Default driver A(i) = sqrt(Rho_PD)*X + sqrt(1-Rho_PD)*Xi(i)
%LGD Driver B(i) = sqrt(Rho_LGD)*(sqrt(Omega)*X +
%sqrt(1-Omega)*Y)+sqrt(1-Rho_LGD)*Zeta(i)

%First, we load our data (a cell of LGD arrays. A 0 means no default).
load('LGDDefaultsCell_e0.003.mat');
load('LGD2006 Cell.mat');
AllLGDs =
[LGDCell{1,1};LGDCell{1,2};LGDCell{1,3};LGDCell{1,4};LGDCell{1,5};LGDCell{1
,6};LGDCell{1,7}];
AllLGDs = nonzeros(AllLGDs);

%Here we calculate the time series of averages from Defaults and LGDs. We
%also record the lengths of the vectors to create a weighted average PD
%later.

nYears = 7; %7 is the number of years in our dataset.

ODFs = zeros(1,7);
Weights = zeros(1,7);
Total = 0;

for i = 1:7
    D = LGDCell{1,i};
    ODFs(i) = length(nonzeros(D))/length(D);
    Weights(i) = length(D);
    Total = Total + length(D);
end

PDEst = sum(ODFs .* Weights)/Total; %Enter the weighted average PD.
DistEst = makedist('beta',0.2124,0.4367); %Enter our chosen distribution.

%First, we estimate Rho1. We know that the number of defaults in a
%year should be distributed according to the Basel formula. We use a
%maximum likelihood estimator.

Rho1Est = MomentRhoData(ODFs,PDEst,LGDCell);

%Now, we estimate the Rho2. For this, we try to observe the LGD risk
%factors by performing a transformation.

LGDFactorVars = zeros(1,7);
LGDCommonFactors = [];

%We first take the 2007 LGDs for the first variable, then we use a loop
%with adjusted indices.
D = LGD2006{1,1};
```

```
LGDFactorVars(1) = var(norminv(cdf(DistEst,nonzeros(D))));
LGDCommonFactors(1) = mean(norminv(cdf(DistEst,nonzeros(D))));

for i = 1:6
    D = LGDCell{1,i};
    LGDFactorVars(i+1) = var(norminv(cdf(DistEst,nonzeros(D))));
    LGDCommonFactors(i+1) = mean(norminv(cdf(DistEst,nonzeros(D))));
end

%Now, for Omega, we will try to observe the common factors for PD and LGD
%using our previously estimated PD and LGD parameters.

p_Est = sqrt(Rho1Est);
q_Est = sqrt(var(LGDCommonFactors));

PDCommonFactors = Basel(Rho1Est,ODFs,PDEst,1);
CovarMatrix = cov(PDCommonFactors, LGDCommonFactors);
Omega2Est = CovarMatrix(1,2);

w_Est = sqrt(Omega2Est);
```

Lagging data

Now, the piece of script that adjusts data to lagged data before the application of estimation methods:

```
%First, we load our data (a cell of LGD arrays. A 0 means no default).
load('LGDDefaultsCell_e0.003.mat');
load('LGD2006 Cell.mat');
LagCell = {
LGD2006{1,1},LGDCell{1,1},LGDCell{1,2},LGDCell{1,3},LGDCell{1,4},LGDCell{1,
5},LGDCell{1,6} };

%Here we calculate the time series of averages from Defaults and LGDs. We
%also record the lengths of the vectors to create a weighted average PD
%later.

MCMC_N = 1000; %Number of MCMC iterations
nYears = 7; %7 is the number of years in our dataset.
L = @WitzanyLikelihood_GCDData;

ODFs = zeros(1,nYears);
Weights = zeros(1,nYears);
Total = 0;

for i = 1:nYears
    D = LGDCell{1,i};
    ODFs(i) = length(nonzeros(D))/length(D);
    Weights(i) = length(D);
    Total = Total + length(D);
end

PDEst = sum(ODFs .* Weights)/Total; %Calculate the weighted average PD.
DistEst = makedist('beta',0.2124,0.4367); %Enter our chosen distribution.
```

Appendix I: Monte Carlo credit loss simulation

Here, we describe the MATLAB script we used for simulating credit losses on a homogenous portfolio with a lagged correlation between PD and LGD. As it turns out, this correlation does not have much impact on the loss distribution. We generate common factors for two consecutive years and simulate credit losses only for the second year.

```
%This is a script for performing Monte Carlo simulation of credit losses.
%It uses a two-factor model described by Witzany with correlation
%parameters p, q and w. It assumes a homogenous portfolio of lenders, all
%with the same PD and expected LGD.

%This simulation has a delayed correlation between LGD: PD follows LGD.

%First, define the inputs.
p = 0.14;
q = 0.15;
w = 0.29;
PD = 0.008;
nLoans = 100000;
load('LGDBetaDist.mat'); %Load a distribution for LGDs called LGDist.
N = 2000000; %Number of simulations.

%Now, we start the simulation of credit loss. We do this in a parfor loop.

P = ProgressBar(N);

%Create an array to hold the losses.
Losses_qw = zeros(1,N);

parfor i = 1:N

    %Generate random common factors for PD (1,:) and LGD (2,:) over t =
    %1:2. Then, correlate PD(t=2) with LGD(t=1).
    Commons = randn(2,2);
    Commons(1,2) = sqrt(1-w^2)*Commons(1,2) + w*Commons(2,1);

    %Now, generate idiosyncratic variables.
    Idios = randn(1,nLoans);
    Idios = sqrt(1-p^2)*Idios + p*Commons(1,2);

    %Now we select the default cases and generate LGDs for them.
    D = Idios >= norminv(1-PD);
    Defaults = find(D==1);
    IdioLGD = randn(1,length(Defaults));
    IdioLGD = sqrt(1-q^2)*IdioLGD + q*Commons(2,2);
    LGDs = icdf(LGDist, normcdf(IdioLGD));

    %Now, we replace the default indicators by the LGDs and calculate the
    %credit loss on the portfolio.

    D2 = D*1;
    D2(Defaults) = LGDs;
    Losses_qw(i) = mean(D2);

    P.progress;
end

P.stop;
save('Losses_qw_2M.mat', 'Losses_qw');
```

Appendix J: Estimations without the Strong Law

When making estimates without the strong law of large numbers, as described in Section 4.3.1, we do not want to evaluate the double integral every time we generate a new dataset. To circumvent this, we generate an array of expected values of moments for various values of *q* and *ω*. When we calculate a new moment, we look up the closest value to the moment in our array and take the corresponding *q* or *ω* value as our estimate. In this appendix, we post both the code used to generate the expected values and the code used to generate data and estimates.

Code used for generating expected values

This section contains two MATLAB functions that generate expected values of moments described in Section 4.3.1.2 and 4.3.1.3. Both functions use a double loop to evaluate an integral. First, the one used to estimate *q*:

```
function ExpProd = ExpLGDProd(q, LGDist, nPoints)

%This function calculates the expected value of the product of two LGDs,
%given a correlation coefficient of q squared and an LGD Dist..

sigmas = [1 q^2;q^2 1];

Delta = 12/(nPoints-1);
Us = -6:Delta:6;
NUs = normcdf(Us);
FNUs = icdf(LGDist,normcdf(Us));

Products = zeros(nPoints,nPoints);

%Since this function is symmetrical, we only need to generate the upper
%diagonal of the matrix.
for i = 1:nPoints
    for j = i:nPoints

        Products(i,j) = FNUs(i)*FNUs(j)*mvnpdf([Us(i); Us(j)],[0;
0],sigmas)*Delta^2;
    end
end


ExpProd = sum(sum(Products))*2 - mean(mean(eye(nPoints).*Products));
```

Now, the script used for ω:

```
function ExpProd = ExpWProd(p, q, w, PD, LGDist, nPoints)

%This function calculates the expected value of Di * DjLGDj, given a
%value of w, p and q, using numeric integration.

sigmas = [1 (q*w);(q*w) 1];

Delta = 12/(nPoints-1);
Us = -6:Delta:6;
BaselUs = normcdf((norminv(PD)+p*Us)/sqrt(1-p^2));
FNUs = icdf(LGDist,normcdf(Us));

Products = zeros(nPoints,nPoints);

for i = 1:nPoints
    for j = 1:nPoints

        Products(i,j) = BaselUs(i)*BaselUs(j)*FNUs(j)*mvnpdf([Us(i);
Us(j)],[0; 0],sigmas)*Delta^2;
    end
end

ExpProd = sum(sum(Products));
```

CODE USED FOR GENERATING DATASETS AND ESTIMATES

The following code was used to generate datasets and then make estimates of q and ω. The arrays that are loaded at the start are sorted expected values of moments. Their position in the array is one hundredth of the corresponding q or ω value. The wProd array was generated using the same input variables used in this script.

```
nYears = 30;
nLoans = 500;
PD = 0.008;
load('Prod_1000.mat');
load('wProd_0.008_500.mat');


qMoms = [];
q_Ests = [];
wMoms = [];
w_Est = [];

%We run a large number of simulations.
for t = 1:500

    %First, we generate data.
    [Defaults, LGDs] = GenerateData(0.2^2,0.2^2,0.2^2,nYears,nLoans,PD);

    Rho1Est = MomentRho(ODFs,PDEst,Defaults);
    p_Est = sqrt(Rho1Est);

    %To estimate q, we calculate the average value of the product of two
    %LGDs in the same year. This will be our moment.

    T = length(ODFs);
    qMomT = zeros(1,T);

    for i = 1:T
```

```
        A = nonzeros(LGDs(:,i));
        AMat = A * A';
        qMomT(i) = mean(nonzeros(triu(AMat)));
    end

    qMoms(t) = mean(qMomT, 'omitnan');
    ProdDif = abs(LogProd - qMoms(t));
    q_Ests(t) = find(ProdDif==min(ProdDif));

    %Next, we estimate w. We calculate the value of our moment.

    wMomT = zeros(1,T);

    for i = 1:T
        A = LGDs(:,i);
        wMomT(i) = mean(A)*(length(nonzeros(A))-1)/(length(A)-1);
    end

    wMoms(t) = mean(wMomT);
    ProdDifw = abs(wProd - wMoms(t))/100;
    w_Ests(t) = find(ProdDifw==min(ProdDifw))/100;
end
```