VILNIUS UNIVERSITY

FACULTY OF MATHEMATICS AND INFORMATICS
MASTER'S STUDY PROGRAMME
DATA SCIENCE

Posterior-Based Estimation and Benchmark-Based Validation of Probability of Default in Low Default Portfolios

Master's Thesis

Author: Enrika Komarovaitė

Supervisor: Doc., Dr. Andrius Grigutis

Vilnius
2026

Abstract

Under the internal ratings-based approach introduced in Basel II, banks rely on internal probability of default (PD) estimates as key inputs to regulatory capital requirements to cover unexpected credit losses. In low-default portfolios, default events are rare and may even be zero over the years, complicating the estimation and validation of banks' internal PD estimates. In this thesis, PD estimators designed explicitly for low default portfolios are reviewed and used as benchmarks to evaluate the conservatism of internal bank PD estimates. The PD estimation methods reviewed in this thesis are the upper confidence bound approach (Pluto & Tasche, 2005), considered at different confidence levels, and the Bayesian approach (Tasche, 2013), considered under conservative and uniform priors. To assess the conservatism of banks' internal PD estimates, two measures are applied: a log-ratio of the upper confidence bound to the bank's internal PD estimate and the posterior tail probability that the true PD exceeds the bank's internal PD estimate. For better interpretability and comparability across portfolios and years, both raw measures are transformed to a standard $[0, 3]$ score scale and mapped to traffic-light colors. The results show that estimates produced by estimators specifically designed for low default portfolios remain well-defined in cases of rare default events, including when no defaults are observed. The proposed validation framework offers a transparent, adjustable, and interpretable way to flag potential underestimation of the bank's internal PD estimate.

Keywords: Credit Risk, Probability of Default, Low Default Portfolio, Bayesian Estimation, Probability of Default Validation.

Santrauka

Pagal Bazelio II (Basel II) pristatytą vidaus reitingais grįstą (IRB) metodą, bankai naudoja savo vidinius įsipareigojimų nevykdymo tikimybės (PD) įverčius reguliacinio kapitalo, kurio paskirtis yra padengti netikėtus nuostolius, skaičiavimui. Praktikoje egzistuoja tokie portfeliai, kuriuose nemokumo įvykiai yra itin reti, jų gali netgi nepasitaikyti, o tai sukelia iššūkių bankams PD vertinimo ir validavimo procese. Šiame darbe analizuojami žemo nemokumo portfeliams sukurti PD įvertinimo metodai. Taip pat jie pritaikomi bankų vidinių PD įverčių konservatyvumui įvertinti. Darbe taikomi PD įvertinimo metodai: viršutinės pasikliautinės ribos (upper confidence bound) metodas (Pluto ir Tasche, 2005) skirtingiems pasikliovimo lygmenims bei Bajeso metodas (Tasche, 2013), naudojant konservatyvųjį ir tolygųjį apriorinius skirstinius. Bankų vidinių PD įverčių konservatyvumui įvertinti taikomi du matai: viršutinės pasikliautinės ribos įverčio ir banko vidinio PD įverčio logaritminis santykis bei aposteriorinės uodegos tikimybė. Siekiant geresnio rezultatų interpretavimo ir palyginamumo tarp portfelių bei metų, pritaikytų matų rezultatai transformuojami į standartinę $[0, 3]$ balų skalę. Rezultatai rodo, kad įverčiai, gauti taikant PD įvertinimo metodus, specialiai sukurtus žemo nemokumo portfeliams, išlieka apibrėžti net ir esant itin retiems nemokumo įvykiams, įskaitant atvejus, kai nemokumo įvykių visiškai nėra. Be to, pasiūlyta validavimo sistema leidžia lengvai identifikuoti potencialų banko vidinio PD įverčio nepakankamą įvertinimą.

Raktiniai žodžiai: Kredito rizika, Įsipareigojimų nevykdymo tikimybė, Žemo nemokumo portfelis, Bajeso vertinimas, Įsipareigojimų nevykdymo tikimybės validavimas.

List of Figures

List of Tables

Contents

1 Introduction

The Basel regulatory framework establishes international standards to strengthen the resilience and stability of the global banking system. Under the internal ratings-based (IRB) approach introduced in Basel II [3], banks are allowed to use internal rating systems to assign obligors to risk classes and to derive internal estimates of key credit risk parameters, namely the probability of default (PD), loss given default (LGD), and exposure at default (EAD). These three parameters serve as the basis for calculating regulatory capital for unexpected credit losses, and for quantifying and monitoring credit risk. Banks that adopt the IRB approach are required to validate their rating systems and estimates to demonstrate that they meaningfully differentiate credit risk and that estimates of risk parameters do not underestimate the undertaken risk [3, 7].

The minimum requirements of the regulatory framework specify that, for each risk class, the one-year PD used for regulatory capital calculation must be estimated as a conservative view of the long-run average of observed default rates, so that estimates remain appropriate under both calm and adverse economic conditions. In addition, a margin of conservatism has to be added to account for data limitations and estimation uncertainty. Similarly, LGD and EAD estimates must reflect downturn conditions and incorporate margins of conservatism [2, 3].

Estimation of risk parameters is particularly challenging for portfolios that have historically experienced few or no defaults. Such portfolios are defined as Low Default Portfolios (LDPs), and challenges extend to the validation process as well. The Basel framework does not specify a numerical threshold for identifying such portfolios, but describes them as portfolios with very few or no observed defaults, for which estimation and backtesting of risk parameters based on internal default data are complicated and unreliable due to the scarcity of default data. Such portfolios do exist in reality, and typical examples are sovereigns, banks, insurance companies, and highly rated corporates. Nevertheless, under the Basel regulatory framework, banks are still required to estimate and validate estimates of risk parameters for such portfolios, even though scarce default data pose challenges. For estimating risk parameters in cases of scarce default data, regulators suggest using alternative data sources and conservative estimation methods, while suggesting greater use of benchmarking in the validation process [6]. This thesis focuses entirely on the PD parameter.

Despite regulatory emphasis on benchmarking and alternative validation approaches for LDPs, academic research investigating how to validate PD estimates in LDPs remains limited. The main goal of this thesis is to analyse and present selected PD estimators specifically designed for LDPs presented by Pluto & Tasche [14], and by Tasche [18], and to use them as benchmarks to assess the conservatism of bank internal PD estimates.

The thesis is structured as follows. First, the literature review presents PD estimation and validation methods in LDPs; then, the methodology and validation framework are thoroughly explained. Later, the data used in the empirical analysis are presented, and the results are reported. Finally, findings are discussed and conclusions drawn.

2 Literature review

2.1 PD estimation perspectives

In practice, PDs are estimated from two perspectives: Point-in-Time (PIT) and Through-the-Cycle (TTC), depending on their intended purpose. PIT PDs reflect the probability of default over a future horizon, which is typically one year, given current economic and borrower conditions, as a result, they vary with the credit cycle. In contrast, TTC PDs represent long-run, cycle-neutral one-year estimates and are primarily used for regulatory capital calculations to cover unexpected losses. In addition, some banks use hybrid PD measures that lie between PIT and TTC [1, 3]. Therefore, the accuracy of PIT PD estimates is essential for effective credit risk management and setting provisions, while appropriately conservative TTC PD estimates are essential to ensure that banks allocate regulatory capital efficiently and remain resilient under adverse economic conditions. This thesis focuses on TTC PD estimates.

2.2 Estimation of PD in LDP

One of the most popular and widely cited PD estimators specifically designed for LDPs is the upper confidence bound (UCB) approach proposed by Pluto & Tasche [14]. They show how to derive UCB estimates of PD for a chosen confidence level under both independent and correlated default assumptions, and, in addition, for one-period and multi-period settings. The proposed estimator avoids zero estimates when no defaults are observed, which is desirable in LDP settings. The level of conservatism is controlled by the choice of the confidence level γ; however, high confidence levels yield extremely conservative estimates. Also, the subjective choice of the confidence level introduces unnecessary uncertainty about which level is appropriate.

To address criticism regarding the extreme conservatism of estimates and the unwanted subjective choice of confidence level γ, Tasche [18] proposed a Bayesian approach to PD estimation. In this approach, conservatism is controlled by a prior assumption rather than the confidence level γ, as in the UCB estimator. In particular, Tasche considers a neutral prior, which assumes that all PDs are equally likely, and a conservative prior, which assumes that larger PDs are more likely. The proposed Bayesian approach avoids selecting a confidence level and produces estimates, which are summarised as posterior means, with moderate conservatism. Moreover, expert-based priors can incorporate portfolio-specific knowledge, thereby improving the accuracy of the resulting PD estimates. Kiefer [11] also adds that PD estimation in LDPs is limited by data scarcity and that no amount of data processing can overcome the lack of information. He also states that incorporation of expert knowledge through Bayesian priors, which will be updated with default counts, will yield proper posterior distributions even in cases when zero defaults are observed. As a practical estimate, Kiefer suggests the posterior mean and highlights that prior assumptions can significantly influence the resulting PD estimates.

2.3 Validation of PD in LDP

Pluto & Tasche [15] themselves emphasize that their presented methodology regarding the UCB estimate of PD is frequently used for PD validation rather than PD estimation. They emphasize that their UCB estimates are constructed with built-in conservatism; therefore, the fact that PD under validation, i.e., PD obtained from an alternative calibration methodology, does not exceed the UCB estimate does not imply underestimation of the PD by the alternative methodology. Also, they say that UCB estimates can serve as benchmarks to provide insights into the degree of conservatism of PDs obtained from alternative calibration methodologies

> "So which benefits can be derived from validation via benchmarking against low default estimates based on upper confidence bounds? As the low default methodology delivers conservative PD estimates, it can offer some insight into the degree of conservatism for PDs calibrated by another method."

[15, p. 96]. Although they state that their presented methodology is often used for PD validation, no references to existing work were provided. So, it seems that this practice is common but not explored in the academic literature.

Dwyer [8] proposes a Bayesian approach to validate internal bank PD estimates in LDPs. He assumes that defaults are driven by a common systematic factor, just like Pluto & Tasche [14], and Tasche [18] in their works. The proposed validation logic is to assume PD under validation and *asset correlation* value as given and then to derive the posterior distribution of the systematic shock from the observed default count. If posterior distribution concentrates on very extreme systematic shock values, it indicates that PD under validation might need recalibration. Kruger [12] presents a practical work on validation of PD estimates in LDPs by following the validation logic proposed by Dwyer [8]. In addition, he shows how to implement this approach practically in statistical software *SAS*.

3 Methodology

In methodology section PD estimators under the assumption of correlated defaults are presented for both a single period and multiple periods. The methodology for the PD upper confidence bound estimator is based on Pluto and Tasche [14], and the Bayesian approach for PD estimation follows Tasche [18].

3.1 Definitions

Here various probability concepts are defined that will be used throughout the thesis (see, e.g., Ross [17], Grimmett and Stirzaker [10]).

Definition 3.1.1 (Binomial random variable) *Let $n \in \mathbb{N}$ and $p \in (0,1)$. If discrete random variable X is said to follow a binomial distribution with parameters (n,p), denoted $X \sim \mathrm{Bin}(n,p)$, its probability mass function is*

$$\mathbb{P}(X = k) = \binom{n}{k} p^k (1-p)^{n-k}, \qquad k = 0, 1, \ldots, n,$$

where $\binom{n}{k} = \dfrac{n!}{k!(n-k)!}$.

The cumulative distribution function is

$$\mathbb{P}(X \leq k) = \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i}, \qquad k = 0, 1, \ldots, n.$$

Definition 3.1.2 (Normal random variable) *Let $\mu \in \mathbb{R}$ and $\sigma > 0$. A continuous random variable X follows a normal distribution with mean μ and variance σ^2, denoted $X \sim \mathcal{N}(\mu, \sigma^2)$, if its probability density function is*

$$\varphi_{\mu,\sigma^2}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \qquad x \in \mathbb{R}.$$

The cumulative distribution function is

$$\Phi_{\mu,\sigma^2}(x) = \mathbb{P}(X \leq x) = \int_{-\infty}^{x} \varphi_{\mu,\sigma^2}(y)\, dy, \qquad x \in \mathbb{R}.$$

Definition 3.1.3 (Law of total probability) *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and let $A \in \mathcal{F}$ be an event. If Y is a continuous random variable with density f_Y, the law of total probability can be written in the form*

$$\mathbb{P}(A) \ = \ \int_{-\infty}^{\infty} \mathbb{P}(A \mid Y = y)\, f_Y(y)\, dy.$$

Definition 3.1.4 (Conditional density, mixed case) *Let (X, Y) be a pair of random variables, where X is a continuous random variable and Y is a discrete random variable. Suppose that (X, Y) has joint probability density $f_{X,Y}(x, y)$. Then the conditional probability density of X given $Y = y$ is defined by*

$$f_{X|Y}(x \mid y) := \frac{f_{X,Y}(x, y)}{p_Y(y)},$$

where $p_Y(y)$ denotes the marginal probability mass function of Y, see [13, Section 2.3.1].

Definition 3.1.5 (Monte Carlo approximation) *Let X be a random variable taking values in $\mathbb{R}$ with probability density function f on $\mathbb{R}$. Let $h : \mathbb{R} \to \mathbb{R}$ be a measurable function and suppose we are interested in the expectation*

$$\mu \; = \; \mathbb{E}_f[h(X)] \; = \; \int_{\mathbb{R}} h(x)\, f(x)\, dx.$$

A Monte Carlo approximation of μ is obtained by simulating i.i.d. random variables $X^{(1)}, \ldots, X^{(M)} \sim f$ and using the sample average

$$\hat{\mu}_M := \frac{1}{M} \sum_{m=1}^{M} h\Big(x^{(m)}\Big),$$

as an estimator of μ, where $x^{(m)}$ is the realised value of the random variable $X^{(m)}$ in the simulation. By the law of large numbers, $\hat{\mu}_M \to \mu$ almost surely as $M \to \infty$. See [16, Section 3.2]

3.2 Definition of default event under correlated defaults

Here the definition of default event under assumption of correlated defaults is presented, and single-obligor probabilty of default is derived. This subsection largely follows the overview of Grigutis in [9].

We begin with the standard continuous compounding framework, let r be the continuously compounded annual rate of return of an asset. Then if compounding occurs $m \in \mathbb{N}$ times per the year, a unit investment grows by $(1 + r/m)^m$. Since

$$\lim_{m \to \infty} \left(1 + \frac{r}{m}\right)^m = e^r,$$

continuous compounding yields

$$V_F \; = \; V_I\, e^r,$$

where $V_I > 0$ and $V_F > 0$ are the initial and final values over the year. The corresponding logarithmic return is defined as

$$r_{\text{log}} := \ln\!\left(\frac{V_F}{V_I}\right) = \ln V_F - \ln V_I.$$

We now treat $r_{\log}$ as a random variable and assume a one-factor model

$$r_{\log} \;=\; \beta S + \xi, \tag{1}$$

where $\beta \in \mathbb{R}$, $S \sim \mathcal{N}(\mu_S, \sigma_S^2)$ with $\sigma_S^2 > 0$ is a *systematic* (macroeconomic) risk factor, and $\xi \sim \mathcal{N}(\mu_\xi, \sigma_\xi^2)$ with $\sigma_\xi^2 > 0$ is an *idiosyncratic* (obligor-specific) shock. The random variables S and ξ are assumed independent. The mean and variance of $r_{\log}$ (1) are then

$$\mu_r := \mathbb{E}[r_{\log}] = \beta\mu_S + \mu_\xi, \qquad \sigma_r^2 := \mathrm{Var}(r_{\log}) = \beta^2\sigma_S^2 + \sigma_\xi^2. \tag{2}$$

To obtain a standardized representation of $r_{\log}$ (1), define

$$Z_r := \frac{r_{\log} - \mu_r}{\sigma_r}, \qquad Z_S := \frac{S - \mu_S}{\sigma_S}, \qquad Z_\xi := \frac{\xi - \mu_\xi}{\sigma_\xi}.$$

Using (1) and the expressions for μ_r and σ_r^2 (2), we obtain

$$\begin{aligned} Z_r &= \frac{\beta S + \xi - (\beta\mu_S + \mu_\xi)}{\sigma_r} \\ &= \frac{\beta\big(S - \mu_S\big)}{\sigma_r} + \frac{\xi - \mu_\xi}{\sigma_r} \\ &= \beta\frac{\sigma_S}{\sigma_r}\,\frac{S - \mu_S}{\sigma_S} + \frac{\sigma_\xi}{\sigma_r}\,\frac{\xi - \mu_\xi}{\sigma_\xi} \\ &= \beta\frac{\sigma_S}{\sigma_r}\,Z_S + \frac{\sigma_\xi}{\sigma_r}\,Z_\xi. \end{aligned}$$

To simplify notation and collect the impact of the common systematic factor into a single parameter, define

$$\rho := \left(\frac{\beta\sigma_S}{\sigma_r}\right)^2 = \frac{\beta^2\sigma_S^2}{\beta^2\sigma_S^2 + \sigma_\xi^2} \in [0,1), \qquad 1 - \rho = \left(\frac{\sigma_\xi}{\sigma_r}\right)^2.$$

Using parameter ρ, we can rewrite

$$Z_r \;=\; \sqrt{\rho}\,Z_S + \sqrt{1-\rho}\,Z_\xi, \tag{3}$$

which satisfies $Z_r \sim \mathcal{N}(0,1)$.[1] The parameter ρ is referred to as the *asset correlation* [20] - it measures the fraction of the variance of the log–return that is explained by the systematic factor S.

Proposition 3.2.1 (see [9]) *The correlation between Z_r and Z_S equals $\sqrt{\rho}$, i.e.*

$$\mathrm{corr}(Z_r, Z_S) \;=\; \sqrt{\rho}.$$

[1] Since Z_S and Z_ξ are independent standard normal variables, any linear combination $aZ_S + bZ_\xi$ is again normally distributed, with mean 0 and variance $a^2 + b^2$, see Ross [17, Section 6.3.3, Proposition 3.2]. In this case $\mathrm{Var}(Z_r) = \rho + (1-\rho) = 1$ and hence $Z_r \sim \mathcal{N}(0,1)$.

Proof. Indeed, using the standardized representation (3), we obtain

$$\begin{aligned}
\operatorname{corr}(Z_r, Z_S) &= \frac{\operatorname{Cov}(Z_r, Z_S)}{\sigma_{Z_r}\,\sigma_{Z_S}} \\
&= \frac{\operatorname{Cov}\Big(\sqrt{\rho}\,Z_S + \sqrt{1-\rho}\,Z_\xi,\ Z_S\Big)}{\sigma_{Z_r}\,\sigma_{Z_S}} \\
&= \frac{\sqrt{\rho}\operatorname{Cov}(Z_S, Z_S) + \sqrt{1-\rho}\operatorname{Cov}(Z_\xi, Z_S)}{\sigma_{Z_r}\,\sigma_{Z_S}} \\
&= \frac{\sqrt{\rho}\operatorname{Var}(Z_S) + \sqrt{1-\rho}\cdot 0}{\sigma_{Z_r}\,\sigma_{Z_S}} \\
&= \frac{\sqrt{\rho}\cdot 1}{1\cdot 1} \\
&= \sqrt{\rho}.
\end{aligned}$$

□

We now link this standardized log-return (3) representation to the default event. Let $p \in (0,1)$ be the one-year default probability of a single obligor, and denote

$$x_p := \Phi^{-1}(p),$$

where Φ is the standard normal distribution function (see **Definition 3.1.2**). For a single obligor, we model default event by the indicator variable

$$D = \begin{cases} 1, & \text{if } \sqrt{\rho}\,Z_S + \sqrt{1-\rho}\,Z_\xi < x_p, \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

we obtain

$$\mathbb{P}(D = 1) = \mathbb{P}(Z_r < x_p) = \Phi(x_p) = p,$$

so $D \sim \text{Bernoulli}(p)$. Conditioning on a realization $Z_S = y$ of the macroeconomic factor, the conditional probability of default is

$$\begin{aligned}
\mathbb{P}(D = 1 \mid Z_S = y) &= \mathbb{P}\Big(\sqrt{\rho}\,Z_S + \sqrt{1-\rho}\,Z_\xi < x_p \,\Big|\, Z_S = y\Big) \\
&= \mathbb{P}\Big(\sqrt{\rho}\,y + \sqrt{1-\rho}\,Z_\xi < x_p \,\Big|\, Z_S = y\Big) \\
&= \mathbb{P}\Big(\sqrt{1-\rho}\,Z_\xi < x_p - \sqrt{\rho}\,y \,\Big|\, Z_S = y\Big) \\
&= \mathbb{P}\Big(Z_\xi < \frac{x_p - \sqrt{\rho}\,y}{\sqrt{1-\rho}} \,\Big|\, Z_S = y\Big) \\
&= \Phi\Big(\frac{\Phi^{-1}(p) - \sqrt{\rho}\,y}{\sqrt{1-\rho}}\Big),
\end{aligned} \tag{5}$$

and hence

$$\mathbb{P}(D = 0 \mid Z_S = y) = 1 - \Phi\left(\frac{\Phi^{-1}(p) - \sqrt{\rho}\, y}{\sqrt{1-\rho}}\right).$$

The random variable

$$\Phi\left(\frac{\Phi^{-1}(p) - \sqrt{\rho}\, Z_S}{\sqrt{1-\rho}}\right), \qquad Z_S \sim \mathcal{N}(0,1),$$

is known as the *Vasicek* distribution [19].

3.3 Distribution of the portfolio defaults count

Consider now a homogeneous portfolio of $n \in \mathbb{N}$ obligors with same probabilities of default p. Given a realisation $Z_S = y$ of the systematic factor, let $D_1, \ldots, D_n$ be conditionally independent copies of D defined in (4). The total number of defaults

$$\tilde{D} := D_1 + \cdots + D_n$$

is then conditional on $Z_S = y$ and binomially distributed with parameters n and success probability $\mathbb{P}(D = 1 \mid Z_S = y)$ given in (5). Hence by **Definition 3.1.1** the conditional probability of observing k defaults is

$$\begin{aligned} \mathbb{P}(\tilde{D} = k \mid Z_S = y) = \binom{n}{k} &\left[\Phi\left(\frac{\Phi^{-1}(p) - \sqrt{\rho}\, y}{\sqrt{1-\rho}}\right)\right]^k \\ &\times \left[1 - \Phi\left(\frac{\Phi^{-1}(p) - \sqrt{\rho}\, y}{\sqrt{1-\rho}}\right)\right]^{n-k}, \qquad k = 0, \ldots, n. \end{aligned} \tag{6}$$

By the law of total probability **Definition 3.1.3**, the unconditional probability of observing at most $k \in \{0, 1, \ldots, n\}$ defaults is expressed as

$$\begin{aligned} \mathbb{P}(\tilde{D} \le k) &= \mathbb{E}\Big[\mathbb{P}(\tilde{D} \le k \mid Z_S)\Big] \\ &= \int_{-\infty}^{\infty} \varphi(x) \sum_{i=0}^{k} \binom{n}{i} \left[\Phi\left(\frac{\Phi^{-1}(p) - \sqrt{\rho}\, x}{\sqrt{1-\rho}}\right)\right]^i \left[1 - \Phi\left(\frac{\Phi^{-1}(p) - \sqrt{\rho}\, x}{\sqrt{1-\rho}}\right)\right]^{n-i} dx, \end{aligned} \tag{7}$$

where φ is the standard normal density.

3.4 Estimation of the one-period probability of default

In this subsection the estimators for the one-period PD are presented. First, following [14] the one-sided upper confidence bound is introduced. Then Bayesian one-period PD estimators under conservative and neutral priors, following the Bayesian framework of [18] are presented.

3.4.1 Upper confidence bound

To estimate probability of default p, we construct a one-sided upper confidence bound that captures values of p at selected confidence level. We fix $\gamma \in (0,1)$ as the confidence level. Intuitively, γ describes how sure we want to be that the true value p does not exceed the estimated bound, while the remaining probability $1-\gamma$ can be viewed as a tolerated type I error, namely the risk that the true p is larger than the bound. We therefore call p *compatible at confidence level* γ if

$$\mathbb{P}_p(\tilde{D} \le k) \;\ge\; 1-\gamma.$$

The one-sided upper bound for the PD is defined as the supremum of all compatible values of p

$$\hat{p}^{\mathrm{UCB}} := \sup\{\, p \in (0,1) : \mathbb{P}_p(\tilde{D} \le k) \ge 1-\gamma \,\}. \tag{8}$$

3.4.2 Bayesian approach

With Bayesian approach the likelihood of the observed default count is combined with a prior distribution for the unknown PD to obtain a posterior distribution for it. The posterior distribution summarizes both the information contained in the data and any prior beliefs about plausible PD values. The posterior is represented by its mean. Following Tasche [18], we consider two priors: a *conservative* prior, which gives relatively more weight to larger PD values, and a *neutral* prior, which is uniform on interval $(0,1)$ and therefore treats all PD values in that range as equally possible.

Let the conditional probability of observing exactly k defaults, given a realisation $Z_S = y$ of the systematic factor, be defined as in (6). Then, by the law of total probability (see **Definition 3.1.3**) the unconditional probability of observing exactly k defaults is

$$\begin{aligned}\mathbb{P}_p(\tilde{D} = k) &= \mathbb{E}\Big[\,\mathbb{P}_p(\tilde{D} = k \mid Z_S)\,\Big] \\ &= \int_{-\infty}^{\infty} \varphi(x) \binom{n}{k} \left[\Phi\!\left(\frac{\Phi^{-1}(p) - \sqrt{\rho}\,x}{\sqrt{1-\rho}}\right)\right]^{k} \left[1 - \Phi\!\left(\frac{\Phi^{-1}(p) - \sqrt{\rho}\,x}{\sqrt{1-\rho}}\right)\right]^{n-k} dx,\end{aligned} \tag{9}$$

where φ is the standard normal density (see **Definition 3.1.2**).

We now treat the probability of default $0 < p < 1$ as a random variable P. To build intuition for the Bayesian update that will later be used in **Proposition 3.4.1**, we first consider a setting with a proper prior density function.

We start by assuming $P \sim U(0,1)$, i.e. all values in interval $(0,1)$ are equally possible. By fixing $p_0 \in (0,1)$ and imposing restriction $\{P \ge p_0\}$, for $p_0 \le u \le 1$, we have the conditional distribution function of a random variable P

$$\mathbb{P}(P \le u \mid P \ge p_0) = \frac{\mathbb{P}(p_0 \le P \le u)}{\mathbb{P}(P \ge p_0)} = \frac{u - p_0}{1 - p_0}.$$

Hence the conditional density is constant on the interval $(p_0, 1)$,

$$\pi(u) := f_{P|P\geq p_0}(u) = \frac{1}{1-p_0}, \qquad p_0 < u < 1. \tag{10}$$

Further, we assume conditionally independent defaults

$$\tilde{D} \mid (P = u) \sim \text{Bin}(n, u), \qquad \mathbb{P}_u(\tilde{D} = k) = \binom{n}{k} u^k (1-u)^{n-k}.$$

Here, the default distribution is assumed to follow the binomial probability mass function (see **Definition 3.1.1**) for simplicity. Then by the law of total probability (see **Definition 3.1.3**) the unconditional (posterior) distribution of $\tilde{D}$ under the prior (10) is a mixture distribution

$$\begin{aligned}
\mathbb{P}(\tilde{D} = k \mid P \geq p_0) &= \int_{p_0}^{1} \mathbb{P}_u(\tilde{D} = k)\, \pi(u)\, du \\
&= \int_{p_0}^{1} \frac{\binom{n}{k} u^k (1-u)^{n-k}}{1-p_0}\, du \\
&= \frac{\binom{n}{k}}{1-p_0} \int_{p_0}^{1} u^k (1-u)^{n-k}\, du \\
&= \frac{\binom{n}{k}}{1-p_0} \left(\int_0^1 u^k (1-u)^{n-k}\, du - \int_0^{p_0} u^k (1-u)^{n-k}\, du \right) \\
&= \frac{\binom{n}{k}}{1-p_0} \left(\int_0^1 u^{(k+1)-1} (1-u)^{(n-k+1)-1}\, du - \int_0^{p_0} u^{(k+1)-1} (1-u)^{(n-k+1)-1}\, du \right) \\
&= \frac{\binom{n}{k}}{1-p_0} \Big[B(k+1, n-k+1) - B_{p_0}(k+1, n-k+1) \Big] \\
&= \frac{\binom{n}{k}}{1-p_0} \left[-\, B_{p_0}(k+1, n-k+1) + \frac{\Gamma(k+1)\Gamma(n-k+1)}{\Gamma(n+2)} \right] \\
&= \frac{\binom{n}{k}}{1-p_0} \left[-\, B_{p_0}(k+1, n-k+1) + \frac{k!(n-k)!}{(n+1)!} \right],
\end{aligned}$$

where $B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}\, dt = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$ is the Beta function, and $B_{p_0}(a,b) = \int_0^{p_0} t^{a-1}(1-t)^{b-1}\, dt$ is the incomplete Beta function. Then prior density is updated by multiplying it with the likelihood and normalising it

$$\begin{aligned}
\pi(u \mid k) &= \frac{\mathbb{P}_u(\tilde{D} = k)\, \pi(u)}{\displaystyle\int_{p_0}^{1} \mathbb{P}_q(\tilde{D} = k)\, \pi(q)\, dq} \\
&= \frac{\mathbb{P}_u(\tilde{D} = k)\, (1-p_0)^{-1}}{\displaystyle\int_{p_0}^{1} \mathbb{P}_q(\tilde{D} = k)\, (1-p_0)^{-1}\, dq},
\end{aligned} \tag{11}$$

where $p_0 < u < 1$ and $0 < p_0 < 1$. The denominator in (11) plays normalising role as it scales the numerator so that $\pi(u \mid k)$ integrates to one over the $(p_0, 1)$, i.e. $\int_{p_0}^{1} \pi(u \mid k)\, du = 1$. The **Proposition 3.4.1** uses exactly the same Bayes update for posterior density, but replaces prior density (10) by Tasche's [18] improper conservative prior density on $(0, 1)$.

Conservative prior

Following Tasche [18], conservatism of random variable P is encoded directly by upweighting larger PD values through the improper prior density

$$\pi_{\text{con}}(p) \propto \frac{1}{1-p}, \qquad 0 < p < 1. \tag{12}$$

It assigns increasing weight as $p \to 1$, therefore leads to conservative posterior estimates. [2]

Proposition 3.4.1 *The posterior probability density of the random variable P given $\tilde{D} = k$ under the conservative prior is*

$$\pi_{\text{con}}(p \mid k) \;=\; \frac{\mathbb{P}_p(\tilde{D} = k)\,(1-p)^{-1}}{\displaystyle\int_0^1 \mathbb{P}_q(\tilde{D} = k)\,(1-q)^{-1}\,dq}, \qquad 0 < p < 1.$$

The corresponding posterior mean (see [18]) is

$$\hat{p}_{\text{con}} \;:=\; \mathbb{E}[P \mid \tilde{D} = k] \;=\; \frac{\displaystyle\int_0^1 p\,\mathbb{P}_p(\tilde{D} = k)\,(1-p)^{-1}\,dp}{\displaystyle\int_0^1 \mathbb{P}_q(\tilde{D} = k)\,(1-q)^{-1}\,dq}. \tag{13}$$

Proof. By conditional density of mixed random variables (see **Definition 3.1.4**) we have

$$\begin{aligned}
\pi_{\text{con}}(p \mid k) &= f_{P|\tilde{D}}(p \mid k) \\
&= \frac{f_{P,\tilde{D}}(p, k)}{f_{\tilde{D}}(k)} \\
&= \frac{\mathbb{P}(\tilde{D} = k \mid P = p)\,\pi_{\text{con}}(p)}{\displaystyle\int_0^1 \mathbb{P}(\tilde{D} = k \mid P = q)\,\pi_{\text{con}}(q)\,dq} \\
&= \frac{\mathbb{P}_p(\tilde{D} = k)\,\pi_{\text{con}}(p)}{\displaystyle\int_0^1 \mathbb{P}_q(\tilde{D} = k)\,\pi_{\text{con}}(q)\,dq} \\
&= \frac{\mathbb{P}_p(\tilde{D} = k)\,(1-p)^{-1}}{\displaystyle\int_0^1 \mathbb{P}_q(\tilde{D} = k)\,(1-q)^{-1}\,dq}.
\end{aligned}$$

[2] The probability density function defined in (12) is not integrable on $(0, 1)$, since $\int_0^1 \frac{1}{1-p}\,dp = \infty$, and is therefore considered as an improper prior. Here, the notation $\propto$ means "proportional to". Nevertheless, the posterior probability density in **Proposition 3.4.1** is well defined as the normalising integral in the denominator is finite.

The corresponding posterior mean is given by

$$\begin{aligned}
\hat{p}_{\text{con}} &= \mathbb{E}[P \mid \tilde{D} = k] \\
&= \int_0^1 p\, \pi_{\text{con}}(p \mid k)\, dp \\
&= \int_0^1 p \, \frac{\mathbb{P}_p(\tilde{D} = k)\, (1-p)^{-1}}{\displaystyle\int_0^1 \mathbb{P}_q(\tilde{D} = k)\, (1-q)^{-1}\, dq}\, dp \\
&= \frac{\displaystyle\int_0^1 p\, \mathbb{P}_p(\tilde{D} = k)\, (1-p)^{-1}\, dp}{\displaystyle\int_0^1 \mathbb{P}_q(\tilde{D} = k)\, (1-q)^{-1}\, dq}.
\end{aligned}$$

□

Neutral prior

As a less informative alternative, we consider a neutral prior probability density which is uniform on an interval $(0, u)$ with $0 < u \leq 1$

$$\pi_{\text{neu}}(p; u) = \begin{cases} \dfrac{1}{u}, & 0 < p < u, \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

The parameter u can be used to set an upper bound on reasonable PD values. The special case $u = 1$ is the unconstrained uniform prior on $(0, 1)$. Compared with the conservative prior, neutral prior probability density reflects the prior belief that all PD values in $(0, u)$ are equally reasonable.

Proposition 3.4.2 *The posterior density of the random variable P given $\tilde{D} = k$ under the neutral prior on $(0, u)$ is*

$$\pi_{\text{neu}}(p \mid k; u) \;=\; \frac{\mathbb{P}_p(\tilde{D} = k)}{\displaystyle\int_0^u \mathbb{P}_q(\tilde{D} = k)\, dq}, \qquad 0 < p < u.$$

The corresponding posterior mean (see [18]) is

$$\hat{p}_{\text{neu}}(u) \;:=\; \mathbb{E}[P \mid \tilde{D} = k; u] \;=\; \frac{\displaystyle\int_0^u p\, \mathbb{P}_p(\tilde{D} = k)\, dp}{\displaystyle\int_0^u \mathbb{P}_q(\tilde{D} = k)\, dq}. \tag{15}$$

Proof. Again, by conditional density of mixed random variables (see **Definition 3.1.4**) we have

$$\begin{aligned}
\pi_{\text{neu}}(p \mid k; u) &= f_{P|\tilde{D}}(p \mid k; u) \\
&= \frac{f_{P,\tilde{D}}(p, k; u)}{f_{\tilde{D}}(k; u)} \\
&= \frac{\mathbb{P}(\tilde{D} = k \mid P = p)\, \pi_{\text{neu}}(p; u)}{\displaystyle\int_0^u \mathbb{P}(\tilde{D} = k \mid P = q)\, \pi_{\text{neu}}(q; u)\, dq} \\
&= \frac{\mathbb{P}_p(\tilde{D} = k)\, \pi_{\text{neu}}(p; u)}{\displaystyle\int_0^u \mathbb{P}_q(\tilde{D} = k)\, \pi_{\text{neu}}(q; u)\, dq} \\
&= \frac{\mathbb{P}_p(\tilde{D} = k)\, u^{-1}}{\displaystyle\int_0^u \mathbb{P}_q(\tilde{D} = k)\, u^{-1}\, dq} \\
&= \frac{\mathbb{P}_p(\tilde{D} = k)}{\displaystyle\int_0^u \mathbb{P}_q(\tilde{D} = k)\, dq}.
\end{aligned}$$

The corresponding posterior mean is given by

$$\begin{aligned}
\hat{p}_{\text{neu}}(u) &= \mathbb{E}[P \mid \tilde{D} = k; u] \\
&= \int_0^u p\, \pi_{\text{neu}}(p \mid k; u)\, dp \\
&= \int_0^u p\, \frac{\mathbb{P}_p(\tilde{D} = k)}{\displaystyle\int_0^u \mathbb{P}_q(\tilde{D} = k)\, dq}\, dp \\
&= \frac{\displaystyle\int_0^u p\, \mathbb{P}_p(\tilde{D} = k)\, dp}{\displaystyle\int_0^u \mathbb{P}_q(\tilde{D} = k)\, dq}.
\end{aligned}$$

□

3.5 Estimation of the multi-period probability of default

In this subsection one-period framework is extended to a multi-period setting by adapting both the upper confidence bound and the Bayesian estimators to this case.

For each year $t = 1, \ldots, T$ we observe the portfolio size n_t and the number of defaults k_t. Denoting the random default count in year t by X_t, the data consist of the time series of pairs

$$(n_1, k_1), \ldots, (n_T, k_T), \qquad k_t = X_t,\ t = 1, \ldots, T.$$

Now, systematic risk is driven by a vector of latent annual factors

$$S = (S_1, \ldots, S_T)^\top, \qquad S \sim \mathcal{N}\big(0, \Sigma_\vartheta\big), \tag{16}$$

where Σ_ϑ is a $T \times T$ correlation matrix with entries $(\Sigma_\vartheta)_{t\tau} = \vartheta^{|t-\tau|}$, $t, \tau = 1, \dots, T$, $0 \le \vartheta < 1$. Explicitly,

$$\Sigma_\vartheta = \begin{pmatrix} 1 & \vartheta & \vartheta^2 & \cdots & \vartheta^{T-1} \\ \vartheta & 1 & \vartheta & \cdots & \vartheta^{T-2} \\ \vartheta^2 & \vartheta & 1 & \cdots & \vartheta^{T-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vartheta^{T-1} & \vartheta^{T-2} & \vartheta^{T-3} & \cdots & 1 \end{pmatrix}.$$

Thus $\mathrm{corr}(S_t, S_\tau) = \vartheta^{|t-\tau|}$ captures correlation of the systematic factor across years. Of course, correlation between a pair of systematic factors decreases exponentially with the time lag.

Conditional on $S_t = s_t$, the default count in year t is binomial with success probability given by (5), for convenience we write

$$p_t(s_t) := G(p, \rho, s_t) = \Phi\left(\frac{\Phi^{-1}(p) - \sqrt{\rho}\, s_t}{\sqrt{1-\rho}}\right).$$

The joint unconditional likelihood of the observed default counts $(k_1, \dots, k_T)$ is then obtained by integrating out the latent factors S

$$\begin{aligned} \mathbb{P}_p[X_1 = k_1, \dots, X_T = k_T] = \int_{\mathbb{R}^T} \varphi_{\Sigma_\vartheta}(s_1, \dots, s_T) \prod_{t=1}^{T} \binom{n_t}{k_t} G(p, \rho, s_t)^{k_t} \\ \times [1 - G(p, \rho, s_t)]^{n_t - k_t}\, d(s_1, \dots, s_T). \end{aligned} \tag{17}$$

where $\varphi_{\Sigma_\vartheta}$ denotes the multivariate normal density with mean 0 and covariance matrix Σ_ϑ defined above.

3.5.1 Upper confidence bound

We denote the random total number of defaults over the T years as the sum $X = X_1 + \cdots + X_T$ and $k = k_1 + \cdots + k_T$ as the realised count of total defaults. For every year t, random default count follows a binomial distribution $X_t \sim \mathrm{Bin}(n_t, p_t)$. When portfolios (n_t) are large and default probabilities (p_t) are small with $\lambda_t = n_t p_t$ of moderate size, $\mathrm{Bin}(n_t, p_t)$ is well approximated by a Poisson distribution $\mathrm{Pois}(\lambda_t)$ (see [17, Section 4.7]). Under this approximation, the random total default count X conditional on the systematic factor path has approximately a Poisson distribution with intensity.

$$\Lambda_{T,\rho}(s) := \sum_{t=1}^{T} n_t\, p_t(s_t) = \sum_{t=1}^{T} n_t\, G(p, \rho, s_t).$$

Proposition 3.5.1 *(see [18]) For a given confidence level* $0 < \gamma < 1$*, an approximate one–sided upper confidence bound for PD is obtained as the solution of*

$$1 - \gamma = \mathbb{P}_p[X \le k] \;\approx\; \int_{\mathbb{R}^T} \varphi_{\Sigma_\vartheta}(s)\, \exp\Big(-\Lambda_{T,\rho}(s)\Big) \sum_{j=0}^{k} \frac{\Lambda_{T,\rho}(s)^j}{j!}\, ds. \tag{18}$$

Proof. For each year t and realisation s_t,

$$X_t \mid (S_t = s_t, P = p) \sim \mathrm{Bin}\Big(n_t, p_t(s_t)\Big) \approx \mathrm{Pois}\Big(n_t p_t(s_t)\Big).$$

Given $S = s$, the default counts $X_1, \dots, X_T$ are conditionally independent, hence

$$X \mid (S = s, P = p) = \sum_{t=1}^{T} X_t \approx \mathrm{Pois}(\Lambda_{T,\rho}(s))\,.$$

Then cumulative distribution function given realisation $S = s$ is

$$\mathbb{P}_p(X \le k \mid S = s) \approx e^{-\Lambda_{T,\rho}(s)} \sum_{j=0}^{k} \frac{\Lambda_{T,\rho}(s)^j}{j!}.$$

By the law of total probability **Definition 3.1.3**, the unconditional probability to observe at most k defaults over T years is

$$\mathbb{P}_p(X \le k) = \mathbb{E}\Big[\mathbb{P}_p(X \le k \mid S)\Big] \approx \int_{\mathbb{R}^T} \varphi_{\Sigma_\vartheta}(s)\, e^{-\Lambda_{T,\rho}(s)} \sum_{j=0}^{k} \frac{\Lambda_{T,\rho}(s)^j}{j!}\, ds,$$

which gives (18). Here $\varphi_{\Sigma_\vartheta}$ denotes the multivariate normal density with mean 0 and covariance matrix Σ_ϑ. $\square$

3.5.2 Bayesian approach

In the multi-period setting we use the same prior probability densities for the PD random variable P as in the one-period case, i.e., a conservative prior with probability density (12) and a neutral prior with probability density (14).

Proposition 3.5.2 *Under the conservative prior, the posterior density of P given $(X_1, \dots, X_T) = (k_1, \dots, k_T)$ is*

$$\pi_{\mathrm{con}}(p \mid k_1, \dots, k_T) = \frac{\mathbb{P}_p[X_1 = k_1, \dots, X_T = k_T]\,(1-p)^{-1}}{\displaystyle\int_0^1 \mathbb{P}_q[X_1 = k_1, \dots, X_T = k_T]\,(1-q)^{-1}\,dq}, \qquad 0 < p, q < 1.$$

The corresponding posterior mean (see [18]) is

$$\hat{p}_{\mathrm{con,mult}} = \frac{\displaystyle\int_0^1 p\,\mathbb{P}_p[X_1 = k_1, \dots, X_T = k_T]\,(1-p)^{-1}\,dp}{\displaystyle\int_0^1 \mathbb{P}_q[X_1 = k_1, \dots, X_T = k_T]\,(1-q)^{-1}\,dq}. \qquad (19)$$

Proof. By conditional density of mixed random variables (see **Definition 3.1.4**) we have

$$\begin{aligned}
\pi_{\text{con}}(p \mid k_1, \dots, k_T) &= f_{P|X_1,\dots,X_T}(p \mid k_1, \dots, k_T) \\
&= \frac{f_{P,X_1,\dots,X_T}(p, k_1, \dots, k_T)}{f_{X_1,\dots,X_T}(k_1, \dots, k_T)} \\
&= \frac{\mathbb{P}_p[X_1 = k_1, \dots, X_T = k_T]\, \pi_{\text{con}}(p)}{\displaystyle\int_0^1 \mathbb{P}_q[X_1 = k_1, \dots, X_T = k_T]\, \pi_{\text{con}}(q)\, dq} \\
&= \frac{\mathbb{P}_p[X_1 = k_1, \dots, X_T = k_T]\, (1-p)^{-1}}{\displaystyle\int_0^1 \mathbb{P}_q[X_1 = k_1, \dots, X_T = k_T]\, (1-q)^{-1}\, dq}.
\end{aligned}$$

The corresponding posterior mean is given by

$$\begin{aligned}
\hat{p}_{\text{con,mult}} &= \mathbb{E}[P \mid X_1 = k_1, \dots, X_T = k_T] \\
&= \int_0^1 p\, \pi_{\text{con}}(p \mid k_1, \dots, k_T)\, dp \\
&= \int_0^1 p\, \frac{\mathbb{P}_p[X_1 = k_1, \dots, X_T = k_T]\, \pi_{\text{con}}(p)}{\displaystyle\int_0^1 \mathbb{P}_q[X_1 = k_1, \dots, X_T = k_T]\, \pi_{\text{con}}(q)\, dq}\, dp \\
&= \frac{\displaystyle\int_0^1 p\, \mathbb{P}_p[X_1 = k_1, \dots, X_T = k_T]\, (1-p)^{-1}\, dp}{\displaystyle\int_0^1 \mathbb{P}_q[X_1 = k_1, \dots, X_T = k_T]\, (1-q)^{-1}\, dq}.
\end{aligned}$$

□

Proposition 3.5.3 *Under the neutral prior, the posterior density of* P *given* $(X_1, \dots, X_T) = (k_1, \dots, k_T)$ *is*

$$\pi_{\text{neu}}(p \mid k_1, \dots, k_T; u) = \frac{\mathbb{P}_p[X_1 = k_1, \dots, X_T = k_T]}{\displaystyle\int_0^u \mathbb{P}_q[X_1 = k_1, \dots, X_T = k_T]\, dq}, \qquad 0 < p < u \leq 1.$$

The corresponding posterior mean (see [18]) is

$$\hat{p}_{\text{neu,mult}} = \frac{\displaystyle\int_0^u p\, \mathbb{P}_p[X_1 = k_1, \dots, X_T = k_T]\, dp}{\displaystyle\int_0^u \mathbb{P}_p[X_1 = k_1, \dots, X_T = k_T]\, dp}. \tag{20}$$

Proof. Again, by conditional density of mixed random variables (see **Definition 3.1.4**) we have

$$\begin{aligned}
\pi_{\text{neu}}(p \mid k_1, \dots, k_T; u) &= \frac{\mathbb{P}_p[X_1 = k_1, \dots, X_T = k_T]\, \pi_{\text{neu}}(p; u)}{\displaystyle\int_0^u \mathbb{P}_q[X_1 = k_1, \dots, X_T = k_T]\, \pi_{\text{neu}}(q; u)\, dq} \\
&= \frac{\mathbb{P}_p[X_1 = k_1, \dots, X_T = k_T]\, u^{-1}}{\displaystyle\int_0^u \mathbb{P}_q[X_1 = k_1, \dots, X_T = k_T]\, u^{-1}\, dq} \\
&= \frac{\mathbb{P}_p[X_1 = k_1, \dots, X_T = k_T]}{\displaystyle\int_0^u \mathbb{P}_q[X_1 = k_1, \dots, X_T = k_T]\, dq},
\end{aligned}$$

The corresponding posterior mean is given by

$$\begin{aligned}
\hat{p}_{\text{neu,mult}} &= \mathbb{E}[P \mid X_1 = k_1, \dots, X_T = k_T] \\
&= \int_0^1 p\, \pi_{\text{neu}}(p \mid k_1, \dots, k_T; u)\, dp \\
&= \int_0^1 p\, \frac{\mathbb{P}_p[X_1 = k_1, \dots, X_T = k_T]}{\displaystyle\int_0^u \mathbb{P}_q[X_1 = k_1, \dots, X_T = k_T]\, dq} dp \\
&= \frac{\displaystyle\int_0^1 p\, \mathbb{P}_p[X_1 = k_1, \dots, X_T = k_T]\, dp}{\displaystyle\int_0^1 \mathbb{P}_p[X_1 = k_1, \dots, X_T = k_T]\, dq}.
\end{aligned}$$

□

The unconditional probabilities in (18), (19) and (20) involve the multivariate normal integral (17). In order to avoid difficult calculations we approximate the multi-variate integral by Monte-Carlo simulation (see **Definition 3.1.5**). We generate n independent draws of jointly normally distributed systematic factors (16)

$$s^{(j)} = (s_1^{(j)}, \dots, s_T^{(j)})^\top, \qquad j = 1, \dots, n,$$

from $\mathcal{N}(0, \Sigma_\vartheta)$ and approximate probability (17) by

$$\mathbb{P}_p[X_1 = k_1, \dots, X_T = k_T] \approx \frac{1}{n} \sum_{j=1}^{n} \prod_{t=1}^{T} \binom{n_t}{k_t} G(p, \rho, s_t^{(j)})^{k_t} \Big[1 - G(p, \rho, s_t^{(j)})\Big]^{n_t - k_t}. \tag{21}$$

The right-hand side of (18) is approximated in the same way by replacing the integral over s with the empirical average over the simulated factor paths. Following Tasche [18], we employ a simple quadrature scheme. For fixed $0 < u \leq 1$ choose a positive integer m and define grid points

$$u_i := \frac{i}{m} u, \qquad i = 0, 1, \dots, m.$$

Combining the Monte Carlo approximation (21) with Riemann sums then yields practical estimators

$$\hat{p}_{\text{con,mult}} \approx \frac{\sum_{i=0}^{m-1} u_i (1-u_i)^{-1} \sum_{j=1}^{n} \prod_{t=1}^{T} G(u_i,\rho,s_t^{(j)})^{k_t} \left[1 - G(u_i,\rho,s_t^{(j)})\right]^{n_t-k_t}}{\sum_{i=0}^{m-1} (1-u_i)^{-1} \sum_{j=1}^{n} \prod_{t=1}^{T} G(u_i,\rho,s_t^{(j)})^{k_t} \left[1 - G(u_i,\rho,s_t^{(j)})\right]^{n_t-k_t}}, \tag{22}$$

$$\hat{p}_{\text{neu,mult}} \approx \frac{\sum_{i=0}^{m} u_i \sum_{j=1}^{n} \prod_{t=1}^{T} G(u_i,\rho,s_t^{(j)})^{k_t} \left[1 - G(u_i,\rho,s_t^{(j)})\right]^{n_t-k_t}}{\sum_{i=0}^{m} \sum_{j=1}^{n} \prod_{t=1}^{T} G(u_i,\rho,s_t^{(j)})^{k_t} \left[1 - G(u_i,\rho,s_t^{(j)})\right]^{n_t-k_t}}. \tag{23}$$

3.6 Most prudent principle

In practice, obligors in portfolios are grouped into finite number of risk classes $RC = 1, \dots, G$, ordered from the best creditworthiness ($RC = 1$) to the worst ($RC = G$). Each risk class is associated with a one-year PD denoted by $p_{RC} \in (0, 1)$. It is assumed that risk classes are ordered correctly, i.e.

$$p_1 \leq p_2 \leq \cdots \leq p_G. \tag{24}$$

To enforce monotonicity in risk classes (24) we apply the *most prudent principle* presented by Pluto and Tasche [14]. The idea is for each risk class to choose the most conservative PDs that still respect the ordering of the grades. When estimating p_1 (the best grade), we assume that all risk classes share the same PD within portfolio, that is

$$p_1 = p_2 = \cdots = p_G.$$

Here we treat the whole portfolio as a single homogeneous risk class with total number of obligors $n = \sum_{RC=1}^{G} n_{RC}$ and total number of defaults $k = \sum_{RC=1}^{G} k_{RC}$.

For an intermediate risk class $RC \in \{2, \dots, G-1\}$, the most prudent assumption is that risk class RC has the same PD as all worse grades, i.e.

$$p_{RC} = p_{RC+1} = \cdots = p_G,$$

while still satisfying $p_{RC-1} \leq p_{RC}$. Hence obligors and default counts of risk classes are pooled to form a homogeneous sample of size $n^{(RC)} = \sum_{h=RC}^{G} n_h$ with $k^{(RC)} = \sum_{h=RC}^{G} k_h$ defaults. For the worst grade G, the PD is estimated using only its own number of obligors and defaults (n_G, k_G).

The presented *most prudent principle* is used to estimate PDs of individual risk classes in this thesis.

3.7 Validation measures and scoring framework

As the goal is to use the LDP estimators as validation benchmarks, two complementary types of validation measures are applied to assess the conservatism of the PD estimates that are under validation:

1. Log–ratio measure based on upper confidence bounds (25);
2. Bayesian tail probability (26).

In both cases, the raw validation quantities are mapped to a common $[0, 3]$ score scale and summarised using a traffic light classification (green, yellow, red). This is motivated by the Basel Committee's three zone framework for the supervisory interpretation of backtesting results, where outcomes are classified into green (no concern), yellow (potential concern) and red (strong indication of a problem) zones depending on how strongly the results deviate from expectations [5]. The benefit of scoring is that it puts different validation outputs on one common scale, making comparisons across portfolios, years and risk classes easy.

Log-ratio measure

Let UCB_α denote the upper confidence bound estimate of PD at selected confidence level α (e.g. $\alpha = 75\%, 90\%, 99\%$), and let p be the corresponding PD under validation. An easy way to compare these two quantities is by log–ratio

$$R_\alpha \;=\; \log\left(\frac{\text{UCB}_\alpha}{p}\right), \qquad \text{where } p \neq 0. \tag{25}$$

This defined measure is dimensionless and is interpretable in such terms:

- $R_\alpha < 0$ if and only if $\text{UCB}_\alpha < p$, i.e. the PD lies above the UCB_α and is therefore conservative compared to this benchmark estimate.
- $R_\alpha = 0$ if and only if $\text{UCB}_\alpha = p$, meaning that the PD sits exactly on the upper confidence bound estimate.
- $R_\alpha > 0$ if and only if $\text{UCB}_\alpha > p$, indicating that the data are compatible with PD values larger than the PD estimate that is under validation at confidence level α and therefore suggesting potential underestimation compared to this benchmark estimate.

In this thesis, a logarithm of a ratio as a validation measure was used instead of a simple ratio, because it has two advantages. The first one is that it makes the measure symmetric on a multiplicative scale. For example, a situation where the UCB estimate is twice the PD under validation and one where it is one half of the PD under validation correspond to log–ratios of the same magnitude but opposite sign. Second, measure does not explode when PDs under validation become small. Moreover, it is easily interpretable and does not overcomplicate quantification of differences.

Bayesian tail probability

The second validation measure is based on the posterior probability density of the PD. Let λ denote the unknown PD, and let $\pi(\lambda \mid \text{data})$ be the posterior probability density which is obtained under either conservative (12) or neutral (14) prior. A natural Bayesian quantity for validation is the tail probability

$$q \;=\; \mathbb{P}\Big(\lambda > p \,\Big|\, \text{data}\Big) \;=\; \int_p^1 \pi(\lambda \mid \text{data})\, d\lambda, \tag{26}$$

which directly answers the question: *given the observed defaults and the model assumptions, how likely is it that the true PD exceeds the PD that is under validation?*

The interpretation is straightforward:

- q close to 0 indicates that values above the PD under validation receive very little probability mass and appears conservative.
- Moderate values of q (e.g. 0.5–0.75) suggest that it is quite plausible that the true PD is larger than the PD under validation, raising some concern regarding underestimation of PD under validation.
- Large values of q (e.g. above 0.75) indicate strong posterior evidence that the PD under validation may be too optimistic.

Conversion to score and traffic light interpretation

For better interpretation of results, the raw validation quantities R_α and q are converted into scores on a common $[0, 3]$ scale. These scores are then mapped to traffic light colours, i.e. green, yellow, red, which is a common practice in model validation reporting.

For the log-ratio measure, we define scoring function as

$$\text{Score}(R_\alpha) \;=\; \begin{cases} 0, & \text{if } R_\alpha \le 0, \\ 3\,\dfrac{R_\alpha}{\log 3}, & \text{if } 0 < R_\alpha < \log 3, \\ 3, & \text{if } R_\alpha \ge \log 3. \end{cases}$$

The visual mapping is presented in Figure 1. This scoring function is introduced specifically for this thesis. A linear mapping is used for simplicity, this ensures that score increases proportionally. Any case where the PD under validation lies above or equals the UCB (i.e. $R_\alpha \leq 0$) receives a score of 0, reflecting the absence of potential underestimation from this perspective. Positive log–ratios are scaled linearly so that a situation where the UCB is three times larger than the PD under validation corresponds to the maximum score 3. Ratios larger than 3 are truncated at this maximum score. The treshold equal to 3 is selected, since a PD under validation that is three times lower than UCB estimate could be treated as a strong signal of potential underestimation, while more extreme gaps are not distinguished further and are simply classified in the worst category.

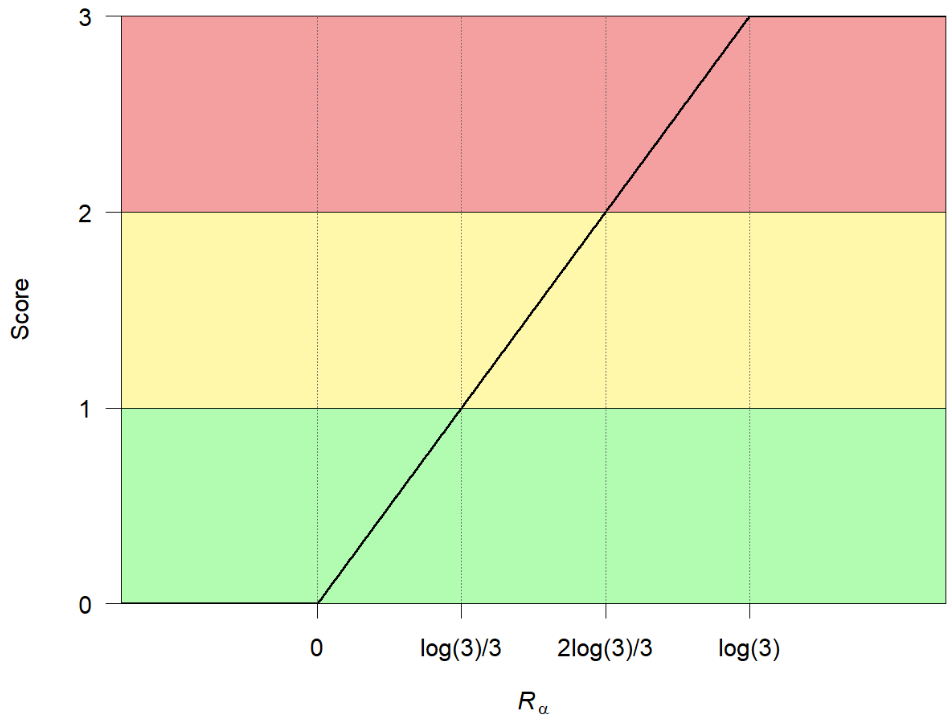


Figure 1: Score map to traffic light colors for log-ratio measure

For the Bayesian tail probability, we define scoring function as

$$\text{Score}(q) = \begin{cases} \dfrac{q}{0.5}, & \text{if } 0 \leq q \leq 0.5, \\ 1 + \dfrac{q - 0.5}{0.25}, & \text{if } 0.5 < q \leq 0.75, \\ 2 + \dfrac{q - 0.75}{0.25}, & \text{if } 0.75 < q \leq 1. \end{cases}$$

The visual mapping is presented in Figure 2. This mapping is also linear and reflects the following rationale. Tail probabilities up to 50% are considered acceptable, in this region score increases from 0 to 1 and is interpreted as green. When q exceeds 0.5, it becomes more likely than not that the true PD is above the PD under validation. Therefore, the interval $(0.5, 0.75]$ is treated as a yellow zone, with scores between 1 and 2. Tail probabilities above 0.75 correspond to a situation in which there is strong posterior evidence of underestimation and are mapped to scores between 2 and 3 which is treated as a red zone.

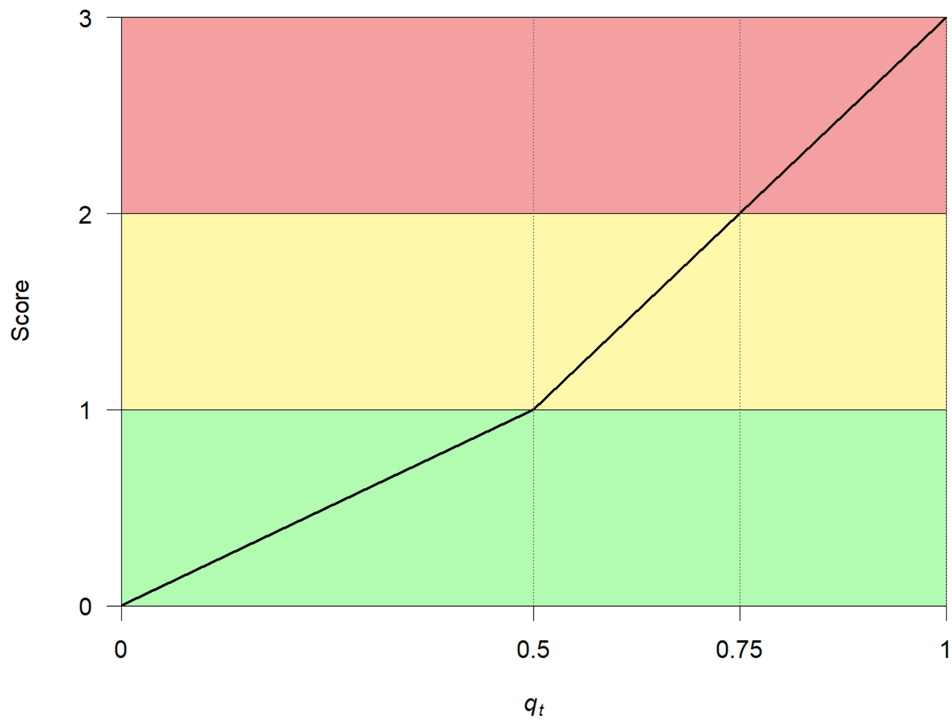


Figure 2: Score map to traffic light colors for Bayesian tail probability

In both cases, the score scale is interpreted through a traffic light scheme:

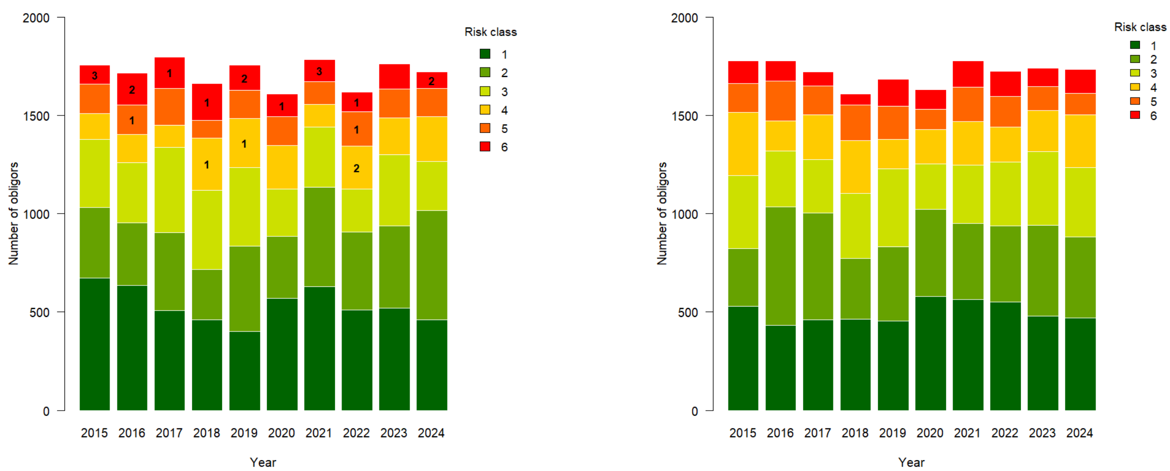
- Green: [0,1);
- Yellow: [1,2);
- Red: [2,3].

Presented thresholds are a practical choice and could be adjusted. The benefit of transformation to scores is that it puts different validation outputs on one standard scale, which makes comparisons across portfolios, years, and parameter settings easier.

4 Data

In this thesis simulated data is used, because real bank portfolios and their PD estimates are confidential and not publicly disclosed. Nevertheless, constructing realistic low default portfolios is not a challenging task, and the simulated portfolios are designed to represent situations that may occur in practice.

Two low default portfolios (A and B) were randomly generated, each consisting of six risk classes (1 - lowest risk, 6 - highest risk). For every risk class, counts of obligors and defaults were generated over a 10 year period, and a random PD was assigned, this PD is interpreted as the bank's internal PD estimate that is subject to validation, i.e. we want to draw conclusions whether this PD estimate is conservative enough. Of course, the riskier risk class is, the larger the PD is.



(a) Portfolio A (b) Portfolio B

Figure 3: Composition of simulated Portfolio A and Portfolio B by risk class and year. Numbers inside the bars are counts of defaults in a given risk class.

The distribution of each portfolio across risk classes is shown in Figure 3. In Portfolio A a few defaults are observed over the years, while in Portfolio B no defaults observed through the years.

5 Results

One-period estimators (8), (13), (15) and multiple period estimators (18), (19), (20) were applied to the simulated portfolios introduced in the data section (see **4 Data**). Upper confidence bounds were calculated for confidence levels $\gamma \in \{75\%, 90\%, 99\%\}$. All Estimates were computed for each year and for the full history, both at the whole portfolio level and at the risk class level. To calculate estimates at the risk class level, the *most prudent principle* was applied (see **3.6 Most prudent principle**). The quantitative validation of PD of

Portfolio A and Portfolio B was performed by using the validation measures and the scoring framework introduced in methodology section (see **3.7 Validation measures and scoring framework**).

Since both, one period as well as multiple period estimators, depend on ρ (asset correlation), simulations were performed for $\rho \in \{0.12, 0.24\}$. These values were chosen because they match the lower and upper bounds of the Basel II IRB supervisory asset correlation function for corporate exposures (including banks and sovereigns) [4]. By using these two values, a sensitivity range is provided, where $\rho = 0.12$ assumes weaker dependence on the common systematic factor, while $\rho = 0.24$ implies stronger dependence and therefore more clustered defaults. Intermediate values are also possible, but focusing on the endpoints shows how the validation conclusions change under low versus high dependence on the common systematic factor. Pluto & Tasche in their work [14] as well as Tasche in his work [18] consider $\rho \in \{0.12, 0.18\}$ in provided examples. In the multiple period setting, ϑ (correlation of the systematic factor across years) was fixed at 0.6. This choice follows the numerical example in Tasche's work [18], where $\vartheta = 0.6$ is used to represent persistence in macroeconomic conditions over time. Of course, other values of ρ and ϑ could also be considered, since these parameters are model inputs and may be chosen based on data availability, expert judgement, or sensitivity analysis.

5.1 Portfolio A

Figure 4 illustrates results at the whole portfolio level of Portfolio A. Besides UCB and Bayesian estimates, the internal PD estimate under validation and the observed default rate are also presented. Throughout the results subsection of Portfolio A (**5.1 Portfolio A**), this bank internal PD estimate of Portfolio A that is subject to validation is denoted by PD_A.

As we can see, UCB and Bayesian estimates remain non-zero in all years, including year 2023, where zero defaults were observed, illustrating their suitability for LDP settings where default rates can be zero and therefore uninformative. Moreover, estimates are consistently above the observed default rates, which is consistent with their conservative construction. The PD_A is consistently above the $\mathrm{UCB}_{75\%}$ and $\mathrm{UCB}_{90\%}$ as well as both Bayesian estimates. For the most conservative $\mathrm{UCB}_{99\%}$, the position is mixed, in several years UCB estimate exceeds the PD_A, while in other years it is below or close to it. When full history (years 2015-2024) is considered (see Figure 5), it can be noticed that all estimates increase with higher risk class which is consistent with the intended risk ranking. The PD_A remains above all estimates except for $\mathrm{UCB}_{99\%}$ in lowest risk class. Also, the distance between PD_A and all benchmarks estimates widens for higher risk classes, indicating that the largest margins occur in risk classes 4–6. Overall, no underestimation of PD_A is suggested by these figures when $\rho = 0.12$ is assumed.

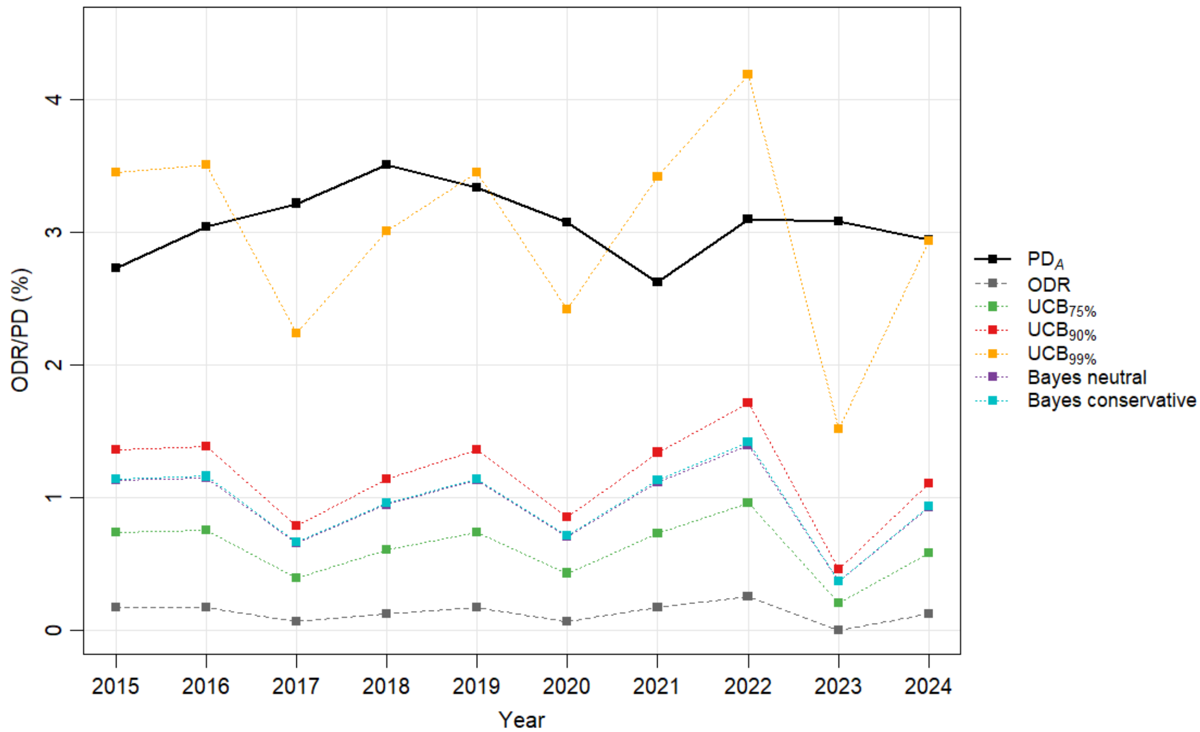


Figure 4: Annual results of Portfolio A, portfolio level, $\rho = 0.12$

For comparison, results are also provided under a higher asset correlation, i.e., $\rho = 0.24$, to assess how this assumption affects estimates. The corresponding plots for $\rho = 0.24$ (Figures 9 and 10 in Appendix B) show the same overall pattern as for $\rho = 0.12$, but with UCB and Bayesian estimates being shifted upwards. This is expected, since a higher asset correlation implies stronger co-movement in defaults and therefore leads to higher estimates, because more defaults are expected. At the portfolio level (see Figure 9 in Appendix B), UCB$_{99\%}$ is now above the PD$_A$ in all years. In addition, during years 2015-2016 and 2021-2022 the PD$_A$ is also below the Bayesian and UCB$_{90\%}$ estimates. This pattern suggests that, under the higher correlation assumption, i.e. $\rho = 0.24$, the PD$_A$ might be too low during those periods with respect to these benchmarks. However, PD$_A$ remains still higher than UCB$_{75\%}$. When full history is considered (see Figure 10 in Appendix B), the PD$_A$ exceeds all benchmark estimates from risk class 3 onwards, whereas for risk classes 1-2 the UCB$_{99\%}$ estimate remains above the PD$_A$.

The additional yearly plots by risk class for Portfolio A (Figures 11 and 12 in Appendix B) provide a more detailed view of the estimates at the risk class level. Across both correlation settings, the benchmark estimates are monotonic and increase with the risk class. The curves show slight variation over time, because defaults are observed. Of course, UCB$_{99\%}$ remains the most conservative estimate in all years and rises more strongly in the highest risk classes.

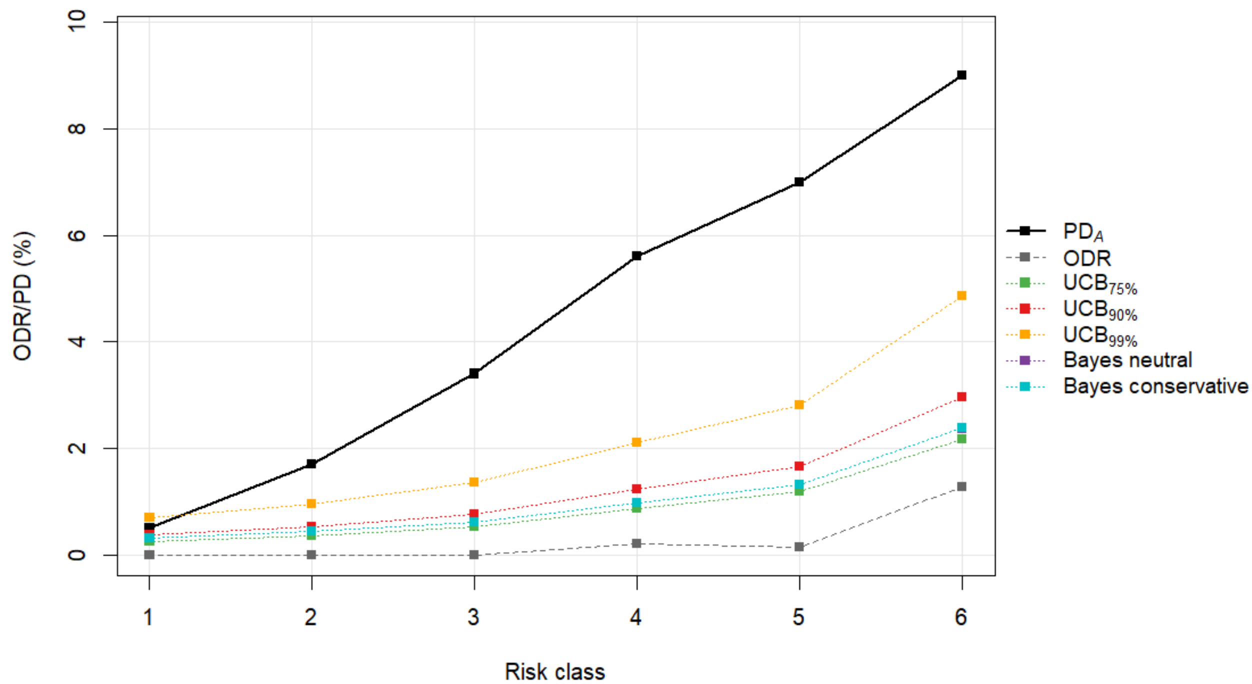


Figure 5: Total history results of Portfolio A, risk class level, $\rho = 0.12$

Finally, increasing ρ shifts all benchmark estimates upwards and reduces the distance to PD_A, but the overall ordering of the curves across risk classes remains the same.

Tables 1 and 2 present the quantified deviations between PD_A and benchmark estimates, corresponding scores are also provided. Since PD_A is above $\mathrm{UCB}_{75\%}$ and $\mathrm{UCB}_{90\%}$ in all years, the corresponding scores are zero and remain in the green zone. For $\mathrm{UCB}_{99\%}$, the log-ratio (R_{99}) is positive in several years, meaning that $\mathrm{UCB}_{99\%}$ exceeds PD_A in those periods, however, the differences are insignificant and therefore assessment remains green. The posterior tail probability measures (see Table 2) are also low under both priors (all below 0.1), suggesting that the posterior probability of the true PD exceeding PD_A is small, so it does not raise concerns regarding underestimation of PD_A. For the total history results (see Tables 3 and 4), scores are zero or close to zero, indicating that PD_A remains above or slightly below the benchmark estimates. In risk class 1, PD_A is below $\mathrm{UCB}_{99\%}$, this leads to a positive log-ratio (R_{99}) and a score close to being in a yellow zone. However, $\mathrm{UCB}_{99\%}$ is extremely conservative estimate, therefore the fact that PD_A is lower, but still higher than $\mathrm{UCB}_{90\%}$ does not imply that PD_A is optimistic in risk class 1. Overall, under $\rho = 0.12$, the validation results for Portfolio A do not indicate underestimation and indicate that PD_A is especially conservative when full history of portfolio is assessed.

Table 1: UCB estimates, quantified deviations and scores for Portfolio A at portfolio level, $\rho = 0.12$.

Year	PD_A (%)	$UCB_{75\%}$	R_{75}	Score(R_{75})	$UCB_{90\%}$	R_{90}	Score(R_{90})	$UCB_{99\%}$	R_{99}	Score(R_{99})
2015	2.726	0.736	-1.310	0.000	1.356	-0.698	0.000	3.446	0.234	0.639
2016	3.041	0.750	-1.399	0.000	1.379	-0.791	0.000	3.503	0.142	0.388
2017	3.213	0.391	-2.105	0.000	0.781	-1.415	0.000	2.232	-0.364	0.000
2018	3.507	0.600	-1.765	0.000	1.135	-1.128	0.000	3.006	-0.154	0.000
2019	3.331	0.736	-1.509	0.000	1.357	-0.898	0.000	3.448	0.035	0.096
2020	3.068	0.427	-1.972	0.000	0.852	-1.282	0.000	2.416	-0.239	0.000
2021	2.620	0.727	-1.282	0.000	1.337	-0.673	0.000	3.411	0.264	0.721
2022	3.097	0.953	-1.178	0.000	1.708	-0.595	0.000	4.181	0.300	0.819
2023	3.076	0.205	-2.707	0.000	0.459	-1.902	0.000	1.516	-0.708	0.000
2024	2.939	0.580	-1.623	0.000	1.103	-0.980	0.000	2.933	-0.002	0.000

Table 2: Bayesian estimates, quantified deviations and scores for Portfolio A at portfolio level, $\rho = 0.12$.

Year	PD_A (%)	$Bayes^{neut}$	q^{neut}	Score(q^{neut})	$Bayes^{cons}$	q^{cons}	Score(q^{cons})
2015	2.726	1.125	0.0801	0.160	1.139	0.0826	0.165
2016	3.041	1.143	0.0641	0.128	1.157	0.0663	0.133
2017	3.213	0.655	0.0166	0.033	0.661	0.0172	0.034
2018	3.507	0.944	0.0291	0.058	0.955	0.0303	0.061
2019	3.331	1.126	0.0491	0.098	1.140	0.0509	0.102
2020	3.068	0.704	0.0229	0.046	0.711	0.0238	0.048
2021	2.620	1.115	0.0859	0.172	1.128	0.0884	0.177
2022	3.097	1.393	0.0933	0.187	1.412	0.0964	0.193
2023	3.076	0.367	0.0059	0.012	0.370	0.0061	0.012
2024	2.939	0.922	0.0445	0.089	0.932	0.0460	0.092

For the higher asset correlation setting, i.e. $\rho = 0.24$, (see Tables 9 and 10 in Appendix B), the overall conclusions are broadly similar, $UCB_{75\%}$ remains below PD_A, so the corresponding scores are zero and stay in the green zone. Then, $UCB_{90\%}$ and the Bayesian estimates exceed PD_A in some years, however, the differences are not alarming and the scores remain in the green range. In contrast, $UCB_{99\%}$ is significantly higher than PD_A, which leads to red assessment scores in most years. For the total history results (see Tables 11 and 12 in Appendix B), the lowest risk class stands out. In risk class 1, PD_A is below $UCB_{90\%}$ and $UCB_{99\%}$, resulting in a score close to yellow for $UCB_{90\%}$ and a red score for $UCB_{99\%}$. The Bayesian tail probabilities for risk class 1 are about 0.55 under both priors and the corresponding scores fall in the yellow zone. Taken together, these results suggest a moderate underestimation of PD_A of the lowest risk class under $\rho = 0.24$ assumption.

To conclude, under $\rho = 0.12$ the validation results for Portfolio A do not indicate underestimation of PD_A. All scores remain in the green zone. Under $\rho = 0.24$, the annual portfolio level results remain largely acceptable relative to $UCB_{75\%}$, $UCB_{90\%}$ and the Bayesian benchmarks, while $UCB_{99\%}$ is substantially above PD_A. The main underestimation signal under $\rho = 0.24$

is concentrated in the total history results for lowest risk class, where validation measures point to moderate underestimation of PD_A.

Table 3: Full history UCB estimates, quantified deviations and scores for Portfolio A at risk class level, $\rho = 0.12$.

Risk class	PD_A (%)	$\mathrm{UCB}_{75\%}$	R_{75}	Score(R_{75})	$\mathrm{UCB}_{90\%}$	R_{90}	Score(R_{90})	$\mathrm{UCB}_{99\%}$	R_{99}	Score(R_{99})
1	0.500	0.258	-0.661	0.000	0.382	-0.269	0.000	0.705	0.344	0.940
2	1.700	0.365	-1.539	0.000	0.534	-1.147	0.000	0.965	-0.566	0.000
3	3.400	0.532	-1.855	0.000	0.767	-1.463	0.000	1.357	-0.919	0.000
4	5.600	0.870	-1.862	0.000	1.228	-1.470	0.000	2.110	-0.976	0.000
5	7.000	1.183	-1.778	0.000	1.655	-1.386	0.000	2.804	-0.915	0.000
6	9.000	2.164	-1.425	0.000	2.968	-1.033	0.000	4.860	-0.616	0.000
TOTAL	3.059	0.258	-2.472	0.000	0.382	-2.080	0.000	0.705	-1.467	0.000

Table 4: Full history Bayesian estimates, quantified deviations and scores for Portfolio A at risk class level, $\rho = 0.12$.

Risk class	PD_A (%)	$\mathrm{Bayes}^{\mathrm{neut}}$	q^{neut}	Score(q^{neut})	$\mathrm{Bayes}^{\mathrm{cons}}$	q^{cons}	Score(q^{cons})
1	0.500	0.319	0.1282	0.256	0.320	0.1287	0.257
2	1.700	0.441	0.0012	0.002	0.442	0.0012	0.002
3	3.400	0.620	0.0000	0.000	0.622	0.0000	0.000
4	5.600	0.981	0.0000	0.000	0.984	0.0000	0.000
5	7.000	1.316	0.0000	0.000	1.320	0.0000	0.000
6	9.000	2.376	0.0001	0.000	2.389	0.0001	0.000
TOTAL	3.059	0.319	0.0000	0.000	0.320	0.0000	0.000

5.2 Portfolio B

Figure 6 illustrates results at the portfolio level of Portfolio B. Besides UCB and Bayesian estimates, the internal PD estimate under validation and the observed default rate are also presented. Throughout the results subsection of Portfolio B (**5.2 Portfolio B**), this bank internal PD estimate of Portfolio B that is subject to validation is denoted by PD_B.

No defaults are observed in this portfolio in any year, so observed default rate is equal to zero throughout the years. As a result, UCB and Bayesian estimates remain closer to zero and vary only slightly over time, mainly reflecting changes in portfolio size rather than realised default events. As we can see, PD_B lies roughly about two times higher above $\mathrm{UCB}_{75\%}$, $\mathrm{UCB}_{90\%}$ and the Bayesian estimates in all years, while the most conservative $\mathrm{UCB}_{99\%}$ remains above PD_B. In the whole history (years 2015-2024) (see Figure 7) all estimates increase with higher risk class. Also, PD_B is above all benchmark estimates for all risk classes, with the gap becoming larger in higher risk classes. Overall, these figures suggest no underestimation of PD_B under $\rho = 0.12$ and indicate that PD_B is extremely conservative when full history is assessed since it is above all benchmark estimates.

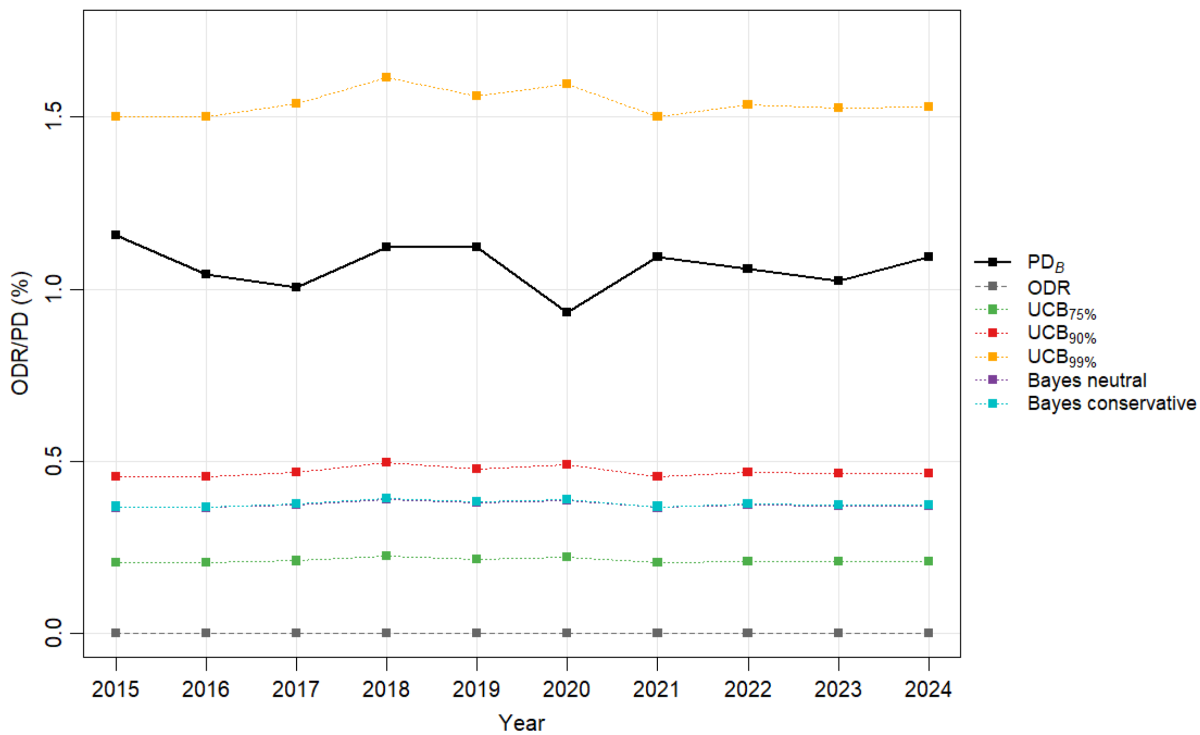


Figure 6: Annual results of Portfolio B, portfolio level, $\rho = 0.12$

For comparison, Figures 13 and 14 in Appendix C present results under a higher asset correlation, $\rho = 0.24$. The overall pattern is similar, but all benchmark estimates shift upwards since higher asset correlation implies stronger co-movement in defaults and therefore leads to higher estimates. In the annual results (see Figure 13 in Appendix C), $\text{UCB}_{75\%}$ remains below PD_B, while the Bayesian estimates and $\text{UCB}_{90\%}$ move close to PD_B and exceed it slightly, $\text{UCB}_{99\%}$ remains about five times higher. In the total history (see Figure 14 in Appendix C), PD_B is still significantly above the benchmark estimates across risk classes, and it is close to $\text{UCB}_{99\%}$ only in the lowest risk class. Overall, even under $\rho = 0.24$, the figures do not suggest a strong underestimation of PD_B at the portfolio level, and the total history results indicate that PD_B still remains conservative.

The additional yearly plots by risk class (Figures 15 and 16 in Appendix C) provide a more detailed view of the estimates at the risk class level for every year. Across both asset correlation settings estimates increase monotonically with the risk class due to applied *most prudent principle*. All curves are also fairly stable over time, since with no observed defaults estimates are mainly driven by portfolio size and the assumed correlation rather than realised default events. As expected, $\text{UCB}_{99\%}$ is the most conservative benchmark in all years and rises sharply in the highest risk classes. Finally, a higher ρ moves all benchmark estimates upward

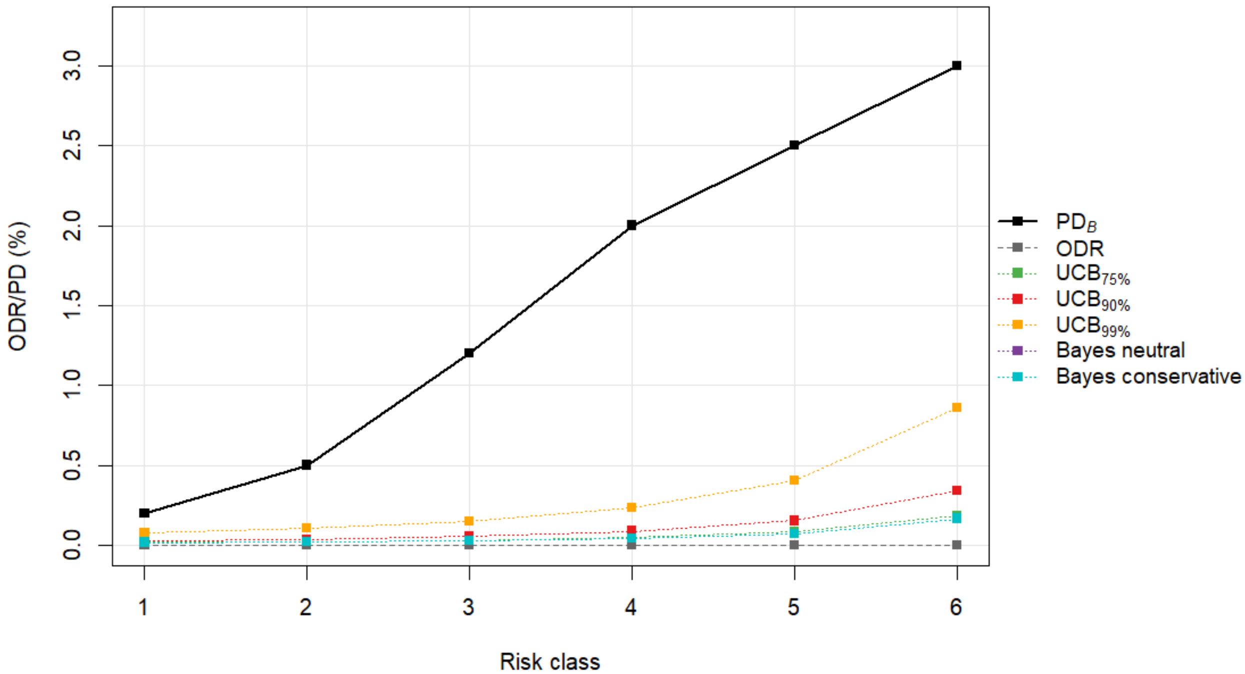


Figure 7: Total history results of Portfolio B, risk class level, $\rho = 0.12$

and brings them closer to the PD$_B$, while the overall pattern across risk classes remains the same.

Tables 5 and 6 present the quantified deviations between PD$_B$ and benchmark estimates, corresponding scores are also provided. As PD$_B$ is higher than UCB$_{75\%}$ and UCB$_{90\%}$ estimates in all years, the resulting scores are zero and remain in the green zone, which indicates no underestimation compared to these benchmark estimates. In contrast, the UCB$_{99\%}$ scores are in the yellow range or close to it, reflecting that PD$_B$ is moderately below this extremely conservative estimate. The Bayesian tail probabilities (see Table 6) are low (roughly 0.06 - 0.1 per year under both priors), indicating that almost whole probability mass of posterior density lies below PD$_B$. When total history results are considered (Tables 7 and 8) all scores are zero indicating that PD$_B$ is above benchmark estimates, indicating no underestimation of PD$_B$. Overall, under $\rho = 0.12$ the validation results for Portfolio B show no strong underestimation signal for PD$_B$ and indicate that PD$_B$ is conservative, especially when full history is assessed.

If higher asset correlation is assumed, i.e. $\rho = 0.24$ (Tables 13 and 14 in Appendix C), the conclusions are still similar. Since UCB$_{75\%}$ remains below PD$_B$, the corresponding scores are zero and stay in the green zone. As we can see, UCB$_{90\%}$ and the Bayesian estimates exceed PD$_B$ slightly, but the differences are not significant and the scores remain in the green zone.

Table 5: UCB estimates, quantified deviations and scores for Portfolio B at portfolio level, $\rho = 0.12$.

Year	PD_B (%)	$UCB_{75\%}$	R_{75}	Score(R_{75})	$UCB_{90\%}$	R_{90}	Score(R_{90})	$UCB_{99\%}$	R_{99}	Score(R_{99})
2015	1.155	0.204	-1.736	0.000	0.456	-0.930	0.000	1.501	0.262	0.714
2016	1.043	0.203	-1.634	0.000	0.456	-0.829	0.000	1.500	0.363	0.992
2017	1.004	0.209	-1.567	0.000	0.468	-0.764	0.000	1.538	0.427	1.166
2018	1.121	0.222	-1.619	0.000	0.495	-0.818	0.000	1.614	0.364	0.995
2019	1.120	0.213	-1.659	0.000	0.477	-0.854	0.000	1.563	0.333	0.909
2020	0.933	0.219	-1.447	0.000	0.489	-0.645	0.000	1.596	0.537	1.467
2021	1.094	0.204	-1.682	0.000	0.456	-0.876	0.000	1.501	0.316	0.863
2022	1.059	0.209	-1.624	0.000	0.467	-0.820	0.000	1.535	0.371	1.014
2023	1.025	0.207	-1.599	0.000	0.463	-0.794	0.000	1.526	0.398	1.087
2024	1.094	0.208	-1.660	0.000	0.465	-0.856	0.000	1.531	0.336	0.916

Table 6: Bayesian estimates, quantified deviations and scores for Portfolio B at portfolio level, $\rho = 0.12$.

Year	PD_B (%)	$Bayes^{neut}$	q^{neut}	Score(q^{neut})	$Bayes^{cons}$	q^{cons}	Score(q^{cons})
2015	1.155	0.364	0.0653	0.131	0.367	0.0664	0.133
2016	1.043	0.364	0.0783	0.157	0.367	0.0795	0.159
2017	1.004	0.372	0.0867	0.173	0.375	0.0879	0.176
2018	1.121	0.389	0.0773	0.155	0.392	0.0785	0.157
2019	1.120	0.377	0.0735	0.147	0.381	0.0746	0.149
2020	0.933	0.385	0.1033	0.207	0.388	0.1047	0.209
2021	1.094	0.364	0.0721	0.144	0.367	0.0732	0.146
2022	1.059	0.371	0.0789	0.158	0.374	0.0801	0.160
2023	1.025	0.369	0.0826	0.165	0.372	0.0838	0.168
2024	1.094	0.370	0.0741	0.148	0.373	0.0753	0.151

In contrast, $UCB_{99\%}$ is significantly higher than PD_B (roughly five times), which results in a red assessment score and indicates a sign of underestimation relative to this especially conservative benchmark. For the total history results (Tables 15 and 16 in Appendix C), no potential underestimation is observed and the assessment remains in the green zone as PD_B remains above all benchmark estimates.

In conlcusion, the validation results for Portfolio B do not indicate underestimation of PD_B at the portfolio level and across risk classes. Under both $\rho = 0.12$ and $\rho = 0.24$, PD_B appears at least adequately conservative. Concerns arise only relative to the particularly conservative $UCB_{99\%}$ benchmark and are mainly driven by the annual results, while the total history results do not indicate any sign of underestimation. However, if PD_B were intended to be conservative at an extreme level (i.e., close to a 99% upper bound), then under $\rho = 0.24$ the annual results would suggest that PD_B is too low relative to $UCB_{99\%}$ benchmark.

Table 7: Full history UCB estimates, quantified deviations and scores for Portfolio B at risk class level, $\rho = 0.12$.

Risk class	PD_B (%)	$UCB_{75\%}$	R_{75}	Score(R_{75})	$UCB_{90\%}$	R_{90}	Score(R_{90})	$UCB_{99\%}$	R_{99}	Score(R_{99})
1	0.200	0.018	-2.391	0.000	0.031	-1.874	0.000	0.080	-0.918	0.000
2	0.500	0.018	-3.308	0.000	0.038	-2.582	0.000	0.106	-1.549	0.000
3	1.200	0.030	-3.706	0.000	0.055	-3.079	0.000	0.148	-2.094	0.000
4	2.000	0.048	-3.721	0.000	0.088	-3.125	0.000	0.236	-2.136	0.000
5	2.500	0.083	-3.400	0.000	0.152	-2.800	0.000	0.404	-1.824	0.000
6	3.000	0.183	-2.796	0.000	0.341	-2.173	0.000	0.856	-1.254	0.000
TOTAL	1.066	0.018	-4.064	0.000	0.031	-3.547	0.000	0.080	-2.590	0.000

Table 8: Full history Bayesian estimates, quantified deviations and scores for Portfolio B at risk class level, $\rho = 0.12$.

Risk class	PD_B (%)	$\text{Bayes}^{\text{neut}}$	q^{neut}	Score(q^{neut})	$\text{Bayes}^{\text{cons}}$	q^{cons}	Score(q^{cons})
1	0.200	0.017	0.0012	0.002	0.017	0.0012	0.002
2	0.500	0.020	0.0001	0.000	0.020	0.0001	0.000
3	1.200	0.027	0.0000	0.000	0.027	0.0000	0.000
4	2.000	0.042	0.0000	0.000	0.042	0.0000	0.000
5	2.500	0.073	0.0000	0.000	0.073	0.0000	0.000
6	3.000	0.165	0.0001	0.000	0.166	0.0001	0.000
TOTAL	1.066	0.017	0.0000	0.000	0.017	0.0000	0.000

6 Discussion

The presented results show that the considered estimators yield strictly positive estimates in LDP settings, even in scenarios with zero observed defaults. The selected confidence level controls the conservatism of UCB estimates; the higher the confidence level, the more conservative the estimate. While for Bayesian estimates, conservatism is controlled through prior assumptions. All estimates are monotonic in risk classes, i.e., they increase with higher risk classes; this was enforced by applying the *most prudent principle* introduced by Pluto & Tasche [14].

Calculations were performed for two *asset correlation* scenarios, i.e. $\rho = \{0.12, 0.24\}$, results show that estimates are more conservative when ρ is higher, which is expected, since stronger dependence on a systematic factor, which is controlled through asset correlation ρ, implies a greater expectation of clustered defaults since it is assumed that all obligors are more dependent by common macroeconomic conditions. Observed results are also consistent with Pluto & Tasche [14], Tasche [18].

For both portfolios A and B, the resulted Bayesian estimates, defined as posterior means, are very similar, even though two different priors were considered: a conservative prior and a uniform prior. This similarity is expected because when only a few defaults are observed relative to the number of obligors, the data make high PD values very unlikely. Therefore, even though the conservative prior puts more weight on higher PDs, the observed data make large PD values unlikely. The resulting posterior distribution becomes very similar to the posterior distribution under the uniform prior, and so do their posterior means. To illustrate this effect, Figure 8 (in Appendix A) presents an example with illustrative values of the obligor count n and default count k, and with ρ fixed at 0.12. In panel (a), with $n = 99$ and $k = 3$, the two posterior density functions almost coincide, and the difference between the corresponding Bayesian estimators is negligible. In panel (b), with $n = 99$ and $k = 33$, the higher number of defaults shifts the posterior mass towards larger PD values, and the differences become noticeable. In particular, because the conservative prior places more weight on higher PDs, the resulting posterior is shifted further to the right and exhibits a heavier right tail than the posterior obtained under the uniform prior.

Results show that the Bayesian estimates under conservative and uniform priors tend to lie between the $\mathrm{UCB}_{75\%}$ and $\mathrm{UCB}_{90\%}$ estimates. This is consistent with the findings of Tasche [18], who reports that in more correlated scenarios, Bayesian estimates under uniform and conservative priors tend to fall between the $\mathrm{UCB}_{75\%}$ and $\mathrm{UCB}_{90\%}$ estimates, while in less correlated scenarios they may fall between lower confidence UCB estimates. As expected, the $\mathrm{UCB}_{99\%}$ is substantially higher than the other estimates, reflecting the very conservative nature of this confidence level.

Regarding the benchmarking results, the proposed comparison measures (quantified difference between benchmark estimate and bank's internal PD estimate that is under validation) are transformed into a score on the interval $[0, 3]$ and mapped to a traffic light color to indicate the significance of the potential underestimation of the PD estimate under validation relative to a selected benchmark estimate. Transformation to scores is based on a conservative principle: if the PD estimate under validation is higher than a benchmark estimate, the score is 0, which signals no underestimation relative to that estimate. Otherwise, when PD that is under validation sits lower than the benchmark estimate, the score increases as the deviation becomes larger. The larger the score, the larger the deviation. Of course, thresholds for score transformation can be adjusted depending on the validation objective and the desired sensitivity to underestimation. If greater conservatism is needed, thresholds could be lowered so that even minor deviations trigger a warning. In contrast, increasing thresholds would make the comparison less conservative, so only larger deviations would trigger a warning. It is essential to note that the raw comparison measure transformed to a score is specific to a benchmark estimate and should be interpreted relative to the chosen estimate. Overall, transforming raw comparison measures into scores and mapping them to traffic light colors helps interpret results more clearly and compare them across years, portfolios, and risk classes.

Several study limitations should be acknowledged:

1. The study is based on simulated data, as real-world data could not be used due to sensitivity. On the other hand, the data is not complex to generate, since it consists only of the number of obligors, defaults, and internal PD estimates for each risk grade, and it still resembles the real-life LDP.

2. Both UCB and Bayesian estimators depend on assumptions regarding *asset correlation* ρ and, in a multi-period scenario, on *time correlation* θ. The selected values affect the magnitude of the estimates. If *asset correlation* and *time correlation* values are not calibrated portfolio-specifically, the results should be interpreted as scenario-based rather than portfolio estimates.

3. Bayesian estimates depend on the chosen prior. In this study, neutral and conservative priors were considered, but alternative priors may yield different posterior estimates. If the prior is expert-based or portfolio-specific historical evidence, it could better reflect a portfolio's characteristics, and PD estimates might be more accurate.

4. The proposed validation measures with selected thresholds and score transformation introduce an additional layer of judgment. The proposed thresholds can be adjusted; therefore, resulting scores and conclusions would also change.

Future work could extend the analysis by considering alternative priors in the Bayesian approach, including expert-based priors. Additionally, other measures to quantify deviations between benchmark estimates and PD estimates under validation could be explored and proposed.

7 Conclusion

In this thesis, two PD estimators, designed explicitly for LDPs, were reviewed in one- and multi-period settings under the assumption of correlated defaults. The considered estimators are the UCB estimator presented by Pluto & Tasche [14] and the Bayesian posterior-based estimation presented by Tasche [18]. Results show that estimates from the latter estimators are strictly positive and well-defined in cases of no or scarce defaults. These estimates are used to assess the conservatism of banks' internal PD estimates by quantifying deviations and converting them into an interpretable $[0, 3]$ score that reflects the severity of potential underestimation.

The main contribution of this thesis is the application of LDP PD estimators to the validation task of banks' internal PD estimates. In such circumstances, traditional backtesting of the internal PD estimate based on observed default rates is unreliable because defaults are rare. Therefore, the thesis adopts a benchmarking approach to validation, consistent with the

Basel guidance for LDPs [6]. In the academic literature, there is limited evidence on using LDP-specific estimators as benchmarking tools for PD validation, and, to the best of my knowledge, no prior work proposes a simple framework comparable to the one developed in this thesis to support interpretation and easier comparisons. Overall, the proposed approach offers a practical way to assess the potential underestimation and the degree of conservatism of a bank's internal PD estimate.

References

[1] Scott Aguais. Designing and implementing a Basel II compliant PIT-TTC ratings framework, 2008.

[2] Basel Committee on Banking Supervision. The Internal Ratings-Based Approach: supporting document to the new Basel capital accord, January 2001. Issued for comment by 31 May 2001.

[3] Basel Committee on Banking Supervision. Basel II: international convergence of capital measurement and capital standards: a revised framework, 2004.

[4] Basel Committee on Banking Supervision. An explanatory note on the Basel II IRB risk weight functions, 2005.

[5] Basel Committee on Banking Supervision. Studies on the validation of internal rating systems (revised), 2005.

[6] Basel Committee on Banking Supervision. Validation of low-default portfolios in the Basel II framework, September 2005. Basel Committee Newsletter No. 6.

[7] Basel Committee on Banking Supervision. Basel III: a global regulatory framework for more resilient banks and banking systems, 2011.

[8] Douglas W. Dwyer. The distribution of defaults and bayesian model validation. *Journal of Risk Model Validation*, 2007.

[9] Andrius Grigutis. Probabilistic overview of probabilities of default for low default portfolios by K. Pluto and D. Tasche. *arXiv preprint*, 2023. arXiv:2303.06148.

[10] Geoffrey R. Grimmett and David R. Stirzaker. *Probability and random processes*. Oxford University Press, 3 edition, 2001.

[11] Nicholas M. Kiefer. Default estimation for low-default portfolios. *Journal of Empirical Finance*, 16(1):164–173, 2009.

[12] Machiel Kruger. Validation and monitoring of PD models for low default portfolios using PROC MCMC. In *Proceedings of SAS Global Forum*, 2015. Paper 1400-2015.

[13] Mark Pinsky and Samuel Karlin. *An introduction to stochastic modeling*. Academic Press, 2010.

[14] Katja Pluto and Dirk Tasche. Estimating probabilities of default for low default portfolios, 2005.

[15] Katja Pluto and Dirk Tasche. Estimating probabilities of default for low default portfolios. In *The Basel II Risk Parameters*, pages 75–101. Springer, 2011.

[16] Christian P. Robert and George Casella. *Monte Carlo statistical methods*. Springer, 2 edition, 2004.

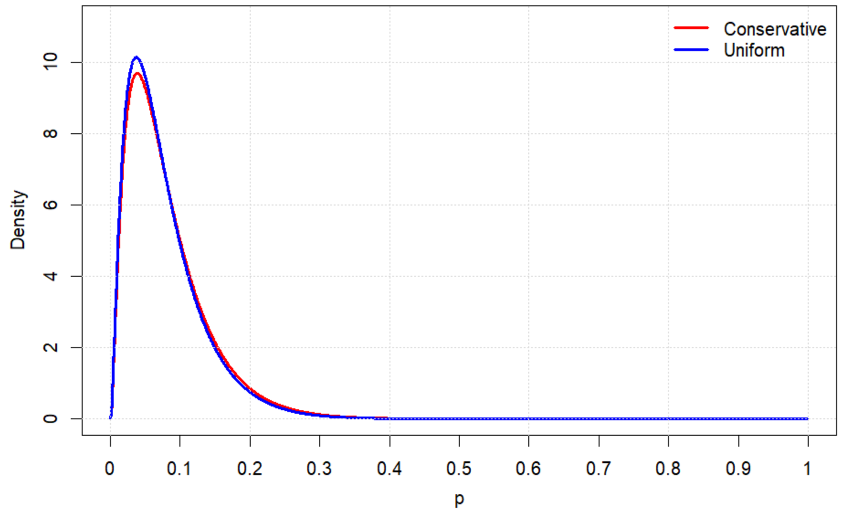
[17] Sheldon M. Ross. *A first course in probability*. Pearson, 9 edition, 2014.

[18] Dirk Tasche. Bayesian estimation of probabilities of default for low default portfolios, 2013.

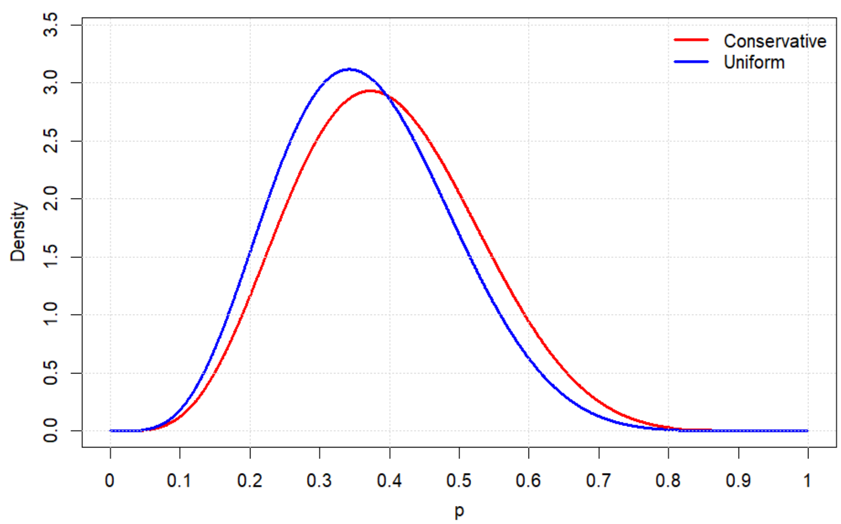
[19] Oldrich A. Vasicek. Probability of loss on loan portfolio. KMV Corporation, 1987.

[20] Jing Zhang, Fanlin Zhu, and Joseph Lee. Asset correlation, realized default correlation, and portfolio credit risk, 2008.

A Posterior Probability Density Function for low and high ODR



(a) Posterior PDF when $n = 99, k = 3$



(b) Posterior PDF when $n = 99, k = 33$

Figure 8: Posterior Probability Density Function for low and high ODR

B Portfolio A

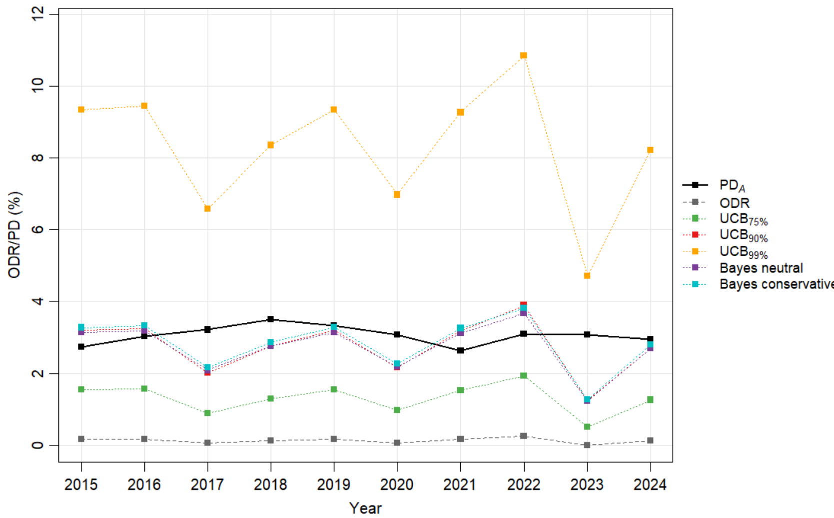


Figure 9: Annual results of Portfolio A, portfolio level, $\rho = 0.24$

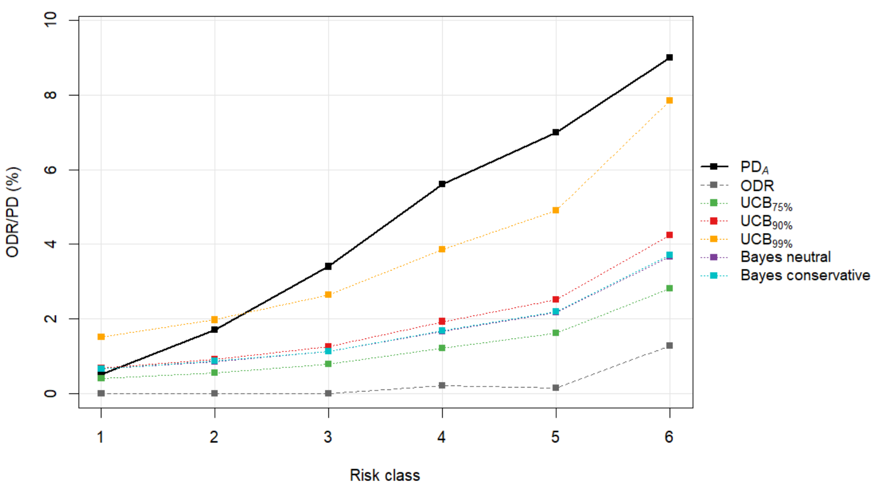


Figure 10: Total history results of Portfolio A, risk class level, $\rho = 0.24$

Table 9: UCB estimates, quantified deviations and scores for Portfolio A at portfolio level, $\rho = 0.24$.

Year	PD_A (%)	$UCB_{75\%}$	R_{75}	Score(R_{75})	$UCB_{90\%}$	R_{90}	Score(R_{90})	$UCB_{99\%}$	R_{99}	Score(R_{99})
2015	2.726	1.536	-0.574	0.000	3.205	0.162	0.442	9.340	1.231	3.000
2016	3.041	1.561	-0.667	0.000	3.253	0.067	0.184	9.453	1.134	3.000
2017	3.213	0.885	-1.290	0.000	2.009	-0.470	0.000	6.582	0.717	1.958
2018	3.507	1.284	-1.005	0.000	2.758	-0.240	0.000	8.356	0.868	2.371
2019	3.331	1.537	-0.774	0.000	3.207	-0.038	0.000	9.345	1.032	2.817
2020	3.068	0.962	-1.160	0.000	2.152	-0.354	0.000	6.975	0.821	2.243
2021	2.620	1.520	-0.545	0.000	3.175	0.192	0.525	9.271	1.264	3.000
2022	3.097	1.918	-0.479	0.000	3.892	0.229	0.624	10.845	1.253	3.000
2023	3.076	0.500	-1.817	0.000	1.252	-0.899	0.000	4.718	0.428	1.168
2024	2.939	1.257	-0.850	0.000	2.695	-0.087	0.000	8.210	1.027	2.805

Table 10: Bayesian estimates, quantified deviations and scores for Portfolio A at portfolio level, $\rho = 0.24$.

Year	PD_A (%)	$Bayes^{neut}$	q^{neut}	Score(q^{neut})	$Bayes^{cons}$	q^{cons}	Score(q^{cons})
2015	2.726	3.145	0.3954	0.791	3.276	0.4079	0.816
2016	3.041	3.178	0.3600	0.720	3.326	0.3727	0.745
2017	3.213	2.087	0.1985	0.397	2.160	0.2070	0.414
2018	3.507	2.760	0.2573	0.515	2.870	0.2685	0.537
2019	3.331	3.146	0.3232	0.646	3.278	0.3359	0.672
2020	3.068	2.191	0.2252	0.450	2.270	0.2344	0.469
2021	2.620	3.125	0.4072	0.814	3.255	0.4195	0.839
2022	3.097	3.670	0.4143	0.829	3.817	0.4285	0.857
2023	3.076	1.235	0.1028	0.206	1.273	0.1075	0.215
2024	2.939	2.700	0.3095	0.619	2.804	0.3207	0.641

Table 11: Full history UCB estimates, quantified deviations and scores for Portfolio A at risk class level, $\rho = 0.24$.

Risk class	PD_A (%)	$UCB_{75\%}$	R_{75}	Score(R_{75})	$UCB_{90\%}$	R_{90}	Score(R_{90})	$UCB_{99\%}$	R_{99}	Score(R_{99})
1	0.500	0.405	-0.211	0.000	0.678	0.305	0.832	1.506	1.103	3.000
2	1.700	0.552	-1.124	0.000	0.910	-0.625	0.000	1.970	0.148	0.403
3	3.400	0.778	-1.474	0.000	1.258	-0.994	0.000	2.639	-0.253	0.000
4	5.600	1.222	-1.522	0.000	1.927	-1.067	0.000	3.850	-0.375	0.000
5	7.000	1.616	-1.466	0.000	2.508	-1.027	0.000	4.897	-0.357	0.000
6	9.000	2.819	-1.161	0.000	4.238	-0.753	0.000	7.852	-0.136	0.000
TOTAL	3.059	0.405	-2.022	0.000	0.678	-1.507	0.000	1.506	-0.708	0.000

Table 12: Full history Bayesian estimates, quantified deviations and scores for Portfolio A at risk class level, $\rho = 0.24$.

Risk class	PD_A (%)	$Bayes^{neut}$	q^{neut}	Score(q^{neut})	$Bayes^{cons}$	q^{cons}	Score(q^{cons})
1	0.500	0.660	0.5548	1.219	0.661	0.5563	1.225
2	1.700	0.861	0.0583	0.117	0.863	0.0591	0.118
3	3.400	1.132	0.0061	0.012	1.136	0.0063	0.013
4	5.600	1.669	0.0020	0.004	1.679	0.0021	0.004
5	7.000	2.171	0.0020	0.004	2.186	0.0021	0.004
6	9.000	3.659	0.0145	0.029	3.699	0.0155	0.031
TOTAL	3.059	0.660	0.0003	0.001	0.661	0.0003	0.001

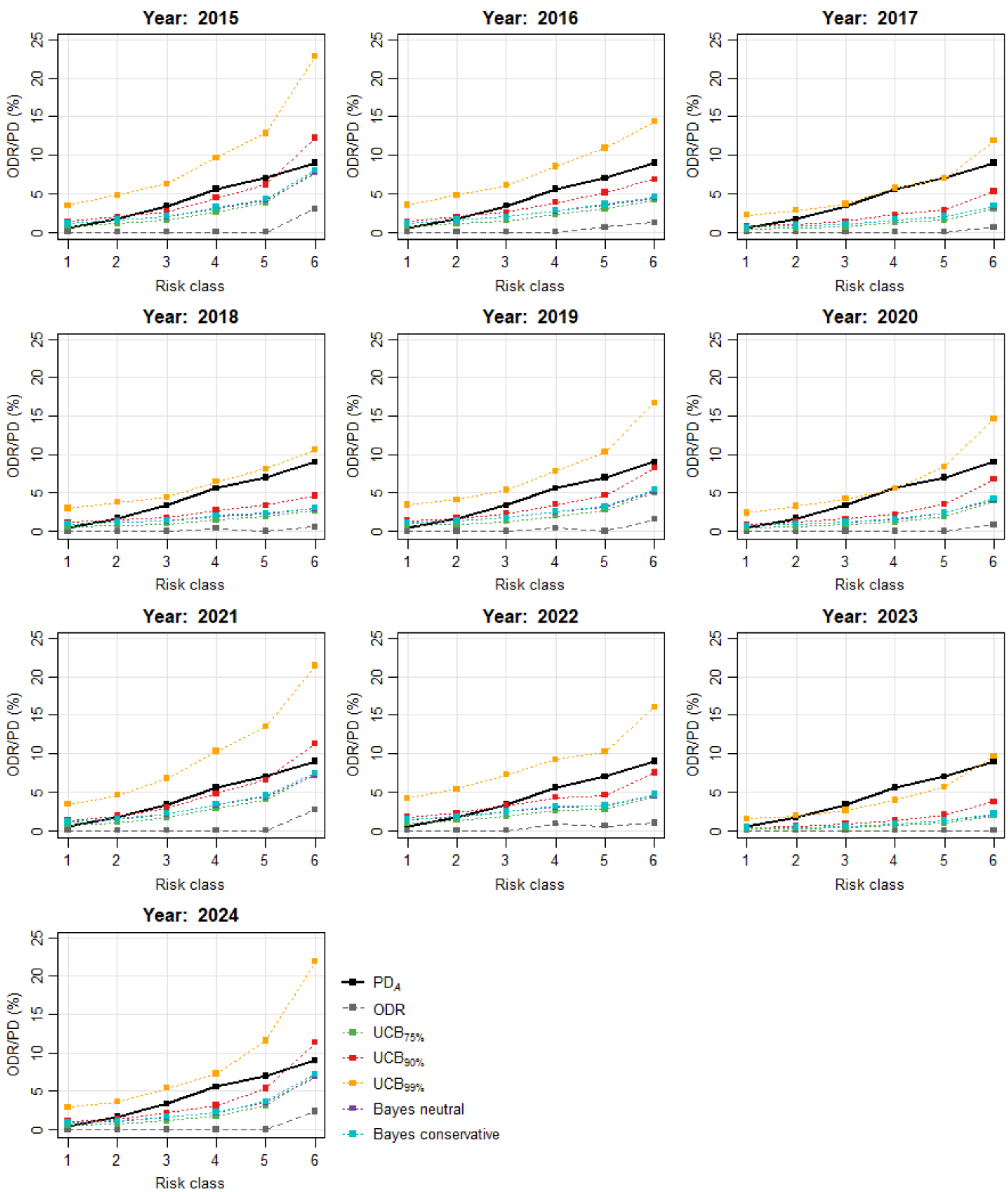


Figure 11: Annual results of Portfolio A, risk class level, $\rho = 0.12$

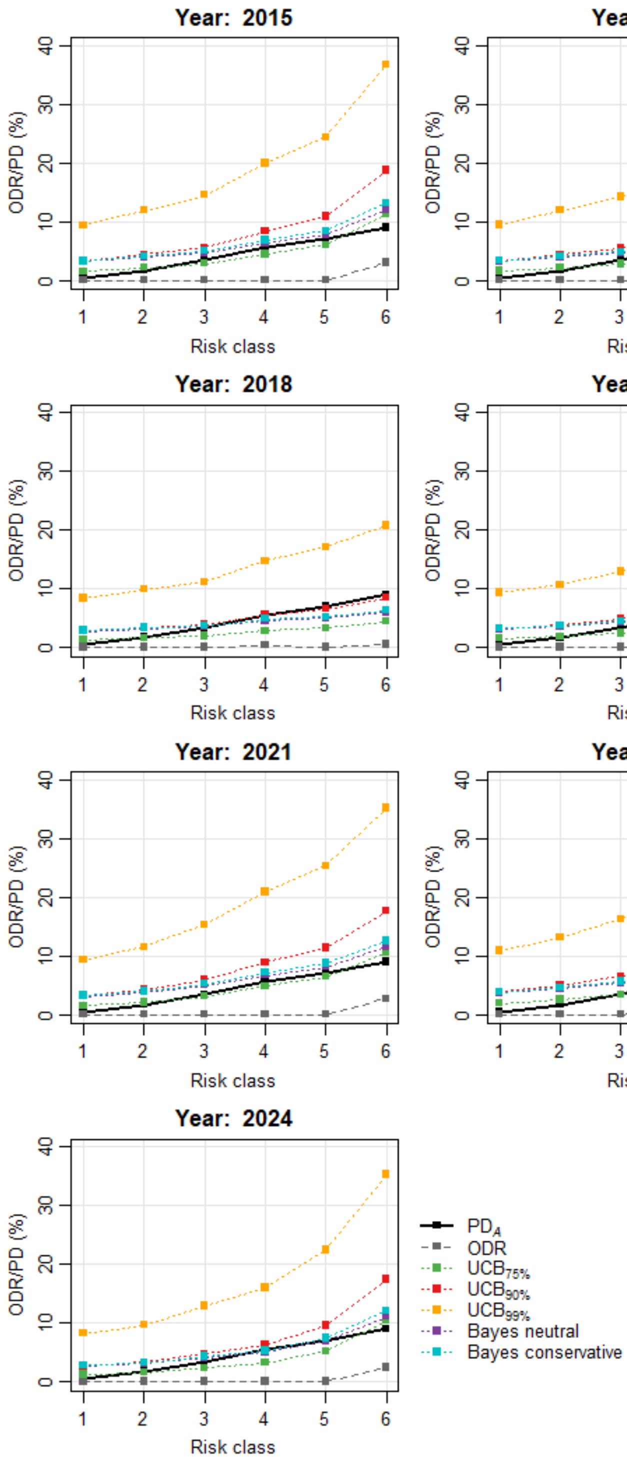


Figure 12: Annual results of Portfolio A, risk class level, $\rho = 0.24$

C Portfolio B

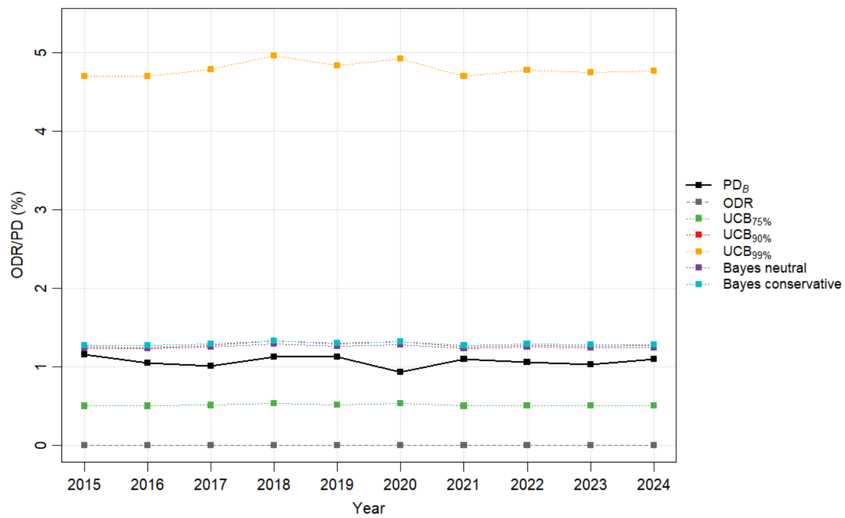


Figure 13: Annual results of Portfolio B, portfolio level, $\rho = 0.24$

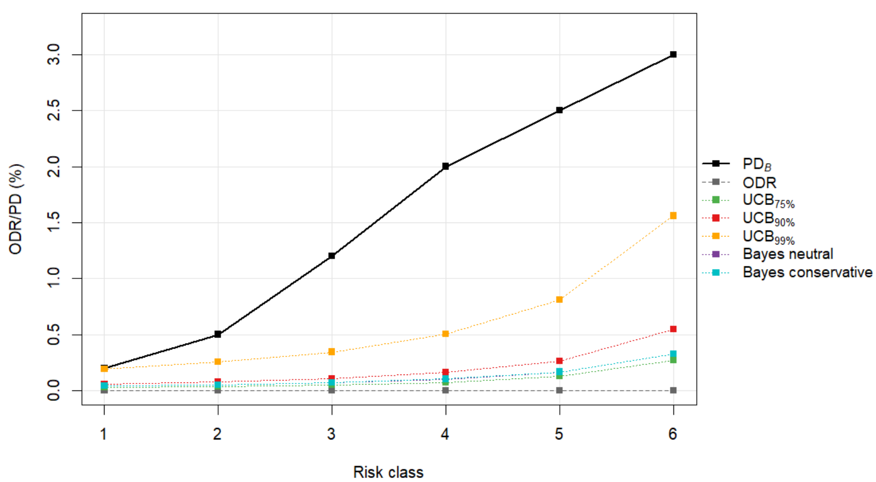


Figure 14: Total history results of Portfolio B, risk class level, $\rho = 0.24$

Table 13: UCB estimates, quantified deviations and scores for Portfolio B at portfolio level, $\rho = 0.24$.

Year	PD_B (%)	$UCB_{75\%}$	R_{75}	Score(R_{75})	$UCB_{90\%}$	R_{90}	Score(R_{90})	$UCB_{99\%}$	R_{99}	Score(R_{99})
2015	1.155	0.497	-0.843	0.000	1.245	0.074	0.203	4.694	1.402	3.000
2016	1.043	0.497	-0.742	0.000	1.244	0.176	0.481	4.693	1.503	3.000
2017	1.004	0.507	-0.684	0.000	1.272	0.236	0.646	4.780	1.566	3.000
2018	1.121	0.535	-0.740	0.000	1.342	0.212	0.458	4.975	1.492	3.000
2019	1.120	0.514	-0.780	0.000	1.293	0.152	0.384	4.842	1.469	3.000
2020	0.933	0.530	-0.565	0.000	1.332	0.346	0.938	4.957	1.668	3.000
2021	1.094	0.497	-0.789	0.000	1.245	0.144	0.352	4.695	1.454	3.000
2022	1.059	0.506	-0.739	0.000	1.268	0.179	0.494	4.762	1.505	3.000
2023	1.025	0.503	-0.712	0.000	1.261	0.206	0.567	4.744	1.535	3.000
2024	1.094	0.504	-0.774	0.000	1.265	0.145	0.397	4.761	1.470	3.000

Table 14: Bayesian estimates, quantified deviations and scores for Portfolio B at portfolio level, $\rho = 0.24$.

Year	PD_B (%)	$Bayes^{neut}$	q_t^{neut}	Score(q_t^{neut})	$Bayes^{cons}$	q_t^{cons}	Score(q_t^{cons})
2015	1.155	1.230	0.3142	0.628	1.267	0.3203	0.641
2016	1.043	1.230	0.3410	0.682	1.267	0.3470	0.694
2017	1.004	1.248	0.3558	0.712	1.286	0.3618	0.724
2018	1.121	1.286	0.3353	0.671	1.326	0.3416	0.683
2019	1.120	1.260	0.3294	0.659	1.299	0.3356	0.671
2020	0.933	1.278	0.3828	0.766	1.318	0.3889	0.778
2021	1.094	1.230	0.3285	0.657	1.267	0.3345	0.669
2022	1.059	1.246	0.3411	0.682	1.285	0.3471	0.694
2023	1.025	1.241	0.3485	0.697	1.279	0.3546	0.709
2024	1.094	1.244	0.3318	0.664	1.282	0.3379	0.676

Table 15: Full history UCB estimates, quantified deviations and scores for Portfolio B at risk class level, $\rho = 0.24$.

Risk class	PD_B (%)	$UCB_{75\%}$	R_{75}	Score(R_{75})	$UCB_{90\%}$	R_{90}	Score(R_{90})	$UCB_{99\%}$	R_{99}	Score(R_{99})
1	0.200	0.027	-1.988	0.000	0.058	-1.234	0.000	0.193	-0.034	0.000
2	0.500	0.037	-2.615	0.000	0.078	-1.858	0.000	0.256	-0.669	0.000
3	1.200	0.046	-3.261	0.000	0.108	-2.407	0.000	0.344	-1.250	0.000
4	2.000	0.073	-3.308	0.000	0.163	-2.508	0.000	0.507	-1.373	0.000
5	2.500	0.128	-2.975	0.000	0.265	-2.246	0.000	0.810	-1.126	0.000
6	3.000	0.268	-2.415	0.000	0.547	-1.702	0.000	1.560	-0.654	0.000
TOTAL	1.066	0.027	-3.661	0.000	0.058	-2.907	0.000	0.193	-1.706	0.000

Table 16: Full history Bayesian estimates, quantified deviations and scores for Portfolio B at risk class level, $\rho = 0.24$.

Risk class	PD_B (%)	$Bayes^{neut}$	q^{neut}	Score(q^{neut})	$Bayes^{cons}$	q^{cons}	Score(q^{cons})
1	0.200	0.041	0.0383	0.077	0.041	0.0385	0.077
2	0.500	0.051	0.0080	0.016	0.051	0.0081	0.016
3	1.200	0.069	0.0011	0.002	0.069	0.0011	0.002
4	2.000	0.102	0.0005	0.001	0.102	0.0005	0.001
5	2.500	0.165	0.0009	0.002	0.166	0.0009	0.002
6	3.000	0.324	0.0037	0.007	0.327	0.0038	0.008
TOTAL	1.066	0.041	0.0002	0.000	0.041	0.0002	0.000

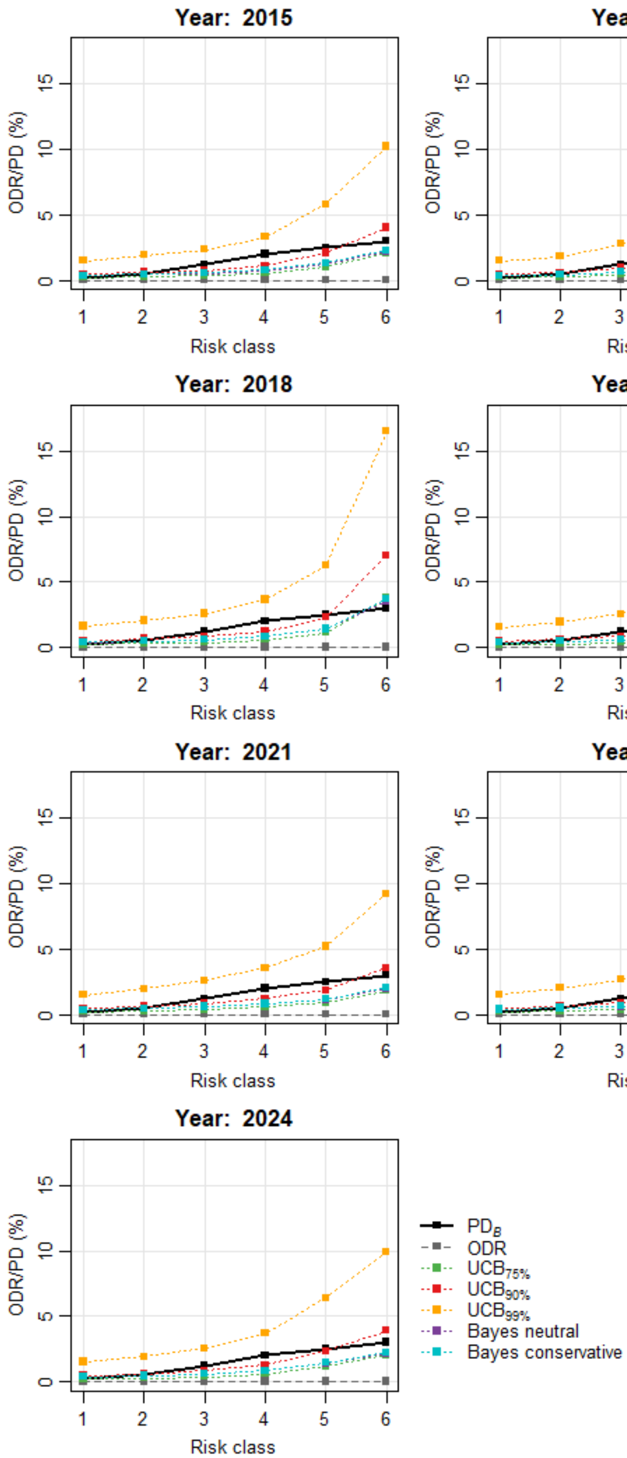


Figure 15: Annual results of Portfolio B, risk class level, $\rho = 0.12$

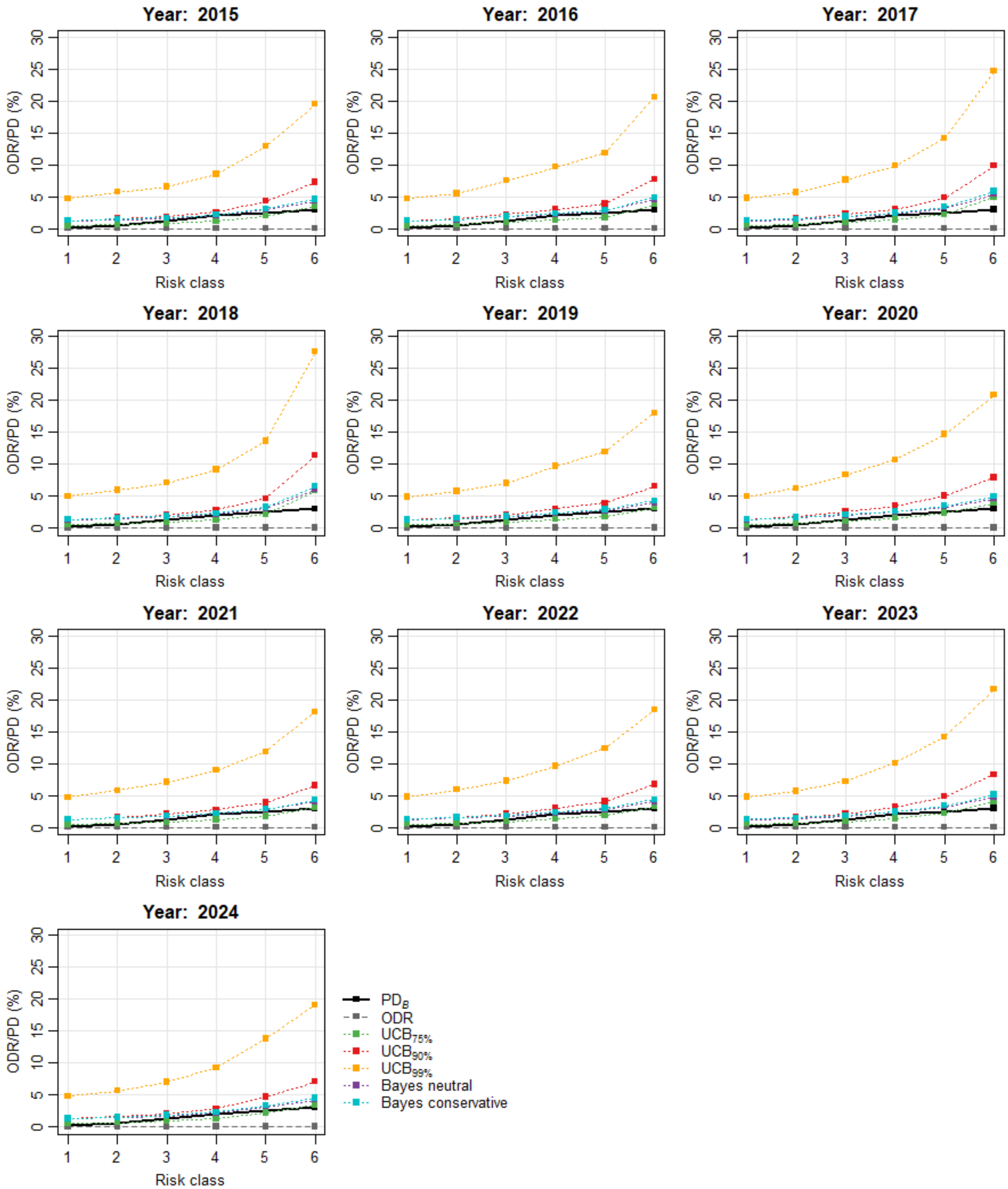


Figure 16: Annual results of Portfolio B, risk class level, $\rho = 0.24$

D AI usage

D.1 Generating formulas in LaTeX format

Almost all formulas in **3 Methodology** section were generated with ChatGPT. Snippets of formulas were uploaded, and it was asked to rewrite them in LaTeX style. The generated formulas were then reviewed and fixed where needed.

D.2 Generating tables in LaTeX format

All tables in this thesis were generated with ChatGPT. The results were provided, and it was requested that a table be generated with the corresponding columns and values. The design of the generated tables was then adjusted, column names were renamed, and the background colors for cells in the scores columns were added. Also, it was checked that the correct values were placed in the cells and adjusted where needed.

E R code

The code is split into two scripts: the main script and the custom functions script. The custom functions script contains functions of PD estimators, functions for various plots, etc.; this script is sourced in the main script.

E.1 Main script

```
rm(list=ls())
library(stats)
library(data.table)
library(cubature)
library(dplyr)

options(scipen = 999)

# Source functions
source("Functions.R")

# Data
data <- rbind(simulate_ldp_data(portfolio_name = "Portfolio A",
                                grade_ttc_pd = c(0.005, 0.017, 0.034, 0.056, 0.07, 0.09),
                                grade_pd = c(0, 0, 0, 0.001, 0.001, 0.01),
                                seed = 1001),
              simulate_ldp_data(portfolio_name = "Portfolio B",
                                grade_ttc_pd = c(0.002, 0.005, 0.012, 0.02, 0.025, 0.03),
                                grade_pd = c(0, 0, 0, 0, 0, 0),
                                seed = 2002))

data.annual <- copy(data) %>%
  .[, ODF := round(Default_count/Obligor_count*100, digits = 2)] %>%
  .[Grade != "TOTAL", Grade_num := as.integer(as.character(Grade))] %>%
  .[order(Portfolio, Year, Grade_num)] %>%
  .[!is.na(Grade_num),
    `:=`(n = rev(cumsum(rev(Obligor_count))),
         k = rev(cumsum(rev(Default_count)))),
    by = .(Portfolio, Year)] %>%
```

```
  .[,
    ‘:=‘(n_total = sum(Obligor_count[!is.na(Grade_num)]),
         k_total = sum(Default_count[!is.na(Grade_num)])),
    by = .(Portfolio, Year)] %>%
  .[Grade == "TOTAL", ‘:=‘(n = n_total, k = k_total)] %>%
  .[, c("n_total", "k_total", "Grade_num") := NULL]

# Save portfolio results
# write.csv(data.annual,"Results/data_annual.csv", row.names = FALSE)

# Summary for multiperiod (will be used later)
data.mult.per.odf <- copy(data.annual) %>%
  .[, .(ODF = round(sum(Default_count)/sum(Obligor_count)*100, digits = 2)),
  by = .(Portfolio, Grade)]

# --- Portfolio composition ---
# Plot portfolio composition
# Portfolio A
plot_portfolio_composition(portfolio = data.annual[Portfolio == "Portfolio A" & Grade != "TOTAL"])
# Portfolio B
plot_portfolio_composition(portfolio = data.annual[Portfolio == "Portfolio B" & Grade != "TOTAL"])

# --- PD estimation ---
# Annual PD estimation
tmp.data.copy <- copy(data.annual)
final.result.table.annual <- data.table()

for (rho in c(0.12, 0.24)) {
  for (portfolio in unique(data.annual$Portfolio)) {
    for (year in unique(data.annual[Portfolio == portfolio]$Year)) {
      for (grade in unique(data.annual[Portfolio == portfolio & Year == year]$Grade)) {
        start_time <- Sys.time()

        tmp.data.row <- tmp.data.copy[Portfolio == portfolio & Year == year & Grade == grade]
        tmp.data.row.upd <- tmp.data.row[, ‘:=‘ (rho = rho,
                                                 ‘75% PT upper bound‘ = round(PT_upper_bound(n = tmp.data.row$n,
                                                                                             k = tmp.data.row$k,
                                                                                             rho = rho,
                                                                                             alpha = 0.75)*100, 4),
                                                 ‘90% PT upper bound‘ = round(PT_upper_bound(n = tmp.data.row$n,
                                                                                             k = tmp.data.row$k,
                                                                                             rho = rho,
                                                                                             alpha = 0.9)*100, 4),
                                                 ‘99% PT upper bound‘ = round(PT_upper_bound(n = tmp.data.row$n,
                                                                                             k = tmp.data.row$k,
                                                                                             rho = rho,
                                                                                             alpha = 0.99)*100, 4),
                                                 ‘Neutral Bayesian on (0, 1)‘ = round(bayes_mean_neutral(n = tmp.data.row$n,
                                                                                                         k = tmp.data.row$k,
                                                                                                         rho = rho,
                                                                                                         u = 1)*100, 4),
                                                 ‘Conservative Bayesian‘ = round(bayes_mean_conservative(n = tmp.data.row$n,
                                                                                                         k = tmp.data.row$k,
                                                                                                         rho = rho)*100, 4),
                                                 ‘Tail probability conservative‘ = calculate_tail_prob(n = n,
                                                                                                       k = k,
                                                                                                       rho = rho,
                                                                                                       p_ttc = TTC_PD/100,
                                                                                                       prior = "cons"),
                                                 ‘Tail probability uniform‘ = calculate_tail_prob(n = n,
                                                                                                  k = k,
                                                                                                  rho = rho,
                                                                                                  p_ttc = TTC_PD/100,
                                                                                                  prior = "unif"))] %>%
          .[, ‘:=‘ (R75 = log(‘75% PT upper bound‘/TTC_PD),
                    R90 = log(‘90% PT upper bound‘/TTC_PD),
                    R99 = log(‘99% PT upper bound‘/TTC_PD))]

        final.result.table.annual <- rbind(final.result.table.annual, tmp.data.row.upd)

        end_time <- Sys.time()
        duration <- end_time - start_time
        cat(portfolio, "|", "Year:", year, "|", "Grade:", grade, "|", "rho:", rho, "|", "Duration:", round(as.numeric(duration, units = "mins"), 2), "minutes\n")
      }
    }
  }
}
```

```
# Save annual results to avoid running a code
# saveRDS(final.result.table.annual, "Results/annual_res.RDS")

# Upload annual results
final.result.table.annual <- readRDS("Results/annual_res.RDS")

# Portfolio level annual results
res.portf.lvl <- copy(final.result.table.annual) %>%
  .[, .(Portfolio, Year, Grade, TTC_PD, rho,
      `75% PT upper bound`, R75,
      `90% PT upper bound`, R90,
      `99% PT upper bound`, R99,
      `Neutral Bayesian on (0, 1)`, `Tail probability uniform`,
      `Conservative Bayesian`, `Tail probability conservative`)] %>%
  .[, `:=` (Score_R75 = score_R(R75),
            Score_R90  = score_R(R90),
            Score_R99 = score_R(R99),
            Score_q_neut = score_q(`Tail probability uniform`),
            Score_q_cons = score_q(`Tail probability conservative`))] %>%
  .[, Score_UCB_max := pmax(Score_R75, Score_R90, Score_R99)] %>%
  .[, Score_Bayes_max := pmax(Score_q_neut, Score_q_cons)]
# Save portfolio level annual results
# write.csv(res.portf.lvl,"Results/res_portf_lvl.csv", row.names = FALSE)

#-------------------------------------------------------------------------------
# Whole history PD estimation
# Select theta value
theta <- 0.6

# For testing purposes
# tmp.data.row <- data.table(Year = 2003:2010,
#                            n = rep(125, 8),
#                            k = c(rep(0, 7), 1))

tmp.data.copy <- copy(data.annual)
final.result.table.mult.period <- data.table()
lst.mult.per.pdf <- list()

for (rho in c(0.12, 0.24)) {
  for (portfolio in unique(data.annual$Portfolio)) {
    for (grade in unique(data.annual$Grade)) {
      start_time <- Sys.time()

      tmp.data.row <- tmp.data.copy[Portfolio == portfolio & Grade == grade] %>%
        .[, TTC_PD := sum(Obligor_count*TTC_PD)/sum(Obligor_count)]

      tmp.data.bay.unif <- Multiple_period_estimator(n = tmp.data.row$n,
                                                     k = tmp.data.row$k,
                                                     rho = rho,
                                                     theta = theta,
                                                     type = "Uniform",
                                                     p_ttc = unique(tmp.data.row$TTC_PD)/100,
                                                     return_posterior = TRUE)

      tmp.data.bay.cons <- Multiple_period_estimator(n = tmp.data.row$n,
                                                     k = tmp.data.row$k,
                                                     rho = rho,
                                                     theta = theta,
                                                     type = "Conservative",
                                                     p_ttc = unique(tmp.data.row$TTC_PD)/100,
                                                     return_posterior = TRUE)

      tmp.data.row.upd <- copy(tmp.data.row)[, `:=` (rho = rho,
                                                     theta = theta,
                                                     `Multiple period: Neutral Bayesian on (0, 1)` = round(tmp.data.bay.unif$lambda_hat*100, 4),
                                                     `Tail probability uniform` = tmp.data.bay.unif$posterior$tail_prob,
                                                     `Multiple period: Conservative Bayesian` = round(tmp.data.bay.cons$lambda_hat*100, 4),
                                                     `Tail probability conservative` = tmp.data.bay.cons$posterior$tail_prob,
                                                     `Multiple period: 75% upper bound` = round(Multiple_period_estimator(n = tmp.data.row$n,
                                                                                                                        k = tmp.data.row$k,
                                                                                                                        rho = rho,
                                                                                                                        theta = theta,
                                                                                                                        type = "UCB",
                                                                                                                        alpha = 0.75)*100, 4),
                                                     `Multiple period: 90% upper bound` = round(Multiple_period_estimator(n = tmp.data.row$n,
                                                                                                                        k = tmp.data.row$k,
```

```
                                                                                                                                   rho = rho,
                                                                                                                                   theta = theta,
                                                                                                                                   type = "UCB",
                                                                                                                                   alpha = 0.9)*100, 4),
                                                                  `Multiple period: 99% upper bound` = round(Multiple_period_estimator(n = tmp.data.row$n,
                                                                                                                                   k = tmp.data.row$k,
                                                                                                                                   rho = rho,
                                                                                                                                   theta = theta,
                                                                                                                                   type = "UCB",
                                                                                                                                   alpha = 0.99)*100, 4))]

      # Save posterior pdf
      lst.mult.per.pdf <- append(lst.mult.per.pdf,
                                 setNames(list(list(Portfolio = portfolio,
                                                    Grade = grade,
                                                    post_grid_unif = tmp.data.bay.unif$posterior$grid,
                                                    post_pdf_unif = tmp.data.bay.unif$posterior$pdf,
                                                    post_grid_cons = tmp.data.bay.cons$posterior$grid,
                                                    post_pdf_cons = tmp.data.bay.cons$posterior$pdf)),
                                          paste0(portfolio, ", grade:", grade)))

      tmp.data.row.upd <- tmp.data.row.upd[, Years := as.character(paste0(min(tmp.data.row$Year), "-", max(tmp.data.row$Year)))] %>%
        .[Grade == "TOTAL", TTC_PD := sum(Obligor_count * TTC_PD)/sum(Obligor_count)] %>%
        .[, c("Obligor_count", "Default_count", "n", "k", "ODF", "Year") := NULL] %>%
        unique()

      final.result.table.mult.period <- rbind(final.result.table.mult.period, tmp.data.row.upd)

      rm(tmp.data.bay.unif, tmp.data.bay.cons)

      end_time <- Sys.time()
      duration <- end_time - start_time
      cat(portfolio, "|", "Grade:", grade, "|", round(as.numeric(duration, units = "mins"), 2), "minutes\n")
    }
  }
}

# Save multiple period results
# saveRDS(final.result.table.mult.period, "Results/mult_per_res.RDS")
# saveRDS(lst.mult.per.pdf, "Results/lst_mult_per_pdf.RDS")

# Upload multiple period results
final.result.table.mult.period <- readRDS("Results/mult_per_res.RDS")

# Add ODF counts
final.result.table.mult.period <- merge(final.result.table.mult.period,
                                        data.mult.per.odf,
                                        by = c("Portfolio", "Grade"),
                                        all.x = TRUE)

# Whole history results
res.portf.lvl.mult <- copy(final.result.table.mult.period) %>%
  .[, .(Portfolio, Grade, TTC_PD, rho, theta,
        `Multiple period: 75% upper bound`,
        `Multiple period: 90% upper bound`,
        `Multiple period: 99% upper bound`,
        `Multiple period: Neutral Bayesian on (0, 1)`, `Tail probability uniform`,
        `Multiple period: Conservative Bayesian`, `Tail probability conservative`)] %>%
  .[, `:=` (R75 = log(`Multiple period: 75% upper bound`/TTC_PD),
            R90 = log(`Multiple period: 90% upper bound`/TTC_PD),
            R99 = log(`Multiple period: 99% upper bound`/TTC_PD))] %>%
  .[, `:=` (Score_R75 = score_R(R75),
            Score_R90  = score_R(R90),
            Score_R99 = score_R(R99),
            Score_q_neut = score_q(`Tail probability uniform`),
            Score_q_cons = score_q(`Tail probability conservative`))] %>%
  .[, Score_UCB_max := pmax(Score_R75, Score_R90, Score_R99)] %>%
  .[, Score_Bayes_max := pmax(Score_q_neut, Score_q_cons)]

# Save pooled results
# write.csv(res.portf.lvl.mult,"Results/res_mult.csv", row.names = FALSE)


### Various plots ###

# Annual results, portfolio level
```

```
# Portfolio A
plot_estimates_by_period_total(portfolio_results = final.result.table.annual[Portfolio == "Portfolio A" & rho == 0.12], portfo = "A")
plot_estimates_by_period_total(portfolio_results = final.result.table.annual[Portfolio == "Portfolio A" & rho == 0.24], portfo = "A")

# Portfolio B
plot_estimates_by_period_total(portfolio_results = final.result.table.annual[Portfolio == "Portfolio B" & rho == 0.12], portfo = "B")
plot_estimates_by_period_total(portfolio_results = final.result.table.annual[Portfolio == "Portfolio B" & rho == 0.24], portfo = "B")

# Annual results, risk class level

# Portfolio A
plot_estimates_by_grade(portfolio_results = final.result.table.annual[Portfolio == "Portfolio A" & rho == 0.12], portfo = "A")
plot_estimates_by_grade(portfolio_results = final.result.table.annual[Portfolio == "Portfolio A" & rho == 0.24], portfo = "A")

# Portfolio B
plot_estimates_by_grade(portfolio_results = final.result.table.annual[Portfolio == "Portfolio B" & rho == 0.12], portfo = "B")
plot_estimates_by_grade(portfolio_results = final.result.table.annual[Portfolio == "Portfolio B" & rho == 0.24], portfo = "B")

# All years

# Portfolio A
plot_estimates_mult(portfolio_results = final.result.table.mult.period[Portfolio == "Portfolio A" & rho == 0.12], portfo = "A")
plot_estimates_mult(portfolio_results = final.result.table.mult.period[Portfolio == "Portfolio A" & rho == 0.24], portfo = "A")

# Portfolio B
plot_estimates_mult(portfolio_results = final.result.table.mult.period[Portfolio == "Portfolio B" & rho == 0.12], portfo = "B")
plot_estimates_mult(portfolio_results = final.result.table.mult.period[Portfolio == "Portfolio B" & rho == 0.24], portfo = "B")


# Scores mapping to traffic light colors plots

# UCB
score_and_colors_plot(est = "UCB")

# Bayes
score_and_colors_plot(est = "Bayes")
```

E.2 Custom functions script

```
# Script that contains all custom functions

#####################
#  Data simulation  #
#####################

simulate_ldp_data <- function(portfolio_name,
                              years = 2015:2024,
                              grades = 1:6,
                              # PDs
                              grade_ttc_pd,
                              # True PDs
                              grade_pd,
                              # Obligor shares
                              grade_shares = c(0.3, 0.24, 0.18, 0.12, 0.09, 0.07),
                              # Counts of obligors (range)
                              n_total_range = c(1600, 1800),
                              seed = NULL) {

  if (!is.null(seed)) {
    set.seed(seed)
  }

  G <- length(grades)

  out_list <- vector("list", length(years))

  for (i in seq_along(years)) {
    y <- years[i]

    # Randomize count of obligors
    alpha <- grade_shares*150
    x <- rgamma(length(alpha), shape = alpha, rate = 1)
    shares_y <- x/sum(x)
```

```
    n_total <- sample(seq(n_total_range[1], n_total_range[2]), 1)
    n_g <- as.vector(rmultinom(1, size = n_total, prob = shares_y))

    # Default count per risk class
    k_g <- rbinom(G, size = n_g, prob = grade_pd)

    # Results
    grade_dt <- data.table(Portfolio = portfolio_name,
                           Year = y,
                           Grade = grades,
                           Obligor_count = n_g,
                           Default_count = k_g,
                           TTC_PD = grade_ttc_pd*100)

    total_dt <- data.table(Portfolio = portfolio_name,
                           Year = y,
                           Grade = "TOTAL",
                           Obligor_count = sum(n_g),
                           Default_count = sum(k_g),
                           TTC_PD = (sum(n_g*grade_ttc_pd)/sum(n_g))*100)

    out_list[[i]] <- rbind(grade_dt, total_dt)
  }

  result <- rbindlist(out_list)
  return(result[])

}


#####################################
#     One period PD estimation      #
#####################################

eps <- 1e-12

# Definition of PIT PD
G <- function(lambda, rho, y) {
  pnorm((qnorm(lambda) - sqrt(rho)*y)/sqrt(1 - rho))
}

# PD upper bound estimator
PT_upper_bound <- function(n, k, rho, alpha) {

  target <- (1 - alpha)

  F_of_p <- function(pd) {
    integrand <- function(y) {
      dnorm(y)*pbinom(k,
                      size = n,
                      prob = G(pd, rho, y))
    }

    stats::integrate(integrand,
                     lower = -Inf,
                     upper = Inf)$value

  }

  func <- function(pd) {
    F_of_p(pd) - target
  }

  uniroot(func, lower = 0, upper = 1)$root
}

# PD Bayes mean estimators
P_x_eq_k <- function(lambda, n, k, rho) {
  vapply(lambda,
         function(lam) {

           if (!is.finite(lam) | lam <= 0 | lam >= 1) return(0)

           integrand <- function(y) {
             dnorm(y)*dbinom(k, size = n, prob = G(lam, rho, y))
           }
```

```
           stats::integrate(integrand,
                            lower = -Inf,
                            upper = Inf)$value # Integrating through all y values
         },
  numeric(1))

}

bayes_mean_conservative <- function(n, k, rho) {

  num_integrand <- function(lam) {
    lam*P_x_eq_k(lam, n, k, rho)/(1 - lam)
  }

  den_integrand <- function(lam) {
    P_x_eq_k(lam, n, k, rho)/(1 - lam)
  }

  num <- integrate(num_integrand,
                   lower = eps,
                   upper = 1 - eps)$value

  den <- integrate(den_integrand,
                   lower = eps,
                   upper = 1 - eps)$value

  num/den
}

bayes_mean_neutral <- function(n, k, rho, u) {

  num_integrand <- function(lam) {
    lam*P_x_eq_k(lam, n, k, rho)
  }

  den_integrand <- function(lam) {
    P_x_eq_k(lam, n, k, rho)
  }

  num <- integrate(num_integrand,
                   lower = eps,
                   upper = u - eps)$value

  den <- integrate(den_integrand,
                   lower = eps,
                   upper = u - eps)$value

  num/den
}

#########################################
#     Multiple period PD estimation     #
#########################################

# Helper functions
Sigma_theta <- function(theta, T) {
  toeplitz(theta^(0:(T-1)))
}

logsumexp <- function(x) {
  m <- max(x)
  m + log(sum(exp(x - m)))
}

# Upper confidence bound estimator (Prop. 8, Eq. 23a)
ucb_multiperiod <- function(alpha, n, k, rho, S) {

  k_tot <- sum(k)

  func <- function(lambda) {
    pmat <- G(lambda, rho, S) # PIT PDs for every scneario and every period
    Ij <- rowSums(sweep(pmat, 2, n, '*'))
    mean(ppois(k_tot, Ij)) - (1 - alpha)
  }

  uniroot(func, lower = eps, upper = 1 - eps)$root
```

```
}

# Estimator
Multiple_period_estimator <- function(n,
                                      k,
                                      rho,
                                      theta,
                                      type,
                                      alpha = NA,
                                      p_ttc = NA,
                                      eps = 1e-4,
                                      grid_m = 2000L,
                                      return_posterior = FALSE) {

  Years <- length(n)

  set.seed(36)
  n_mc <- 16000L
  m <- 2500L

  # systemic factors
  Sigma <- Sigma_theta(theta = theta,
                       T = Years)

  S <- mvtnorm::rmvnorm(n = n_mc,
                        sigma = Sigma)

  # log likelihood for one lambda
  loglik_lambda <- function(lambda) {
    p_mat <- G(lambda, rho, S)
    p_mat <- pmin(pmax(p_mat, eps), 1 - eps)
    ll_j  <- sweep(log(p_mat), 2, k, `*`) + sweep(log(1 - p_mat), 2, (n - k), `*`)
    logsumexp(rowSums(ll_j))
  }

  if (type == "Conservative") {

    # grid for summation
    u_grid <- (0:(m - 1))/m
    u_grid[u_grid < eps] <- eps
    u_grid[u_grid > 1 - eps] <- 1 - eps

    # numerator and denominator initial values
    num <- 0
    den <- 0

    # 25b approximation
    for (ui in u_grid) {
      ls <- loglik_lambda(ui)
      num <- num + (ui/(1 - ui))*exp(ls)
      den <- den + (1/(1 - ui))*exp(ls)
    }

    lambda_hat <- as.numeric(num/den) # estimate

  } else if (type == "Uniform") {

    # grid for summation
    u_neutral <- 1
    u_grid <- (0:m)/m*u_neutral
    u_grid[u_grid < eps] <- eps
    u_grid[u_grid > 1 - eps] <- 1 - eps

    # numerator and denominator initial values
    num <- 0
    den <- 0

    # 25c approximation
    for (ui in u_grid) {
      ls <- loglik_lambda(ui)
      num <- num + ui*exp(ls)
      den <- den + 1*exp(ls)
    }

    lambda_hat <- as.numeric(num/den) # estimate

  } else if (type == "UCB") {
```

```r
  return(ucb_multiperiod(alpha, n, k, rho, S))

}

if (!return_posterior) {

  return(lambda_hat)

}

# log of prior density
log_prior_fun <- function(lambda, prior = c("Conservative", "Uniform")) {
  prior <- match.arg(prior)
  if (prior == "Conservative") {

    logdens <- -log(1 - lambda) - log(-log(eps))

  } else if (prior == "Uniform"){

    logdens <- -log(1 - eps)

  }

  ifelse(lambda > 0 & lambda < (1 - eps), logdens, -Inf)

}

# Create lambda grid
lam_grid <- seq(eps, 1 - eps, length.out = grid_m)

# weights
dlam <- diff(lam_grid) # same spacing between grid points
w <- numeric(length(lam_grid))
w[1] <- 0.5*dlam[1] # first weight - half a step
w[length(lam_grid)] <- 0.5*dlam[length(dlam)] # last weight - half a step
if (length(lam_grid) > 2) {
  w[2:(length(lam_grid)-1)] <- dlam[-length(dlam)]
}

# log-likelihood for defined grid
loglik_vec <- vapply(lam_grid, loglik_lambda, numeric(1))

# Prior
log_prior <- log_prior_fun(lambda = lam_grid, prior = type)

# unnormalized log posterior on definedd grid
log_post <- loglik_vec + log_prior

# log normalization constant
logZ <- {m <- max(log_post + log(w))
         m + log(sum(exp((log_post + log(w)) - m)))}

# posterior pdf at grid
log_pdf <- log_post - logZ
pdf <- exp(log_pdf)

# posterior mean
post_mean <- {m <- max(log_pdf + log(w))
              num <- sum(exp((log_pdf + log(w)) - m)*lam_grid)
              den <- sum(exp((log_pdf + log(w)) - m))
              num/den}

# CDF
cum_num <- {m <- max(log_post + log(w))
            cumsum(exp((log_post + log(w)) - m))}

C <- cum_num/cum_num[length(cum_num)]
C[length(C)] <- 1

# single tail probability
a <- min(max(p_ttc, eps), 1 - eps)
# evaluation of CDF at a by interpolation
cdf_at_a <- {
  idx <- findInterval(a, lam_grid, all.inside = TRUE) + 1L
  if (idx == 1L) {
    0}
```

```
    else {
      x0 <- lam_grid[idx - 1]
      x1 <- lam_grid[idx]
      y0 <- C[idx - 1]
      y1 <- C[idx]
      y0 + (a - x0)*(y1 - y0)/(x1 - x0 + 1e-16)}
  }

  tail_prob <- max(0, min(1, 1 - cdf_at_a))

  list(lambda_hat = lambda_hat,
    posterior = list(grid = lam_grid,
                     pdf = pdf,
                     mean = post_mean,
                     tail_prob  = tail_prob,
                     prior_type = type))
}

###########################
#     Score functions     #
###########################

# Bayesian tail probability
score_q <- function(q) {
  ifelse(is.na(q), NA_real_,
         ifelse(q <= 0.50, q/0.50,
                ifelse(q <= 0.75, 1 + (q - 0.50)/0.25,
                       2 + (q - 0.75)/0.25)
    )
  )
}

# log ratio
score_R <- function(R) {
  Rmax <- log(3)
  ifelse(is.na(R), NA_real_,
         ifelse(R <= 0, 0,
                pmin(3, 3*(R/Rmax))
    )
  )
}

###########################
#     Various plots       #
###########################

# Scores to traffic ligt mapping plot
add_rectangles <- function(xmin, xmax) {
  rect(xmin, 0, xmax, 1, col = adjustcolor("palegreen", 0.75))
  rect(xmin, 1, xmax, 2, col = adjustcolor("khaki1", 0.75))
  rect(xmin, 2, xmax, 3, col = adjustcolor("lightcoral", 0.75))
}

score_and_colors_plot <- function(est) {
  if (est == "UCB") {

    Rmax <- log(3)
    thr <- c(0, Rmax/3, 2*Rmax/3, Rmax)

    R <- seq(-0.5, Rmax + 0.5, length.out = 2001)
    S_R <- score_R(R)

    par(mar = c(4.5, 5, 1.5, 1))
    par(xaxs = "i", yaxs = "i")
    plot(R, S_R, type = "n",
         xlab = expression(italic(R)[alpha]),
         ylab = expression(Score),
         ylim = c(0, 3.004), yaxt = "n", xaxt = "n")

    add_rectangles(min(R), max(R))
    lines(R, S_R, lwd = 2)
    abline(v = thr, lty = 3, col = "grey40")

    axis(2, at = 0:3, las = 1)
    axis(1, at = thr, labels = c("0", "log(3)/3", "2log(3)/3", "log(3)"), las = 1)

  } else if (est == "Bayes") {
```

```
    q <- seq(0, 1, length.out = 2001)
    S_q <- score_q(q)

    par(mar = c(4.5, 5, 1.5, 1))
    par(xaxs = "i", yaxs = "i")
    plot(q, S_q, type = "n",
         xlab = expression(italic(q[t])),
         ylab = expression(Score),
         ylim = c(0, 3.004), yaxt = "n", xaxt = "n")

    add_rectangles(0, 1)
    lines(q, S_q, lwd = 2)
    abline(v = c(0, 0.5, 0.75, 1), lty = 3, col = "grey40")

    axis(2, at = 0:3, las = 1)
    axis(1, at = c(0, 0.5, 0.75, 1), labels = c("0", "0.5", "0.75", "1"), las = 1)

  }
}

plot_portfolio_composition <- function(portfolio) {

  tbl <- with(portfolio,
              tapply(Obligor_count,
                     list(Grade, Year),
                     sum,
                     na.rm = TRUE))

  # defaults in grade
  tbl_def <- with(portfolio,
                  tapply(Default_count,
                         list(Grade, Year),
                         sum,
                         na.rm = TRUE))
  # colors for grades
  cols <- colorRampPalette(c("darkgreen", "yellow", "red"))(nrow(tbl))

  # Layout
  layout(matrix(c(1, 2), nrow = 1), widths = c(10, 1))

  # Bar plot
  par(mar = c(5, 4, 4, 6))

  portf_plot <- barplot(tbl,
                        beside = FALSE,
                        col = cols,
                        border = "white",
                        xlab = "Year",
                        ylab = "Number of obligors",
                        main = "",
                        las  = 1,
                        ylim = c(0, 2000))

  # Annotate defaults in grades
  cum_h <- apply(tbl, 2, cumsum)
  seg_center <- cum_h - tbl/2

  for (i in seq_len(nrow(tbl))) {
    for (j in seq_len(ncol(tbl))) {
      def <- tbl_def[i, j]
      if (is.finite(def) & def > 0 & tbl[i, j] > 0) {
        text(x = portf_plot[j],
             y = seg_center[i, j],
             labels = def,
             cex = 0.9,
             col = "black",
             font = 2)
      }
    }
  }

  # legend
  par(mar = c(5, 4, 4, 8))
  par(xpd = NA)

  legend("topright",
```

```r
         inset = c(-0.15, 0),
         legend = rownames(tbl),
         fill = cols,
         title = "Risk class",
         bty = "n",
         cex = 0.95)

}

plot_estimates_by_period_total <- function(portfolio_results, portfo){

  col.to.plot <- c("TTC_PD", "ODF",
                   "75% PT upper bound", "90% PT upper bound", "99% PT upper bound",
                   "Neutral Bayesian on (0, 1)", "Conservative Bayesian")

  cols <- c("TTC_PD" = "black",
            "ODF" = "grey40",
            "75% PT upper bound" = "#4DAF4A",
            "90% PT upper bound" = "#E41A1C",
            "99% PT upper bound" = "orange",
            "Neutral Bayesian on (0, 1)" = "#7B3F98",
            "Conservative Bayesian" = "#00BFC4")

  ltys <- c("TTC_PD" = 1,
            "ODF" = 2,
            "75% PT upper bound" = 3,
            "90% PT upper bound" = 3,
            "99% PT upper bound" = 3,
            "Neutral Bayesian on (0, 1)" = 3,
            "Conservative Bayesian" = 3)

  lwds <- c("TTC_PD" = 2,
            "ODF" = 1.5,
            "75% PT upper bound" = 1.5,
            "90% PT upper bound" = 1.5,
            "99% PT upper bound" = 1.5,
            "Neutral Bayesian on (0, 1)" = 1.5,
            "Conservative Bayesian" = 1.5)

  grades.to.plot <- "TOTAL"

  par(mar = c(3.2, 3.2, 2.0, 11), # more space on the right margin
      mgp = c(2, 0.7, 0),
      lend = 2,
      ljoin = 2,
      xpd = FALSE)

  for (grade in grades.to.plot) {
    dt.tmp.grade <- portfolio_results[Grade == grade, c("Year", col.to.plot), with = FALSE][order(Year)]

    y.range  <- range(unlist(dt.tmp.grade[, ..col.to.plot]), na.rm = TRUE)

    pad <- if (is.finite(diff(y.range))) {
      diff(y.range)*0.08
    } else {0.5}

    y.lim <- c(max(0, y.range[1] - pad), y.range[2] + pad)

    plot(x = dt.tmp.grade$Year,
         y = dt.tmp.grade$TTC_PD,
         type = "l",
         ylim = y.lim,
         xlab = "Year",
         ylab = "ODR/PD (%)",
         col = "black",
         lty = 1,
         lwd = 2,
         main = "",
         xaxt = "n",
         panel.first = {abline(h = pretty(y.lim), col = "grey90", lty = 1)
                        abline(v = dt.tmp.grade$Year, col = "grey90", lty = 1)})

    axis(1, at = dt.tmp.grade$Year, labels = dt.tmp.grade$Year)

    points(x = dt.tmp.grade$Year,
           y = dt.tmp.grade$TTC_PD,
           pch = 15,
```

```
          cex = 0.9,
          col = "black")

    for (var in setdiff(col.to.plot, "TTC_PD")) {

      y <- dt.tmp.grade[[var]]

      lines(x = dt.tmp.grade$Year,
            y = y,
            col = cols[var],
            lty = ltys[var],
            lwd = lwds[var])

      points(x = dt.tmp.grade$Year,
             y = y,
             pch = 15,
             cex = 0.9,
             col = cols[var])

    }
  }

  par(xpd = NA)

  legend("right",
         inset = c(-0.28, 0),
         legend = c(switch(portfo,
                           "A" = expression(PD[italic(A)]),
                           "B" = expression(PD[italic(B)])),
                    "ODR",
                    expression(UCB[`75%`]), expression(UCB[`90%`]), expression(UCB[`99%`]),
                    "Bayes neutral", "Bayes conservative"),
         col = cols[col.to.plot],
         lty = ltys[col.to.plot],
         lwd = lwds[col.to.plot],
         pch = 15,
         bty = "n",
         cex = 0.9)
}


plot_estimates_by_grade <- function(portfolio_results, portfo){

  col.to.plot <- c("TTC_PD", "ODF",
                   "75% PT upper bound", "90% PT upper bound", "99% PT upper bound",
                   "Neutral Bayesian on (0, 1)", "Conservative Bayesian")

  cols <- c("TTC_PD" = "black",
            "ODF" = "grey40",
            "75% PT upper bound" = "#4DAF4A",
            "90% PT upper bound" = "#E41A1C",
            "99% PT upper bound" = "orange",
            "Neutral Bayesian on (0, 1)" = "#7B3F98",
            "Conservative Bayesian" = "#00BFC4")

  ltys <- c("TTC_PD" = 1,
            "ODF" = 2,
            "75% PT upper bound" = 3,
            "90% PT upper bound" = 3,
            "99% PT upper bound" = 3,
            "Neutral Bayesian on (0, 1)" = 3,
            "Conservative Bayesian" = 3)

  lwds <- c("TTC_PD" = 2,
            "ODF" = 1.5,
            "75% PT upper bound" = 1.5,
            "90% PT upper bound" = 1.5,
            "99% PT upper bound" = 1.5,
            "Neutral Bayesian on (0, 1)" = 1.5,
            "Conservative Bayesian" = 1.5)

  yrs.to.plot <- unique(portfolio_results$Year)

  dt.tmp.all <- portfolio_results[Grade != "TOTAL", c("Grade", col.to.plot), with = FALSE]
  y.range  <- range(unlist(dt.tmp.all[, ..col.to.plot]), na.rm = TRUE)

  pad <- if (is.finite(diff(y.range))) {
```

```
    diff(y.range)*0.08
  } else {0.5}

  y.lim <- c(max(0, y.range[1] - pad), y.range[2] + pad)

  layout(matrix(1:12, 4, 3, byrow = TRUE))

  par(mar = c(3.2, 3.2, 2.0, 0.8),
      mgp = c(2, 0.7, 0),
      lend = 2,
      ljoin = 2)

  for (yr in yrs.to.plot) {
    dt.tmp.yr <- portfolio_results[Year == yr & Grade != "TOTAL", c("Grade", col.to.plot), with = FALSE][order(Grade)]

    plot(x = dt.tmp.yr$Grade,
         y = dt.tmp.yr$TTC_PD,
         type = "l",
         ylim = y.lim,
         xlab = "Risk class",
         ylab = "ODR/PD (%)",
         col = "black",
         lty = 1,
         lwd = 2,
         main =  paste("Year: ", yr),
         panel.first = { grid(col = "grey90", lty = 1) })

    points(x = dt.tmp.yr$Grade,
           y = dt.tmp.yr$TTC_PD,
           pch = 15,
           cex = 0.9,
           col = "black")

    for (var in setdiff(col.to.plot, "TTC_PD")) {
      y <- dt.tmp.yr[[var]]

      lines(x = dt.tmp.yr$Grade,
            y = y,
            col = cols[var],
            lty = ltys[var],
            lwd = lwds[var])
      points(x = dt.tmp.yr$Grade,
             y = y,
             pch = 15,
             cex = 0.9,
             col = cols[var])
    }

  }

  par(mar = c(0, 0, 0, 0))
  plot.new()
  legend("left",
         legend = c(switch(portfo,
                           "A" = expression(PD[italic(A)]),
                           "B" = expression(PD[italic(B)])),
                    "ODR",
                    expression(UCB[`75%`]), expression(UCB[`90%`]), expression(UCB[`99%`]),
                    "Bayes neutral", "Bayes conservative"),
         col = cols[col.to.plot],
         lty = ltys[col.to.plot],
         lwd = lwds[col.to.plot],
         pch = 15,
         bty = "n",
         cex = 0.95,
         title = "")
}



plot_estimates_mult <- function(portfolio_results, portfo){

  col.to.plot <- c("TTC_PD", "ODF",
                   "Multiple period: 75% upper bound", "Multiple period: 90% upper bound",
                   "Multiple period: 99% upper bound",
                   "Multiple period: Neutral Bayesian on (0, 1)", "Multiple period: Conservative Bayesian")
```

```
cols <- c("TTC_PD" = "black",
          "ODF" = "grey40",
          "Multiple period: 75% upper bound" = "#4DAF4A",
          "Multiple period: 90% upper bound" = "#E41A1C",
          "Multiple period: 99% upper bound" = "orange",
          "Multiple period: Neutral Bayesian on (0, 1)" = "#7B3F98",
          "Multiple period: Conservative Bayesian" = "#00BFC4")

ltys <- c("TTC_PD" = 1,
          "ODF" = 2,
          "Multiple period: 75% upper bound" = 3,
          "Multiple period: 90% upper bound" = 3,
          "Multiple period: 99% upper bound" = 3,
          "Multiple period: Neutral Bayesian on (0, 1)" = 3,
          "Multiple period: Conservative Bayesian" = 3)

lwds <- c("TTC_PD" = 2,
          "ODF" = 1.5,
          "Multiple period: 75% upper bound" = 1.5,
          "Multiple period: 90% upper bound" = 1.5,
          "Multiple period: 99% upper bound" = 1.5,
          "Multiple period: Neutral Bayesian on (0, 1)" = 1.5,
          "Multiple period: Conservative Bayesian" = 1.5)

dt.tmp <- copy(portfolio_results)[Grade != "TOTAL"][order(Grade)]

y.range  <- range(unlist(dt.tmp[, ..col.to.plot]), na.rm = TRUE)

pad <- if (is.finite(diff(y.range))) {
  diff(y.range)*0.08
} else {0.5}

y.lim <- c(max(0, y.range[1] - pad), y.range[2] + pad)

par(mfrow = c(1, 1),
    mar = c(5.1, 4.1, 4.1, 9.5),
    mgp = c(3, 1, 0),
    oma = c(0, 0, 0, 0),
    las = 0,
    cex = 1,
    lend = "round",
    ljoin = "round",
    xpd = FALSE)

plot(x = dt.tmp$Grade,
     y = dt.tmp$TTC_PD,
     type = "l",
     ylim = y.lim,
     xlab = "Risk class",
     ylab = "ODR/PD (%)",
     col = "black",
     lty = 1,
     lwd = 2,
     main = "",
     panel.first = {grid(col = "grey90", lty = 1)})

points(x = dt.tmp$Grade,
       y = dt.tmp$TTC_PD,
       pch = 15,
       cex = 0.9,
       col = "black")

for (var in setdiff(col.to.plot, "TTC_PD")) {
  y <- dt.tmp[[var]]
  if (all(is.na(y))) next
  lines(x = dt.tmp$Grade,
        y = y,
        col = cols[var],
        lty = ltys[var],
        lwd = lwds[var])
  points(x = dt.tmp$Grade,
         y = y,
         pch = 15,
         cex = 0.9,
         col = cols[var])
}
```

```
  legend("right",
         legend = c(switch(portfo,
                           "A" = expression(PD[italic(A)]),
                           "B" = expression(PD[italic(B)])),
                    "ODR",
                    expression(UCB[`75%`]), expression(UCB[`90%`]), expression(UCB[`99%`]),
                    "Bayes neutral", "Bayes conservative"),
         col = cols[col.to.plot],
         lty = ltys[col.to.plot],
         lwd = lwds[col.to.plot],
         pch = 15,
         bty = "n",
         cex = 0.95,
         title = "",
         xpd = NA,
         inset = c(-0.3, 0))
}

# Prior PDF and posterior PDF
prior_conservative <- function(lambda, eps = 1e-4) {
  ifelse(lambda > 0 & lambda < (1 - eps),
         1/((1 - lambda)*(-log(eps))),
         0)
}

prior_uniform <- function(lambda, eps = 1e-4) {
  ifelse(lambda > 0 & lambda < (1 - eps),
         1/(1 - eps),
         0)
}

norm_const <- function(n, k, rho, eps = 1e-4, prior = c("unif", "cons")) {

  prior <- match.arg(prior)

  integrand <- switch(prior,
                      "unif" = function(lam) {prior_uniform(lam, eps)*P_x_eq_k(lam, n, k, rho)},
                      "cons" = function(lam) {prior_conservative(lam, eps)*P_x_eq_k(lam, n, k, rho)}
  )

  stats::integrate(integrand, lower = 0, upper = (1 - eps))$value

}

posterior_pdf <- function(lambda, n, k, rho, eps = 1e-4, prior = c("unif", "cons")) {

  prior <- match.arg(prior)

  denom <- norm_const(n, k, rho, eps, prior)

  # define numerator
  sapply(lambda, function(lam) {
    if (prior == "unif") {
      num <- P_x_eq_k(lam, n, k, rho)*prior_uniform(lam, eps)
    } else if (prior == "cons") {
      num <- P_x_eq_k(lam, n, k, rho)*prior_conservative(lam, eps)
    }

    # Posterior pdf
    num/denom

  })

}

calculate_tail_prob <- function(n, k, rho, p_ttc, eps = 1e-4, prior = c("unif", "cons")) {

  prior <- match.arg(prior)

  posterior <- function(l) {
    post <- posterior_pdf(l, n, k, rho, eps, prior)
    post[!is.finite(post)] <- 0
    post
  }

  stats::integrate(posterior, lower = p_ttc, upper = (1 - eps),
                   rel.tol = 1e-6, subdivisions = 2e4)$value}
```