

Bayesian Subset Selection for Weibull Populations

Jezerca Hodaj, Gary C. McDonald

Department of Mathematics and Statistics, Oakland University, Rochester, MI, USA

Email: jhodaj@oakland.edu, mcdonald@oakland.edu

How to cite this paper: Hodaj, J. and McDonald, G.C. (2026) Bayesian Subset Selection for Weibull Populations. *Applied Mathematics*, 17, 526-571.

<https://doi.org/10.4236/am.2026.178030>

Received: July 23, 2026

Accepted: August 21, 2026

Published: August 24, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International

License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Selecting the best population is an important problem in reliability analysis, particularly when lifetime data are right censored and the populations are characterized by threshold behavior. This paper presents a Bayesian subset selection procedure for multiple populations modeled by three-parameter Weibull distributions. The best population is defined as the population having the largest threshold parameter, which represents the minimum lifetime before failure can occur. Posterior probabilities of being the best are estimated based on Markov Chain Monte Carlo samples, and the selected subset is defined as the smallest collection of populations whose cumulative posterior probability of containing the best population reaches a prespecified level. Posterior inference is conducted using a random-walk Metropolis algorithm with a transformed parameterization that maintains the constraints of the three-parameter Weibull model. Convergence and sampling efficiency are evaluated through graphical and numerical diagnostics. The methodology is demonstrated using a semi-simulated fatigue-life study based on right-censored Alloy T7987 data. Model adequacy is assessed with posterior predictive methods, and posterior estimates are reported for the Weibull threshold, shape, and scale parameters for each population. In the Alloy-based study, the posterior probability of being best was distributed primarily among three populations, reflecting substantial uncertainty in the population ranking. At a target posterior probability level of 0.95, these three populations constituted the selected subset. The findings demonstrate that the proposed procedure incorporates posterior uncertainty and prevents the forced selection of a single population when the lifetime data do not offer clear separation. This approach offers a flexible Bayesian framework for selection among right-censored Weibull reliability populations.

Keywords

Bayesian Subset Selection, Three-Parameter Weibull Distribution, Right-Censored Data, Reliability Analysis, Posterior Probability of Being

1. Introduction

Ranking and selecting among multiple populations is a key statistical problem with many uses in reliability, manufacturing, engineering design, and quality improvement. Often, the main goal is not just to estimate parameters, but to find the population that performs best based on a meaningful scientific standard. Traditional ranking-and-selection methods usually pick either a single population or a subset that is likely to include the best one. Early studies focused on populations with normal distributions and laid the groundwork for indifference-zone and subset-selection methods [1] [2]. Later research expanded these ideas to more types of distributions and applications, as discussed in detail by Gupta and Panchapakesan [3], Bechhofer *et al.* [4], and Gibbons *et al.* [5].

In reliability analysis, we usually assess how populations perform by looking at lifetime measures like average lifetime, certain percentiles, survival rates at specific times, or other parameters with clear physical meaning. Reliability data often include right-censored observations, since some units are still working at the end of the study or are removed before failing. Because of this, methods designed for complete data may not work as intended, and both parameter estimation and population comparisons need to account for censored data in the likelihood [6]-[8]. As a result, ranking and subset-selection methods for censored populations have become an important topic in reliability research [9].

The Weibull distribution is commonly used in lifetime analysis because it can model different patterns of failure rates. Its shape parameter lets it represent decreasing, constant, or increasing failure rates, while the scale parameter sets the typical lifetime for the population [10] [11]. The three-parameter Weibull model also includes a threshold, or location, parameter that marks the earliest possible failure time. In this setup, failures cannot happen before the threshold. This threshold is especially useful for reliability studies when populations differ in the earliest time failures can occur.

Adding a threshold parameter brings extra statistical challenges. The smallest observed lifetime limits the possible values for this parameter, and estimating it becomes harder when data are censored or when the likelihood behaves oddly near the boundary. Maximum likelihood estimation for Weibull models with both complete and censored data has been studied in depth [12], and several methods exist for estimating Weibull parameters and reliability percentiles [13] [14]. Still, there can be a lot of uncertainty in the threshold parameter, especially if populations have similar lifetime distributions or only a few failures are observed. So, simply picking the population with the highest threshold estimate might hide important uncertainty in the ranking.

Researchers have developed several frequentist methods for selecting or ranking Weibull populations. Kingston and Patel proposed ways to choose the best among several Weibull populations and to select restricted subsets [15] [16]. Hsu [17] and Gupta and Miescke [18] studied optimal and two-stage selection methods, while Tseng and Wu [19] looked at selecting more reliable Weibull populations when data are Type-II censored. Similar methods have been created for exponential populations, including ways to select good populations and compare threshold parameters [20]-[23]. Together, these studies highlight the need for formal selection rules that consider sampling variability, instead of just using estimated rankings.

Traditional subset-selection methods are usually set up to meet certain probability requirements through repeated sampling. For example, a method might be designed so that the chosen subset is likely to include the true best population with at least a specified probability across a range of parameter values. A Bayesian approach offers a different perspective. Once data are collected, uncertainty about the population parameters is described by their joint posterior distribution. The posterior probability that each population is best can be estimated directly, making it possible to measure the evidence for each ranking. Bayesian ranking and subset-selection methods have been used in areas like traffic safety and engineering design [24] [25], and have also been studied for censored Weibull populations and Weibull-component models [26] [27]. Unlike methods that rely only on point estimates, using posterior probabilities allows for keeping several possible candidates when the data do not clearly show which population is best.

This paper presents a Bayesian subset-selection method for several populations modeled with three-parameter Weibull distributions and right-censored data. Here, the best population is the one with the largest threshold parameter. For each population, we estimate the posterior probability of having the largest threshold using Markov Chain Monte Carlo samples. Populations are ranked by these probabilities, and the selected subset is the smallest group whose combined posterior probability reaches or exceeds a chosen target. This way, the method does not force the selection of a single population when there is still a lot of uncertainty.

We use a random-walk Metropolis algorithm for posterior inference, working in a transformed parameter space. This transformation keeps the Weibull shape and scale parameters positive and makes sure the threshold stays below the smallest observed lifetime. The Metropolis algorithm, along with Hastings' generalization, provides a flexible way to sample from posterior distributions that do not have closed-form solutions [28]-[30]. We check convergence and sampling quality with both graphical and numerical diagnostics, and we assess model fit using posterior predictive methods [31]-[33].

We demonstrate the proposed method with a semi-simulated fatigue-life study using right-censored Alloy T7987 data. The study keeps the same sample size and main features as the original Alloy data, but creates several related reliability populations with different Weibull characteristics. This setup lets us test the subset-

selection method in a setting that combines real engineering data with controlled differences between populations. We present the posterior estimates of the Weibull parameters, the posterior probabilities of being best, and the selected subset. We also check how sensitive the population ranking is to using different reliability criteria based on Weibull lifetime percentiles.

The rest of the paper is organized as follows. Section 2 covers the three-parameter Weibull model, the likelihood for right-censored data, and the transformed parameterization. Section 3 explains the prior and posterior distributions. Section 4 defines the best population, the posterior probabilities, and the Bayesian subset-selection rule. Section 5 describes the random-walk Metropolis method and how we check for convergence. Section 6 presents the Alloy T7987-based study and its results. Finally, we discuss the advantages, limitations, and possible extensions of our approach.

2. Three-Parameter Weibull Model

2.1. Three-Parameter Weibull Distribution

For population i , let the lifetime observations follow a three-parameter Weibull distribution with threshold parameter γ_i , scale parameter η_i , and shape parameter β_i . The probability density function is

$$f_i(t|\gamma_i, \eta_i, \beta_i) = \frac{\beta_i}{\eta_i} \left(\frac{t - \gamma_i}{\eta_i} \right)^{\beta_i - 1} \exp \left[- \left(\frac{t - \gamma_i}{\eta_i} \right)^{\beta_i} \right], \quad t > \gamma_i, \quad (2.1)$$

where $\eta_i > 0$ and $\beta_i > 0$. The corresponding distribution function is

$$F_i(t|\gamma_i, \eta_i, \beta_i) = 1 - \exp \left[- \left(\frac{t - \gamma_i}{\eta_i} \right)^{\beta_i} \right], \quad t > \gamma_i, \quad (2.2)$$

and the survival function is

$$S_i(t|\gamma_i, \eta_i, \beta_i) = \exp \left[- \left(\frac{t - \gamma_i}{\eta_i} \right)^{\beta_i} \right], \quad t > \gamma_i. \quad (2.3)$$

The threshold parameter γ_i determines the lower endpoint of the lifetime distribution. Under the model, failures cannot occur before γ_i . The scale parameter η_i controls the scale of the lifetimes above the threshold, while the shape parameter β_i determines the shape of the distribution and the behavior of its hazard function [10] [11].

The hazard function is

$$h_i(t|\gamma_i, \eta_i, \beta_i) = \frac{f_i(t|\gamma_i, \eta_i, \beta_i)}{S_i(t|\gamma_i, \eta_i, \beta_i)} = \frac{\beta_i}{\eta_i} \left(\frac{t - \gamma_i}{\eta_i} \right)^{\beta_i - 1}, \quad t > \gamma_i. \quad (2.4)$$

Thus, the hazard rate is decreasing when $\beta_i < 1$, constant when $\beta_i = 1$, and increasing when $\beta_i > 1$. This flexibility makes the Weibull distribution useful in fatigue-life and other reliability applications.

2.2. Likelihood for Right-Censored Data

The semi-simulated Alloy reliability study includes right-censored observations. Let t_{ij} denote the observed time for unit j in population i , and let

$$d_{ij} = \begin{cases} 1, & \text{if the observation is an exact failure time,} \\ 0, & \text{if the observation is right-censored.} \end{cases} \quad (2.5)$$

The likelihood contribution for observation j in population i is

$$\left[f_i(t_{ij} | \gamma_i, \eta_i, \beta_i) \right]^{d_{ij}} \left[S_i(t_{ij} | \gamma_i, \eta_i, \beta_i) \right]^{1-d_{ij}}. \quad (2.6)$$

Therefore, the likelihood for population i is

$$L_i(\gamma_i, \eta_i, \beta_i | \mathbf{t}, \mathbf{d}_i) = \prod_{j=1}^n \left[f_i(t_{ij} | \gamma_i, \eta_i, \beta_i) \right]^{d_{ij}} \left[S_i(t_{ij} | \gamma_i, \eta_i, \beta_i) \right]^{1-d_{ij}}. \quad (2.7)$$

The corresponding log-likelihood is

$$\ell_i(\gamma_i, \eta_i, \beta_i) = \sum_{j=1}^n d_{ij} \log f_i(t_{ij} | \gamma_i, \eta_i, \beta_i) + \sum_{j=1}^n (1-d_{ij}) \log S_i(t_{ij} | \gamma_i, \eta_i, \beta_i). \quad (2.8)$$

Substituting the Weibull density and survival functions gives

$$\begin{aligned} \ell_i(\gamma_i, \eta_i, \beta_i) = & \sum_{j=1}^n d_{ij} \left[\log(\beta_i) - \beta_i \log(\eta_i) + (\beta_i - 1) \log(t_{ij} - \gamma_i) \right] \\ & - \sum_{j=1}^n \left(\frac{t_{ij} - \gamma_i}{\eta_i} \right)^{\beta_i}. \end{aligned} \quad (2.9)$$

2.3. Transformation Used for Posterior Computation

The threshold parameter must satisfy

$$\gamma_i < t_{i(1)}, \quad (2.10)$$

where

$$t_{i(1)} = \min_{1 \leq j \leq n} t_{ij}. \quad (2.11)$$

Transformations of constrained parameters are commonly used to improve computational behavior and simplify posterior simulation [34] [35]. To enforce the parameter constraints during posterior simulation, the following transformation is used:

$$\theta_{i1} = \log(\beta_i), \quad \theta_{i2} = \log(\eta_i), \quad \theta_{i3} = \log(\delta_i), \quad (2.12)$$

where

$$\delta_i = t_{i(1)} - \gamma_i - \varepsilon, \quad (2.13)$$

and ε is a small positive constant. Equivalently,

$$\gamma_i = t_{i(1)} - \exp(\theta_{i3}) - \varepsilon. \quad (2.14)$$

Because $\exp(\theta_{i3}) > 0$, this parameterization guarantees that

$$\gamma_i < t_{i(1)}.$$

For every proposed value of θ_{i3} . The logarithmic transformations similarly guarantee that $\beta_i > 0$ and $\eta_i > 0$.

The prior distributions assigned to the transformed model parameters and the resulting posterior distribution are presented in the following section.

3. Bayesian Specification

3.1. Prior Distributions

The specification of priors for the three-parameter Weibull distribution requires consideration of the positivity constraints on the shape and scale parameters, as well as the threshold constraint on the location parameter. Independent prior distributions are assigned to the Weibull parameters for each population. Since $\beta_i > 0$ and $\eta_i > 0$, gamma prior distributions are chosen for the shape and scale parameters. The threshold parameter is handled by introducing the positive distance parameter,

$$\delta_i = t_{i(1)} - \gamma_i - \varepsilon,$$

where $\varepsilon = 10^{-6}$ is a fixed numerical offset used to keep the threshold parameter away from the boundary $t_{i(1)}$. This offset helps prevent numerical instability when evaluating terms involving $t_{ij} - \gamma_i$, is negligible relative to the scale of the observed fatigue lifetimes, and is not an estimated model parameter. The prior is assigned directly to the transformed parameter

$$\theta_{i3} = \log(\delta_i),$$

which is defined on the real line. Prior specification for Weibull reliability models and the use of formal prior distributions in Bayesian analysis are discussed in [36]-[39].

In the semi-simulated Alloy fatigue study, the prior distributions are specified as follows:

$$\beta_i \sim \text{Gamma}(\alpha_{\beta_i} = 6, \theta_{\beta_i} = 0.25), \quad (3.1)$$

$$\eta_i \sim \text{Gamma}(\alpha_{\eta_i} = 0.01, \theta_{\eta_i} = 10000), \quad (3.2)$$

and

$$\theta_{i3} = \log(\delta_i) \sim N(\mu_{\theta_{i3}} = 0, \sigma_{\theta_{i3}}^2 = 2^2). \quad (3.3)$$

In Equations (3.1) and (3.2), the second parameter represents the scale parameter of the gamma distribution.

The prior for η_i is specified on a scale that aligns with the observed fatigue lifetimes, while allowing for substantial uncertainty. The gamma prior for β_i respects the positivity constraint on the Weibull shape parameter while remaining sufficiently flexible for the data to inform posterior inference. The prior assigned to θ_{i3} is particularly important, as the threshold parameter is related to the smallest observed time through

$$\gamma_i = t_{i(1)} - \exp(\theta_{i3}) - \varepsilon.$$

Larger values of δ_i correspond to smaller values of γ_i , whereas smaller values of δ_i position the threshold nearer to the smallest observed lifetime. Therefore, the prior on θ_{i3} determines the proximity of the threshold to the lower bound of the observed data. The Alloy study investigates the sensitivity of posterior conclusions to the prior standard deviation of θ_{i3} .

3.2. Posterior Distribution on the Transformed Scale

For population i , let

$$\boldsymbol{\theta}_i = (\theta_{i1}, \theta_{i2}, \theta_{i3})^T = (\log(\beta_i), \log(\eta_i), \log(\delta_i))^T. \quad (3.4)$$

The original model parameters can be recovered from the transformed parameters using the following relationships:

$$\beta_i = \exp(\theta_{i1}), \quad \eta_i = \exp(\theta_{i2}), \quad \delta_i = \exp(\theta_{i3}), \quad (3.5)$$

and

$$\gamma_i = t_{i(1)} - \exp(\theta_{i3}) - \varepsilon. \quad (3.6)$$

Applying the right-censored likelihood from Section 2.2, the posterior density for population i on the transformed scale is proportional to the product of the likelihood, the prior densities, and the Jacobian of the transformation:

$$\pi(\boldsymbol{\theta}_i | \mathbf{t}_i, \mathbf{d}_i) \propto L_i(\gamma_i, \eta_i, \beta_i | \mathbf{t}_i, \mathbf{d}_i) \pi(\beta_i) \pi(\eta_i) \pi(\theta_{i3}) \left| \frac{\partial(\beta_i, \eta_i)}{\partial(\theta_{i1}, \theta_{i2})} \right|. \quad (3.7)$$

Because

$$\beta_i = \exp(\theta_{i1}) \text{ and } \eta_i = \exp(\theta_{i2}),$$

the Jacobian term is

$$\left| \frac{\partial(\beta_i, \eta_i)}{\partial(\theta_{i1}, \theta_{i2})} \right| = \exp(\theta_{i1}) \exp(\theta_{i2}) = \beta_i \eta_i. \quad (3.8)$$

No additional Jacobian term is necessary for θ_{i3} because the normal prior in Equation (3.3) is assigned directly to $\theta_{i3} = \log(\delta_i)$.

Therefore, the posterior density targeted in the analysis is

$$\pi(\boldsymbol{\theta}_i | \mathbf{t}_i, \mathbf{d}_i) \propto L_i(\gamma_i, \eta_i, \beta_i | \mathbf{t}_i, \mathbf{d}_i) \pi(\beta_i) \pi(\eta_i) \pi(\theta_{i3}) \exp(\theta_{i1}) \exp(\theta_{i2}). \quad (3.9)$$

Equivalently, the log-posterior density is given by

$$\begin{aligned} \ell_i(\boldsymbol{\theta}_i) = & \log L_i(\gamma_i, \eta_i, \beta_i | \mathbf{t}_i, \mathbf{d}_i) + \log \pi(\beta_i) + \log \pi(\eta_i) \\ & + \log \pi(\theta_{i3}) + \theta_{i1} + \theta_{i2} + C, \end{aligned} \quad (3.10)$$

where C is a constant independent of the model parameters. The posterior distribution does not have a closed form and is evaluated using the random-walk Metropolis algorithm described in Section 5.

4. Bayesian Subset Selection

The Alloy fatigue study involves a comparison of five independent three-parameter Weibull populations. The subset-selection procedure is described below for a

general case of k populations, with $k = 5$ in this application. In contrast to the frequentist probability of correct selection, which is evaluated over repeated samples with fixed parameter values, the Bayesian approach utilizes the posterior distribution obtained after observing the right-censored lifetime data. This method quantifies the posterior evidence that each population possesses the largest threshold parameter and constructs a subset whose posterior probability of containing the best population meets a prespecified criterion.

4.1. Definition of the Best Population

For population i , let

$$\boldsymbol{\psi}_i = (\gamma_i, \eta_i, \beta_i) \quad (4.1)$$

denote the vector of three-parameter Weibull parameters, where γ_i is the threshold parameter, η_i is the scale parameter, and β_i is the shape parameter.

The best population is defined as the population with the largest threshold parameter. Thus, population i is best if

$$\gamma_i = \max_{1 \leq j \leq k} \gamma_j. \quad (4.2)$$

This definition is motivated by reliability applications, where a larger threshold parameter indicates a greater lower endpoint of the lifetime distribution and, consequently, a later earliest possible failure time.

Accordingly, this criterion is intended for engineering decisions in which protection against early failure, or equivalently a later model-implied lower lifetime bound, is the primary objective. It is not intended as a universal ordering of Weibull populations. When the engineering objective concerns median lifetime, an upper lifetime percentile, expected lifetime, reliability at a specified time, or another performance measure, the corresponding reliability functional should determine the ranking criterion. The percentile comparisons reported in this paper are therefore supplementary analyses that illustrate how the population rankings may change under alternative reliability objectives; they are not part of the formal threshold-based subset-selection rule.

Let i^* denote the index of the best population:

$$i^* = \arg \max_{1 \leq j \leq k} \gamma_j. \quad (4.3)$$

Because the posterior distributions considered here are continuous, exact ties among the threshold parameters occur with probability zero.

4.2. Posterior Probability of Being Best

After observing the right-censored lifetime data, uncertainty about the Weibull parameters is represented by their posterior distribution. The posterior probability that population i is best is defined as

$$p_i = \Pr\left(\gamma_i = \max_{1 \leq j \leq k} \gamma_j \mid \mathbf{t}, \mathbf{d}\right), \quad (4.4)$$

Here, $\mathbf{t}$ and $\mathbf{d}$ represent the observed times and censoring indicators for all

populations. The values $p_1, \dots, p_k$ summarize the posterior evidence supporting each population as the best. Because one population is identified as best in each posterior draw,

$$\sum_{i=1}^k p_i = 1. \quad (4.5)$$

In practice, these probabilities are estimated from the retained posterior draws. After burn-in and thinning, suppose M draws of the threshold parameters are available:

$$\{\gamma_1^{(m)}, \dots, \gamma_k^{(m)}\}_{m=1}^M.$$

For draw m , the population having the largest sampled threshold parameter is recorded as the best population. The posterior probability that population i is best is estimated by

$$\hat{p}_i = \frac{1}{M} \sum_{m=1}^M I\left(\gamma_i^{(m)} = \max_{1 \leq j \leq k} \gamma_j^{(m)}\right). \quad (4.6)$$

Here, $I(\cdot)$ denotes the indicator function. Therefore, $\hat{p}_i$ represents the proportion of retained posterior draws in which population i has the largest threshold parameter.

4.3. Posterior Probability of Correct Selection

Let S denote a selected subset of populations. Correct selection occurs when the selected subset contains the best population:

$$i^* \in S. \quad (4.7)$$

Under the Bayesian model, the posterior probability of correct selection is

$$\Pr(CS | \mathbf{t}, \mathbf{d}) = \Pr(i^* \in S | \mathbf{t}, \mathbf{d}). \quad (4.8)$$

Since p_i is the posterior probability that population i is best, this probability can be written as

$$\Pr(CS | \mathbf{t}, \mathbf{d}) = \sum_{i \in S} p_i. \quad (4.9)$$

This posterior probability differs from the frequentist probability of correct selection. The frequentist probability is a pre-data measure evaluated over repeated samples with fixed parameter values. In contrast, Equation (4.9) is conditional on the observed Alloy data and the specified prior model.

4.4. Bayesian Selected Subset

Let P^* denote the desired posterior probability level. To construct the selected subset, the estimated posterior probabilities of being best are ordered from largest to smallest:

$$\hat{p}_{(1)} \geq \hat{p}_{(2)} \geq \dots \geq \hat{p}_{(k)}, \quad (4.10)$$

where $\hat{p}_{(r)}$ denotes the r th largest estimated posterior probability and $i_{(r)}$ de-

notes the population index associated with that probability.

The Bayesian selected subset is

$$S_B = \{i_{(1)}, \dots, i_{(m)}\}, \quad (4.11)$$

where m is the smallest integer satisfying

$$\sum_{r=1}^m \hat{p}_{(r)} \geq P^*. \quad (4.12)$$

Thus, S_B is the smallest subset, determined by the ordering of the estimated posterior probabilities, whose estimated posterior probability of containing the best population meets or exceeds P^* .

If a single population has a dominant posterior probability of being best, the selected subset may consist solely of that population. When posterior uncertainty is higher, additional populations are included until the desired posterior probability threshold is achieved. In the Alloy study, the primary analysis sets $P^* = 0.95$.

4.5. Percentile-Based Reliability Comparisons

Although the formal definition of the best population and the Bayesian selected subset are based on the threshold parameter γ_i , the same posterior comparison can be applied to Weibull lifetime percentiles. For a three-parameter Weibull distribution, the p th percentile for population i is

$$t_{p,i} = \gamma_i + \eta_i \left[-\log(1-p) \right]^{1/\beta_i}, \quad 0 < p < 1. \quad (4.13)$$

The 10th, 50th, and 90th percentiles are examined:

$$t_{0.10,i}, \quad t_{0.50,i}, \quad t_{0.90,i}.$$

The quantity $t_{p,i}$ indicates the time by which a proportion p of units from population i is expected to have failed according to the Weibull model.

For each retained posterior draw, the largest value of $t_{p,i}$ among the populations is identified. The posterior probability that population i has the largest p th percentile is estimated by

$$\hat{p}_{t_{p,i}} = \frac{1}{M} \sum_{m=1}^M I \left(t_{p,i}^{(m)} = \max_{1 \leq j \leq k} t_{p,j}^{(m)} \right). \quad (4.14)$$

This calculation is performed separately for $p = 0.10, 0.50$, and 0.90 .

These percentile-based comparisons do not define the formal Bayesian selected subset, which remains based on the threshold parameter γ_i . Rather, they serve as supplementary reliability summaries to evaluate whether populations with greater posterior support for the largest threshold parameter also tend to exhibit larger lifetime percentiles.

5. Posterior Computation

Posterior inference is conducted independently for each of the five Alloy populations using a random-walk Metropolis algorithm [28] [29]. Sampling is performed

on the transformed parameter scale.

$$\boldsymbol{\theta}_i = (\theta_{i1}, \theta_{i2}, \theta_{i3})^T = (\log(\beta_i), \log(\eta_i), \log(\delta_i))^T,$$

This transformation ensures that the positivity constraints on β_i , η_i , and δ_i , as well as the threshold constraint $\gamma_i < t_{i(1)}$, are automatically satisfied. After sampling, the retained draws are mapped back to the original Weibull parameter scale using

$$\beta_i = \exp(\theta_{i1}), \quad \eta_i = \exp(\theta_{i2}), \quad \delta_i = \exp(\theta_{i3}),$$

and

$$\gamma_i = t_{i(1)} - \exp(\theta_{i3}) - \varepsilon.$$

5.1. Random-Walk Metropolis Algorithm

Let

$$\ell_i(\boldsymbol{\theta}_i) = \log \pi(\boldsymbol{\theta}_i | \mathbf{t}_i, \mathbf{d}_i)$$

denote the log-posterior density for population i on the transformed scale. At iteration m , a candidate value is generated from

$$\boldsymbol{\theta}_i^* \sim N_3(\boldsymbol{\theta}_i^{(m-1)}, c^2 \hat{\boldsymbol{\Sigma}}_{\text{pilot},i}), \quad (5.1)$$

where c is a tuning constant and $\hat{\boldsymbol{\Sigma}}_{\text{pilot},i}$ is a covariance matrix estimated from a preliminary random-walk run.

Because the proposal distribution is symmetric, the proposal-density terms cancel in the Metropolis-Hastings ratio. The candidate is accepted with probability

$$\alpha_{\text{RW}} = \min \left\{ 1, \frac{\pi(\boldsymbol{\theta}_i^* | \mathbf{t}_i, \mathbf{d}_i)}{\pi(\boldsymbol{\theta}_i^{(m-1)} | \mathbf{t}_i, \mathbf{d}_i)} \right\}. \quad (5.2)$$

Equivalently, computation is performed on the log scale using

$$\log(\alpha_{\text{RW}}) = \min \{ 0, \ell_i(\boldsymbol{\theta}_i^*) - \ell_i(\boldsymbol{\theta}_i^{(m-1)}) \}. \quad (5.3)$$

If the proposed value is not accepted, the current value is retained for the next iteration.

5.2. Starting Values and Proposal Tuning

For each population, starting values are obtained by fitting shifted Weibull models across a grid of possible values for δ_i . These values are then used to initialize

$$(\log(\beta_i), \log(\eta_i), \log(\delta_i)).$$

A preliminary random-walk run of 8000 iterations is conducted for each chain. The empirical covariance matrix from this pilot run is used as $\hat{\boldsymbol{\Sigma}}_{\text{pilot},i}$, which accounts for the posterior scales and dependencies among the three transformed parameters. In the main random-walk proposal, the tuning constant is set to $c = 1.2$.

Four independent chains are run for each population. Each chain contains

200,000 iterations, of which the first 20,000 are discarded as burn-in. Every third post-burn-in draw is retained, yielding 60,000 retained draws per chain. The retained draws from the four chains are pooled for posterior estimation and subset selection.

Because the five populations are modeled independently and independent priors are assigned across populations, the joint posterior distribution factors into the product of the five population-specific posterior distributions. Joint posterior draws used in the subset-selection calculations are therefore formed by combining the corresponding retained draws from the separately fitted population models.

5.3. Convergence Assessment

Convergence and sampling performance for all five populations are assessed using traceplots, autocorrelation functions, acceptance rates, rank-normalized split and folded-split $\hat{R}$ diagnostics, and bulk and tail effective sample sizes. Traceplots and autocorrelation functions are examined separately within each chain for the transformed parameters

$$\log(\beta_i), \log(\eta_i), \log(\delta_i).$$

The $\hat{R}$, bulk ESS, and tail ESS diagnostics are calculated jointly across the four chains while preserving chain identity. These diagnostics are evaluated for the Weibull parameters γ_i , η_i , β_i , and δ_i , and for the selected lifetime percentiles $t_{0.10,i}$, $t_{0.50,i}$, and $t_{0.90,i}$. After convergence has been assessed, the retained draws from the four chains are combined for posterior summaries and subset-selection probabilities. The resulting numerical summaries and representative graphical displays are presented alongside the Alloy study results in Section 6.

6. Alloy T7987 Fatigue-Life Study

The proposed Bayesian subset-selection procedure is illustrated using a semi-simulated five-population study based on the Alloy T7987 fatigue-life data. The original data were reported by Meeker, Escobar, and Pascual and are included in the reliability-data collection accompanying *Statistical Methods for Reliability Data*, second edition [7]. The dataset contains 72 specimens, of which 67 failed during testing and five were right-censored at 300 thousand cycles. Time is measured in thousands of cycles. The original Alloy data provide a realistic censored-data baseline, while the additional populations allow the subset-selection procedure to be examined when several reliability populations have similar lifetime distributions.

6.1. Original Alloy Data and Weibull Fit

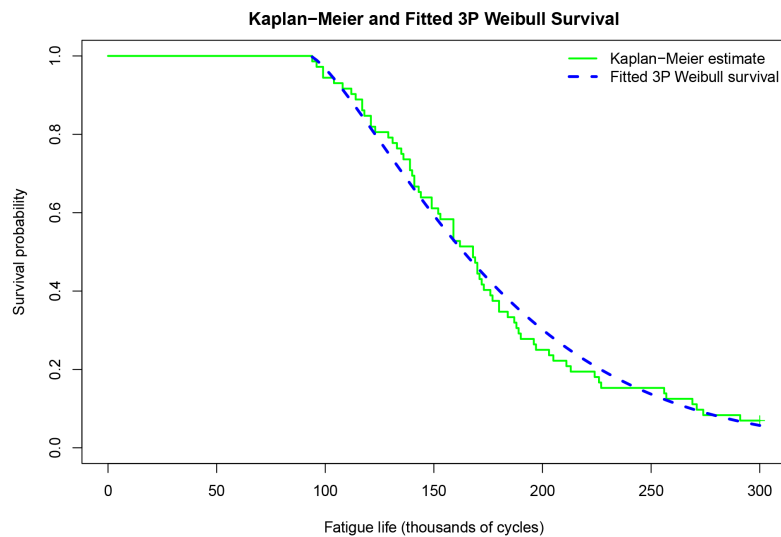
Before constructing the five-population study, a three-parameter Weibull distribution was fit to the original Alloy data using the right-censored likelihood. The maximum likelihood estimates were

$$\hat{\gamma} = 92.9928, \quad \hat{\eta} = 93.2641, \quad \hat{\beta} = 1.3202.$$

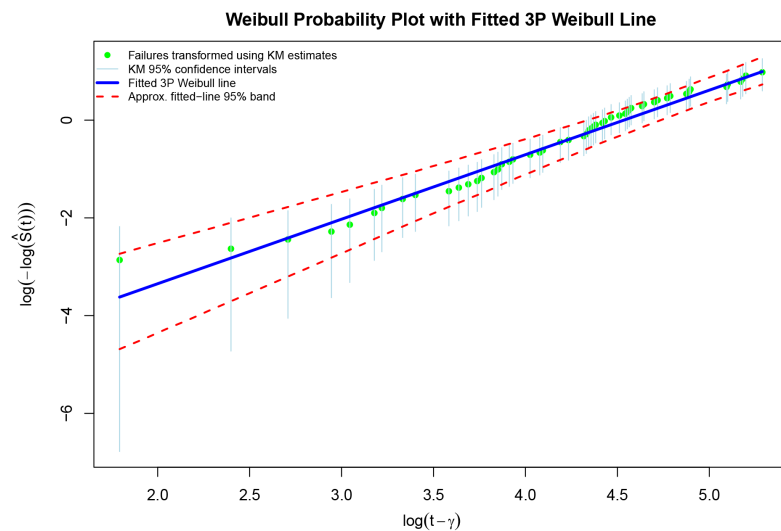
The estimated shape parameter is greater than one, corresponding to an in-

creasing fitted hazard rate. This behavior is consistent with a fatigue process in which the instantaneous risk of failure increases as the number of cycles accumulates.

Figure 1 provides two complementary graphical assessments of the fitted model. The fitted Weibull survival curve follows the Kaplan-Meier estimate closely over most of the observed range and remains positive near 300 thousand cycles, as expected from the censored specimens. On the Weibull probability scale, the transformed Kaplan-Meier points are approximately linear, with most points lying close to the fitted line or within its approximate pointwise confidence band. These results support the three-parameter Weibull distribution as a reasonable working model for the Alloy fatigue-life data.



(a) Kaplan-Meier estimate and fitted Weibull survival curve



(b) Weibull probability plot with fitted line and approximate confidence band

Figure 1. Graphical assessment of the three-parameter Weibull fit to the original Alloy T7987 fatigue-life data.

6.2. Construction of the Five Populations

A Bayesian three-parameter Weibull model was first fitted to the original Alloy T7987 fatigue-life data. The posterior mean estimates of γ , η , and β from this fit were adopted as the baseline generating values for Population P1. The remainder of the semi-simulated study was constructed relative to this Bayesian baseline. Populations P2-P5 were generated with progressively larger threshold parameters, while their scale and shape parameters were kept close to the posterior estimates obtained from the original Alloy data. Thus, the five populations retain broadly similar fatigue-life behavior, while their intended ordering is determined primarily by the threshold parameter γ .

Table 1 gives the parameter values used in the construction and the resulting censoring summaries. Each population contains 72 observations, with censoring percentages ranging from 6.94% to 12.50%. Since P5 has the largest generating threshold parameter, $\gamma_5 = 100.993$, it is the true best population under the formal threshold-based definition.

Table 1. Construction and censoring summary for the semi-simulated Alloy fatigue-life populations.

Population	γ	η	β	n	Failures	Censored	Percent censored
P1	92.993	92.7443	1.2737	72	67	5	6.94
P2	94.993	90.8894	1.2482	72	63	9	12.50
P3	96.993	92.7443	1.2737	72	67	5	6.94
P4	98.993	94.5992	1.2992	72	65	7	9.72
P5	100.993	96.4541	1.3246	72	67	5	6.94

6.3. Posterior Estimates and Threshold-Based Selection

The censored three-parameter Weibull model was fit separately to each population using the random-walk Metropolis algorithm described in Section 5. **Table 2** reports the posterior means and 95% credible intervals for the principal Weibull parameters. The posterior mean of γ_i is largest for P5, followed by P4 and P3. However, the credible intervals for P3-P5 overlap substantially, reflecting the similarity of the populations and the uncertainty introduced by right-censoring. The posterior intervals for η_i and β_i also overlap, as expected from the construction of the study.

Table 2. Random-walk posterior means and 95% credible intervals for the Weibull parameters.

Population	γ : mean and 95% CrI	η : mean and 95% CrI	β : mean and 95% CrI
P1	92.6351 [87.4915, 93.9723]	94.4843 [77.1021, 114.5755]	1.3145 [1.0572, 1.6130]
P2	99.2982 [95.3508, 100.2460]	102.5622 [81.2505, 127.6475]	1.1512 [0.9149, 1.4246]
P3	100.3579 [94.9090, 101.7160]	98.7007 [80.0642, 120.2761]	1.2832 [1.0274, 1.5845]
P4	100.3876 [96.7317, 101.2585]	102.8857 [81.3138, 128.4027]	1.1231 [0.8970, 1.3835]
P5	101.1260 [95.7891, 102.5059]	97.1267 [78.9459, 117.9522]	1.3025 [1.0450, 1.6035]

Posterior means are shown in black; 95% credible intervals are shown in brown.

Figure 2 displays the random-walk posterior densities of the five threshold parameters on a common horizontal scale. The densities shift toward larger values from P1 to the upper-threshold populations, but substantial overlap remains among P3, P4, and P5. This overlap explains why the posterior evidence is distributed across several possible best populations rather than being concentrated entirely on P5.

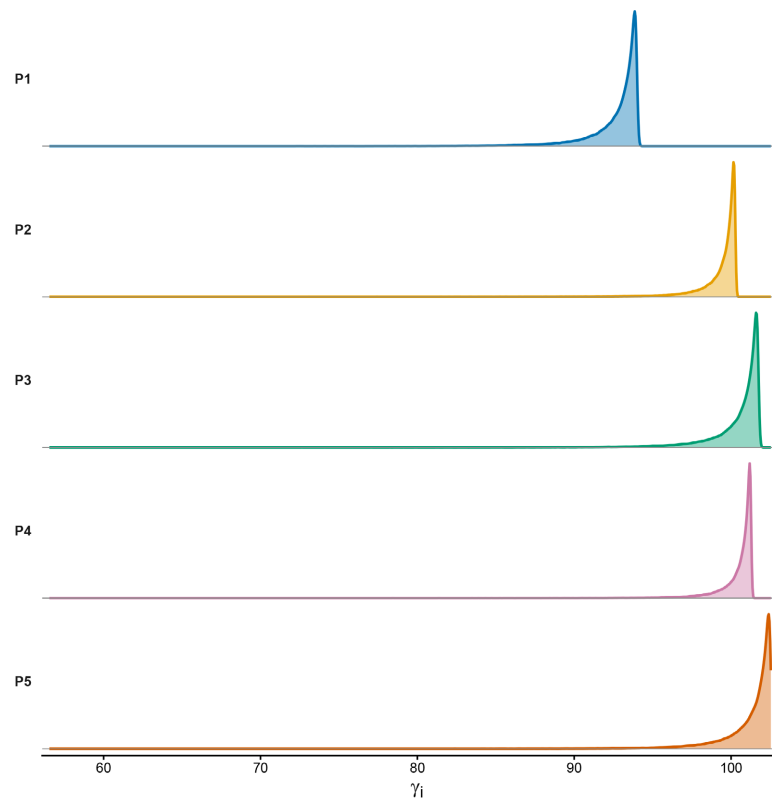


Figure 2. Random-walk posterior densities of the threshold parameters for the five semi-simulated Alloy populations.

The ranked posterior probabilities of being best are shown in **Table 3**. P5 has the largest posterior probability, 0.6702, followed by P3 with probability 0.2178 and P4 with probability 0.1062. P2 has only a small probability of being best, and the estimated probability for P1 is zero to four decimal places.

Table 3. Random-walk posterior probabilities of being best, cumulative probabilities, and inclusion in the selected subset for $P^* = 0.95$.

Rank	Population	$\hat{p}_i$	Cumulative	Included
1	P5	0.6702	0.6702	Yes
2	P3	0.2178	0.8880	Yes
3	P4	0.1062	0.9942	Yes
4	P2	0.0058	1.0000	No
5	P1	0.0000	1.0000	No

The Monte Carlo uncertainty in the estimated posterior probabilities was assessed using a binomial approximation. Let X_i denote the number of retained draws within a chain for which population i has the largest threshold parameter. Then

$$X_i \sim \text{Binomial}(n, p_i), \quad \hat{p}_i = \frac{X_i}{n},$$

where p_i is the posterior probability that population i has the largest threshold. Therefore,

$$\text{SD}(\hat{p}_i) = \sqrt{\frac{p_i(1-p_i)}{n}} \leq \frac{1}{2\sqrt{n}}.$$

With $n = 60000$ retained draws per chain, the upper bound is approximately 0.00204. Thus, under the binomial approximation, the Monte Carlo uncertainty in the estimated posterior probabilities of being best is small. Serial dependence in the MCMC draws is assessed separately through the chain-aware effective sample size diagnostics.

P5 alone does not attain the target posterior probability $P^* = 0.95$. Adding P3 raises the cumulative probability to 0.8880, which remains below the target. After P4 is added, the cumulative probability becomes 0.9942. Therefore, the Bayesian selected subset is

$$\mathcal{S}_B = \{P5, P3, P4\}.$$

The selected subset contains the true best population while acknowledging that the observed censored data do not separate P5 sufficiently from P3 and P4 to justify selecting P5 alone.

6.4. Percentile-Based Reliability Assessments

The posterior Weibull percentiles provide a complementary comparison because they depend jointly on γ_i , η_i , and β_i . **Table 4** reports the posterior mean of each percentile and the posterior probability that the corresponding population has the largest percentile.

Table 4. Random-walk posterior means and probabilities of having the largest Weibull percentile.

Population	$t_{0.10}$		$t_{0.50}$		$t_{0.90}$	
	Mean	$P(\text{largest})$	Mean	$P(\text{largest})$	Mean	$P(\text{largest})$
P1	109.7321	0.0100	163.9910	0.0268	272.2456	0.0210
P2	113.8767	0.0827	173.6953	0.2339	313.3131	0.3612
P3	117.4985	0.3431	174.3818	0.2428	291.0458	0.0926
P4	114.3337	0.0971	174.4447	0.2636	318.8830	0.4603
P5	118.4326	0.4670	174.2870	0.2329	286.8809	0.0650

For the early-life percentile $t_{0.10}$, P5 has the largest posterior mean and the largest probability of being best, although that probability is only 0.4670. The pos-

terior means of the median $t_{0.50}$ are nearly equal for P2-P5, and the probability of having the largest median is distributed across these populations. For $t_{0.90}$, P4 and P2 receive the largest posterior probabilities. Thus, the upper-tail ranking differs from the threshold-based ranking because upper percentiles are particularly sensitive to uncertainty in the scale and shape parameters.

The posterior probabilities that the threshold-based and percentile-based criteria identify the same population are 0.3804, 0.2095, and 0.1281 for $t_{0.10}$, $t_{0.50}$, and $t_{0.90}$, respectively. The decline in agreement illustrates that the formal threshold criterion and the broader percentile comparisons answer related but distinct reliability questions. The selected subset in this paper remains defined by the largest threshold parameter.

6.5. MCMC Diagnostics

The convergence and sampling diagnostics described in Section 5.3 were evaluated for all five populations. **Figure 3** shows the random-walk traceplots for P5 as a representative case. The four chains remain stable and explore common posterior regions without visible long-term drift. The $\log(\delta_5)$ chains exhibit greater variability than the $\log(\beta_5)$ and $\log(\eta_5)$ chains because the threshold-gap parameter is directly related to the lower-endpoint constraint.

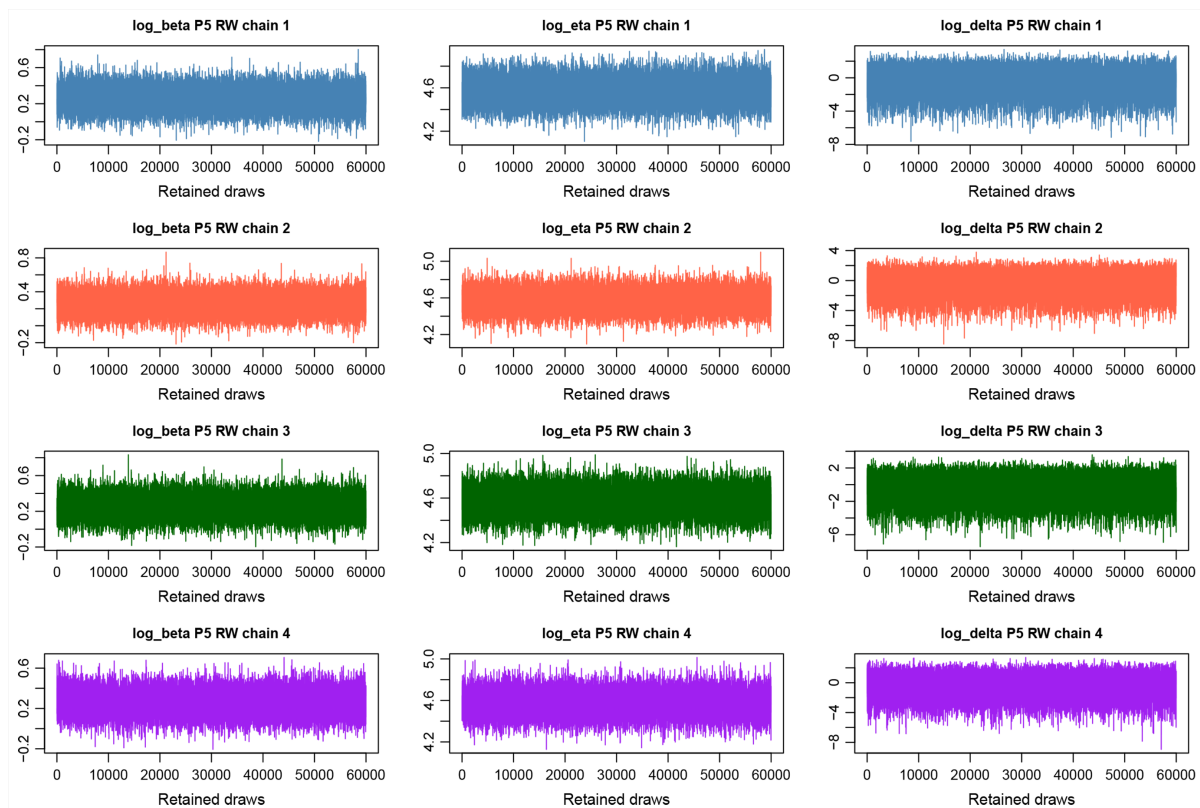


Figure 3. Random Walk traceplots for Population P5. The columns correspond to $\log(\beta_5)$, $\log(\eta_5)$, and $\log(\delta_5)$, and the rows correspond to the four chains. The horizontal axis indexes the 60,000 retained post-burn-in draws from each chain after retaining every third draw.

Table 5 provides a compact summary of the random-walk diagnostics. The rank-normalized split/folded-split $\hat{R}$, bulk effective sample size, and tail effective sample size were calculated jointly across the four chains while preserving chain identity (Vehtari *et al.* [40]). Bulk ESS assesses sampling efficiency in the central region of the posterior distribution, whereas tail ESS assesses sampling efficiency in the posterior tails.

The mean random-walk acceptance rates range from 0.356 to 0.367. All maximum $\hat{R}$ values are at most 1.0001. Across the model parameters γ , η , β , and δ , the minimum bulk and tail ESS values are 57515.1 and 74180.7, respectively. Across the Weibull percentiles $t_{0.10}$, $t_{0.50}$, and $t_{0.90}$, the corresponding minimum values are 60267.6 and 79309.0. Autocorrelation functions were also examined separately within each chain for every population and decreased with increasing lag; they are not displayed to avoid repetitive graphical material.

Table 5. Summary of chain-aware random-walk MCMC diagnostics.

Population	Mean acceptance	Maximum $\hat{R}$	Minimum parameter bulk ESS	Minimum parameter tail ESS	Minimum percentile bulk ESS	Minimum percentile tail ESS
P1	0.364	1.0001	57515.1	74901.4	61876.6	81398.3
P2	0.367	1.0001	58130.5	74180.7	62194.6	80512.8
P3	0.357	1.0000	58281.4	77910.1	61040.0	79655.2
P4	0.367	1.0001	60755.9	77662.6	62627.0	79309.0
P5	0.356	1.0000	58868.8	77531.5	60267.6	79861.1

Taken together, the trace plots, autocorrelation functions, acceptance rates, rank-normalized $\hat{R}$ values, and bulk and tail effective sample sizes indicate satisfactory convergence and sampling efficiency for the retained random-walk posterior draws.

6.6. Prior Sensitivity and Posterior Predictive Checks

Sensitivity to the prior on the threshold-gap parameter was examined by varying the standard deviation of the Normal prior for $\theta_{i3} = \log(\delta_i)$ over 1, 2, and 3. **Figure 4** shows that the posterior means of γ_i remain relatively stable across the three specifications. The credible intervals widen, particularly through their lower limits, as the prior becomes more diffuse. P5 retains the largest posterior mean threshold under all three prior settings, indicating that the main threshold-based conclusion is not driven by the selected prior standard deviation.

A similar sensitivity analysis was done by varying the location parameter of the Normal prior distribution for θ_{i3} , $\log(\delta_i)$, over $\log(0.5)$, $\log(1)$, and $\log(2)$, while the prior standard deviation was held fixed at 2. As before the posterior means of γ_i remained relatively stable. The corresponding 95% credible intervals widened as the prior median of δ_i increased, primarily through decreases in their lower limits. The upper credible limits remained similar, and the

intervals overlapped substantially across the prior specifications. Although the posterior probabilities of being best also varied numerically, the population ordering remained $P5 > P3 > P4 > P2 > P1$, and the selected subset at $P^* = 0.95$ remained $\{P5, P3, P4\}$. Thus, the main subset-selection conclusion was not sensitive to the prior location values considered.

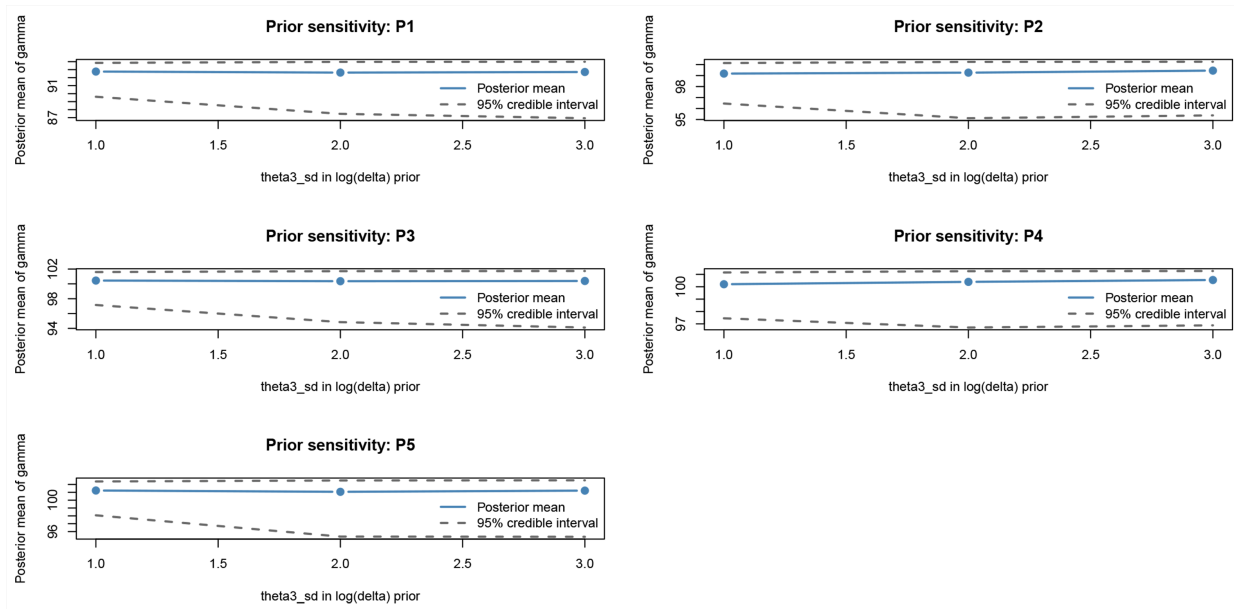


Figure 4. Prior sensitivity of the posterior threshold estimates. Each panel shows the posterior mean and 95% credible interval for γ_i as the prior standard deviation of $\log(\delta_i)$ varies.

Posterior predictive checks were performed for all five fitted populations. Replicated lifetime datasets were generated from the posterior predictive distribution and subjected to the same right-censoring mechanism as the observed datasets. The checks included the minimum, mean, standard deviation, numbers of failures and censored observations, median, and lower- and upper-tail quantiles.

Figure 5 shows representative checks for the minimum, mean, and standard deviation of P5. In each panel, the vertical line marks the observed statistic. The observed values lie within the main body of the corresponding posterior predictive distributions. Across all five populations and all examined statistics, the posterior predictive p -values ranged from 0.267 to 0.797, with no values near zero or one. The fitted censored Weibull models therefore reproduce the principal features of the observed datasets, including their lower endpoints, centers, spreads, tails, and censoring patterns.

Overall, the semi-simulated Alloy study demonstrates the role of Bayesian subset selection in a censored reliability setting. P5 receives the largest posterior probability of being best, but the evidence is not sufficiently concentrated to select P5 alone at $P^* = 0.95$. The resulting subset $\{P5, P3, P4\}$ preserves the target posterior probability while representing the uncertainty among the leading populations. The calculations in this section were carried out using the R-code in **Appendix A**.

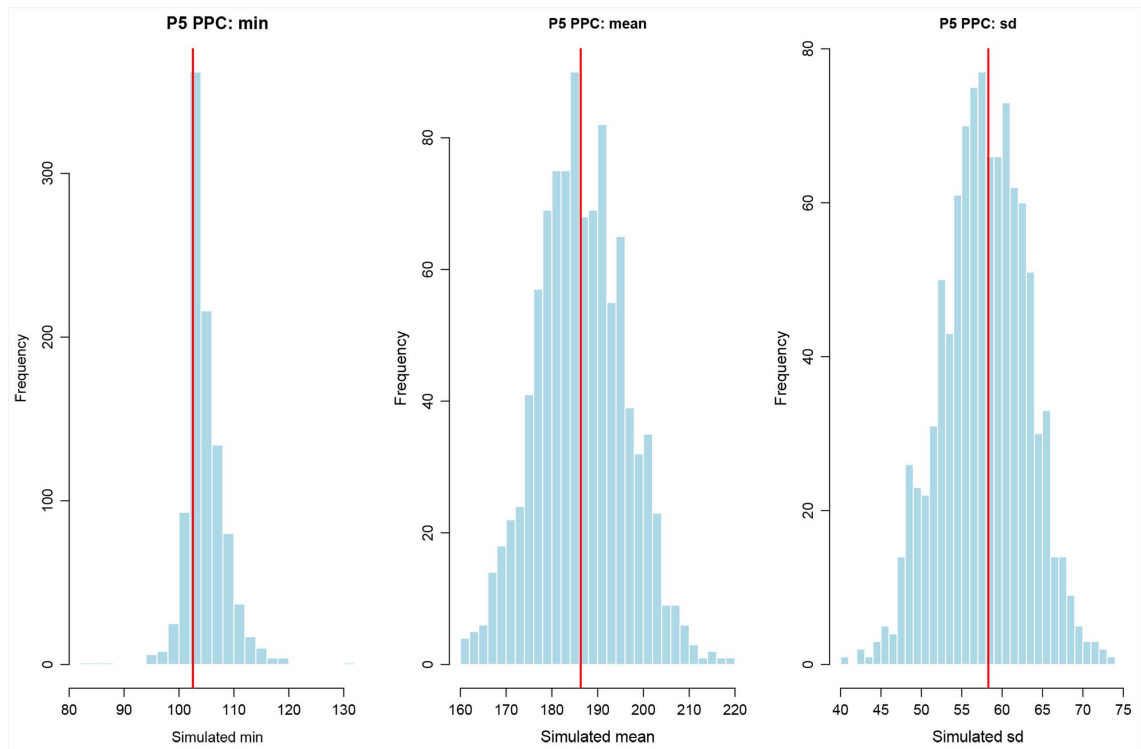


Figure 5. Representative posterior predictive checks for Population P5. The histograms show replicated values of the minimum, mean, and standard deviation; the vertical lines mark the corresponding observed values.

7. Discussion

The semi-simulated Alloy T7987 study illustrates how Bayesian subset selection can be used when several reliability populations have similar lifetime distributions and the available data are subject to right-censoring. The posterior probability that Population P5 has the largest threshold parameter is 0.6702, which is higher than the corresponding probability for any other population but is not sufficient to select P5 alone when the target posterior content is $P^* = 0.95$. After P3 and P4 are added according to their posterior probabilities of being best, the cumulative probability reaches 0.9942. The resulting selected subset is therefore

$$\mathcal{S}_B = \{P5, P3, P4\}.$$

This result reflects the substantial overlap among the posterior distributions of the leading threshold parameters shown in **Figure 2**. The inclusion of multiple populations should not be interpreted as a failure to identify a useful result. Rather, it provides a direct representation of the uncertainty supported by the observed censored data and avoids overstating the evidence in favor of a single population.

The threshold-based and percentile-based comparisons also provide different perspectives on reliability performance. The formal selection rule defines the best population as the one with the largest threshold parameter, whereas a Weibull percentile depends jointly on γ_i , η_i , and β_i . Consequently, a population with the largest lower lifetime bound need not have the largest median or upper-tail

percentile. In the Alloy study, P5 has the largest posterior mean and posterior probability of being best for the early-life percentile $t_{0.10}$, but the ranking becomes less concentrated for $t_{0.50}$ and differs more substantially for $t_{0.90}$, as shown in **Table 4**. These differences emphasize that threshold-based and percentile-based comparisons answer related but distinct reliability questions. The threshold criterion, which defines “best” throughout this paper, is appropriate when the primary objective is to compare lower lifetime bounds, while percentile comparisons may be more informative when performance over the broader lifetime distribution is of interest.

A principal advantage of the Bayesian approach is that it produces directly interpretable posterior probabilities for each population being best. These probabilities can be used to construct a selected subset with a prespecified posterior content without forcing the analysis to identify a single winner when the evidence is uncertain. The same posterior simulation also provides parameter estimates, credible intervals, reliability percentiles, ranking probabilities, and uncertainty summaries within a unified framework. In addition, the right-censored Weibull likelihood permits both exact failure times and censored observations to contribute appropriately to the analysis.

The random-walk Metropolis implementation performed well in this study. Acceptance rates were similar across populations, the Gelman-Rubin diagnostics were essentially one, and the effective sample sizes were large, as summarized in **Table 5**. The traceplots showed stable exploration of the posterior distributions, and the posterior predictive checks indicated that the fitted models were able to reproduce the main characteristics of the observed datasets. The prior sensitivity analyses showed that the posterior threshold summaries changed somewhat across the examined prior standard deviations and prior locations for $\log(\delta_i)$, but the main population ranking and selected subset remained unchanged. Together, these results support the numerical reliability of the posterior summaries used for the subset-selection decision.

Several limitations should nevertheless be considered. First, only P1 is directly tied to the original Alloy T7987 data; Populations P2-P5 were generated relative to the Bayesian fit of that dataset. The application therefore provides a realistic semi-simulated illustration rather than a comparison of five independently observed physical populations. Second, the analysis assumes that each population follows a three-parameter Weibull distribution. Although the graphical assessment and posterior predictive checks indicate that the three-parameter Weibull model fits the present data reasonably well, the results could be affected if this distribution is not appropriate for data from other applications. Third, the population-specific models were fitted independently, so the procedure does not borrow information across populations even when their lifetime distributions are similar. Finally, the definition of the best population is tied to the largest threshold parameter and may not align with objectives based on a median, an upper percentile, expected lifetime, or another reliability criterion.

Future work could extend the procedure in several directions. Hierarchical formulations could permit partial pooling among related populations and may improve estimation when sample sizes are small. Alternative lifetime distributions, interval censoring, progressive censoring, and competing-risk settings could also be considered. Another extension would allow the decision criterion to depend jointly on the threshold and one or more reliability percentiles. A comprehensive repeated-sampling study could evaluate the procedure's operating characteristics across different threshold separations, censoring levels, and sample sizes. Outcomes could include the probability of correct selection, the expected selected subset size, the probability of selecting the true best population alone, and population-specific inclusion probabilities. Computational work could investigate more efficient posterior samplers for applications involving a larger number of populations or more complex models. Most importantly, applying the procedure to studies containing multiple independently observed reliability populations would provide additional evidence about its practical performance.

8. Conclusions

This paper developed a Bayesian subset-selection procedure for comparing multiple populations that follow three-parameter Weibull distributions. The best population was defined as the one with the largest threshold parameter, and posterior simulation was used to estimate the probability that each population is best. Ordering these probabilities and accumulating them until a prescribed posterior content is reached yields a selected subset that explicitly accounts for uncertainty in the population ranking.

The method was illustrated using a semi-simulated censored study based on the Alloy T7987 fatigue-life data. A Bayesian three-parameter Weibull fit to the original data supplied the baseline parameter estimates used to construct the five-population study. Under the threshold-based criterion, P5 received the largest posterior probability of being best, but the evidence was not sufficiently concentrated to select P5 alone at $P^* = 0.95$. The selected subset {P5, P3, P4} retained the target posterior probability while representing the uncertainty among the leading populations.

The study also showed that rankings based on the Weibull threshold need not coincide with rankings based on reliability percentiles, since the latter depend on the threshold, scale, and shape parameters jointly. Overall, the proposed procedure provides a probabilistically interpretable method for comparing multiple reliability populations in the presence of right-censored lifetime data. Further development for hierarchical models, alternative lifetime distributions, unequal sample sizes, and additional censoring structures would broaden its use in practical reliability studies.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Bechhofer, R.E. (1954) A Single-Sample Multiple Decision Procedure for Ranking Means of Normal Populations with Known Variances. *The Annals of Mathematical Statistics*, **25**, 16-39. <https://doi.org/10.1214/aoms/1177728845>
- [2] Gupta, S.S. (1965) On Some Multiple Decision (Selection and Ranking) Rules. *Technometrics*, **7**, 225-245. <https://doi.org/10.1080/00401706.1965.10490251>
- [3] Gupta, S.S. and Panchapakesan, S. (1979) Multiple Decision Procedures. John Wiley & Sons, Inc.
- [4] Bechhofer, R.E., Santner, T.J. and Goldsman, D.M. (1995) Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons. John Wiley & Sons, Inc.
- [5] Gibbons, J.D., Olkin, I. and Sobel, M. (1999) Selecting and Ordering Populations: A New Statistical Methodology. Society for Industrial and Applied Mathematics. <https://doi.org/10.1137/1.9781611971101>
- [6] Lawless, J.F. (2002) Statistical Models and Methods for Lifetime Data. Wiley. <https://doi.org/10.1002/9781118033005>
- [7] Meeker, W.Q., Escobar, L.A. and Pascual, F.G. (2022) Statistical Methods for Reliability Data. 2 Edition, John Wiley & Sons, Inc.
- [8] Nelson, W. (2004) Applied Life Data Analysis. John Wiley & Sons, Inc.
- [9] Singh, H. and Gill, A.N. (1995) Ranking and Subset Selection Procedures for Populations with Censored Data: A Review. In: Koul, H.L. and Deshpande, J.V., Eds., *Institute of Mathematical Statistics Lecture Notes-Monograph Series*, Institute of Mathematical Statistics, 219-230. <https://doi.org/10.1214/lnms/1215452222>
- [10] Rinne, H. (2009) The Weibull Distribution. Chapman & Hall/CRC.
- [11] Jiang, R. and Murthy, D.N.P. (2011) A Study of Weibull Shape Parameter: Properties and Significance. *Reliability Engineering & System Safety*, **96**, 1619-1626. <https://doi.org/10.1016/j.res.2011.09.003>
- [12] Cohen, A.C. (1965) Maximum Likelihood Estimation in the Weibull Distribution Based on Complete and on Censored Samples. *Technometrics*, **7**, 579-588. <https://doi.org/10.1080/00401706.1965.10490300>
- [13] Barbero, E., Fernández-Sáez, J. and Navarro, C. (1999) On the Estimation of Percentiles of the Weibull Distribution. *Journal of Materials Science Letters*, **18**, 1441-1443. <https://doi.org/10.1023/a:1006635810580>
- [14] Evans, J., Kretschmann, D. and Green, D. (2019) Procedures for Estimation of Weibull Parameters.
- [15] Kingston, J.V. and Patel, J.K. (1980) Selecting the Best One of Several Weibull Populations. *Communications in Statistics-Theory and Methods*, **9**, 383-398. <https://doi.org/10.1080/03610928008827887>
- [16] Kingston, J.V. and Patel, J.K. (1980) A Restricted Subset Selection Procedure for Weibull Populations. *Communications in Statistics-Theory and Methods*, **9**, 1371-1383. <https://doi.org/10.1080/03610928008827966>
- [17] An Hsu, T. (1982) On Some Optimal Selection Procedures for Weibull Populations. *Communications in Statistics-Theory and Methods*, **11**, 2657-2668. <https://doi.org/10.1080/03610928208828413>
- [18] Gupta, S.S. and Miescke, K.J. (1986) Optimum Two-Stage Selection Procedures for Weibull Populations. *Journal of Statistical Planning and Inference*, **15**, 147-156. [https://doi.org/10.1016/0378-3758\(86\)90093-5](https://doi.org/10.1016/0378-3758(86)90093-5)

- [19] Sheng-Tsaing Tseng, and Wu, H.-J. (1990) Selecting, under Type-II Censoring, Weibull Populations That Are More Reliable. *IEEE Transactions on Reliability*, **39**, 193-198. <https://doi.org/10.1109/24.55881>
- [20] Ng, C.K. (2013) Procedures for Selecting Good Exponential Populations. *Communications in Statistics-Simulation and Computation*, **42**, 1681-1692. <https://doi.org/10.1080/03610918.2012.674598>
- [21] McDonald, G.C. and Hodaj, J. (2025) Selection Rules for Exponential Population Threshold Parameters. *Applied Mathematics*, **16**, 1-14. <https://doi.org/10.4236/am.2025.161001>
- [22] McDonald, G.C. and Hodaj, J. (2025) Operating Characteristics of Subset Selection Rules for Exponential Population Threshold Parameters. *Applied Mathematics*, **16**, 441-460. <https://doi.org/10.4236/am.2025.165024>
- [23] McDonald, G.C. and Hodaj, J. (2025) A Note Comparing Two Subset Selection Procedures for the Threshold Parameters of Two Exponential Populations. *Applied Mathematics*, **16**, 877-883. <https://doi.org/10.4236/am.2025.1612046>
- [24] McDonald, G.C. (2016) Applications of Subset Selection Procedures and Bayesian Ranking Methods in Analysis of Traffic Fatality Data. *WIREs Computational Statistics*, **8**, 222-237. <https://doi.org/10.1002/wics.1385>
- [25] Kondo, Y., Zidek, J.V., Taylor, C.G. and van Eeden, C. (2018) Bayesian Subset Selection Methods for Finding Engineering Design Values: An Application to Lumber Strength. *Sankhya A*, **80**, 146-172. <https://doi.org/10.1007/s13171-018-00157-w>
- [26] Gupta, S.S. and He, S. (1997) Selecting the Best Weibull Population Based on Type-I Censored Data: A Bayesian Approach. Technical Report TR97-03C, Department of Statistics, Purdue University.
- [27] Kondo, Y. and Zidek, J.V. (2013) Bayesian Nonparametric Subset Selection Procedures with Weibull Components. Technical Report Technical Report 273, Department of Statistics, The University of British Columbia.
- [28] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. (1953) Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, **21**, 1087-1092. <https://doi.org/10.1063/1.1699114>
- [29] Hastings, W.K. (1970) Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, **57**, 97-109. <https://doi.org/10.1093/biomet/57.1.97>
- [30] Chib, S. and Greenberg, E. (1995) Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, **49**, 327-335. <https://doi.org/10.1080/00031305.1995.10476177>
- [31] Gelman, A. and Rubin, D.B. (1992) Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, **7**, 457-472. <https://doi.org/10.1214/ss/1177011136>
- [32] Brooks, S.P. and Gelman, A. (1998) General Methods for Monitoring Convergence of Iterative Simulations. *Journal of Computational and Graphical Statistics*, **7**, 434-455. <https://doi.org/10.1080/10618600.1998.10474787>
- [33] Gelman, A., Meng, X.L. and Stern, H. (1996) Posterior Predictive Assessment of Model Fitness via Realized Discrepancies. *Statistica Sinica*, **6**, 733-760.
- [34] Christensen, R., Johnson, W., Branscum, A. and Hanson, T.E. (2010) Bayesian Ideas and Data Analysis. CRC Press.
- [35] Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A. and Rubin, D.B. (2013) Bayesian Data Analysis. 3rd Edition, Chapman and Hall/CRC.
- [36] Martz, H.F. and Waller, R.A. (1982) Bayesian Reliability Analysis. John Wiley & Sons,

Inc.

- [37] Hamada, M.S., Wilson, A.G., Reese, S.C. and Martz, H.F. (2008) Bayesian Reliability. Springer.
- [38] Kass, R.E. and Wasserman, L. (1996) The Selection of Prior Distributions by Formal Rules. *Journal of the American Statistical Association*, **91**, 1343-1370.
<https://doi.org/10.1080/01621459.1996.10477003>
- [39] Sun, D. (1997) A Note on Noninformative Priors for Weibull Distributions. *Journal of Statistical Planning and Inference*, **61**, 319-338.
[https://doi.org/10.1016/s0378-3758\(96\)00155-3](https://doi.org/10.1016/s0378-3758(96)00155-3)
- [40] Vehtari, A., Gelman, A., Simpson, D., Carpenter, B. and Bürkner, P. (2021) Rank-normalization, Folding, and Localization: An Improved $\hat{R}$ for Assessing Convergence of MCMC (with Discussion). *Bayesian Analysis*, **16**, 667-718.
<https://doi.org/10.1214/20-ba1221>

Appendix A

```
# =====
# Bayesian 3-Parameter Weibull Model
# Semi-Simulated Alloy T7987 Five-Population Censored Example
# Random-Walk Metropolis
# Exact reproduction of the dissertation random-walk draws
# Updated for reviewer Comments 4 and 6:
#   - prior-location sensitivity for log(delta)
#   - rank-normalized split/folded-split R-hat, bulk ESS, and tail
#     ESS
# P1 = original Alloy T7987 fitted baseline data
# P2-P5 = generated censored populations similar to Alloy T7987
# =====
rm(list = ls())
library(LearnBayes)
library(MASS)
library(dplyr)
library(ggplot2)
library(posterior)
set.seed(2026)
# -----
# Settings
# -----
PAD <- 1e-6
n_chains <- 4
iters_long <- 200000
burn_in <- 20000
thin <- 3
rw_scale_main <- 1.2
# -----
# Reproducibility settings
# -----
# The dissertation code ran an independence sampler immediately
# after
# each random-walk chain. That sampler consumed random numbers
# before
# the next random-walk chain began. To reproduce the dissertation's
# random-walk draws exactly under set.seed(2026), the same
# computation
# is retained internally below, but its draws and summaries are
# discarded.
# Only random-walk results are stored, summarized, and printed.
indep_inflate <- 4
mixture_df <- 5
mixture_p <- 0.20
# -----
# 1. Read semi-simulated five-population censored data
# -----
five_pop_data <- read.csv("AlloyT7987_OriginalPlus4Generated_
  Censored_5Pop.csv")
five_pop_data$Population <-
as.character(five_pop_data$Population)
pop_names <- sort(unique(five_pop_data$Population))
K <- length(pop_names)
# Split into a list by population.
datasets <- split(
  five_pop_data[, c("time", "status")],
  five_pop_data$Population)
# Censoring summary.
censoring_summary <- five_pop_data %>%
  group_by(Population) %>%
  summarise(
```

```

n = n(),
Failures = sum(status == 1),
Censored = sum(status == 0),
Percent_Censored = round(mean(status == 0) * 100, 2),
Min_Time = min(time),
Max_Time = max(time),
.groups = "drop" )
# True/fitted parameter table.
true_params <- read.csv("AlloyT7987_5Pop_Parameters.csv")
# -----
# True/fitted Weibull percentiles based on the parameter table
# -----
# Standardize column names in case the CSV uses lowercase names.
names(true_params) <- gsub(
  "gamma", "Gamma", names(true_params), ignore.case = TRUE)
names(true_params) <- gsub(
  "eta", "Eta", names(true_params), ignore.case = TRUE)
names(true_params) <- gsub(
  "beta", "Beta", names(true_params), ignore.case = TRUE)
true_percentiles <- true_params %>%
  mutate(
    t10 = Gamma + Eta * ((-log(1 - 0.10))^(1 / Beta)),
    t50 = Gamma + Eta * ((-log(1 - 0.50))^(1 / Beta)),
    t90 = Gamma + Eta * ((-log(1 - 0.90))^(1 / Beta))
  ) %>%
  dplyr::select(Population, Gamma, Eta, Beta, t10, t50, t90) %>%
  mutate(
    Gamma = round(Gamma, 4),
    Eta = round(Eta, 4),
    Beta = round(Beta, 4),
    t10 = round(t10, 4),
    t50 = round(t50, 4),
    t90 = round(t90, 4)
  )
# -----
# 2. Priors
# -----
# Gamma(shape, scale) parameterization is used in R.
# beta ~ Gamma(6, scale = 0.25), mean = 1.5
main_a_beta <- 6
main_b_beta <- 0.25
# eta ~ Gamma(0.01, scale = 10000)
main_a_eta <- 0.01
main_b_eta <- 10000
# Threshold gap:
# gamma = min(time) - delta - PAD
# theta3 = log(delta)
main_theta3_mean <- log(1)
main_theta3_sd <- 2.0
# Prior-location sensitivity grid for log(delta).
# These values correspond to prior medians for delta of 0.5, 1, and
# 2.
theta3_mean_grid <- log(c(0.5, 1, 2))
# Percentiles.
perc <- c(0.10, 0.50, 0.90)
z_p <- -log(1 - perc)
# Posterior predictive checks.
ppc_nrep <- 1000
# -----
# 3. Censored log-posterior for the 3P Weibull model
# -----
# theta = (log beta, log eta, log delta)
# For failures:

```

```

# contribution = log f(t)
# For censored observations:
# contribution = log S(t)
logpost_weibull3_cens <- function(
  theta,
  data,
  a_beta = main_a_beta,
  b_beta = main_b_beta,
  a_eta = main_a_eta,
  b_eta = main_b_eta,
  theta3_mean = main_theta3_mean,
  theta3_sd = main_theta3_sd,
  pad = PAD) {
  beta <- exp(theta[1])
  eta <- exp(theta[2])
  delta <- exp(theta[3])
  t1 <- min(data$time)
  gamma <- t1 - delta - pad
  t <- data$time
  d <- data$status
  y <- t - gamma
  if (!is.finite(beta) || !is.finite(eta) || !is.finite(delta)) {
    return(-Inf)
  }
  if (beta <= 0 || eta <= 0 || delta <= 0) {
    return(-Inf)
  }
  if (any(y <= 0)) {
    return(-Inf)
  }
  log_f <- log(beta) - beta * log(eta) +
    (beta - 1) * log(y) -
    (y / eta)^beta
  log_S <- -(y / eta)^beta
  loglik <- sum(d * log_f + (1 - d) * log_S)
  # Priors with Jacobian terms for beta and eta.
  lp_beta <- dgamma(
    beta,
    shape = a_beta,
    scale = b_beta,
    log = TRUE
  ) + theta[1]
  lp_eta <- dgamma(
    eta,
    shape = a_eta,
    scale = b_eta,
    log = TRUE
  ) + theta[2]
  # Prior directly on theta3 = log(delta).
  lp_theta3 <- dnorm(
    theta[3],
    mean = theta3_mean,
    sd = theta3_sd,
    log = TRUE
  )
  loglik + lp_beta + lp_eta + lp_theta3
}
make_logpost_cens <- function(
  theta3_sd = main_theta3_sd,
  theta3_mean = main_theta3_mean,
  a_beta = main_a_beta,
  b_beta = main_b_beta,
  a_eta = main_a_eta,

```



```

    b_eta = main_b_eta) {
  force(theta3_sd)
  force(theta3_mean)
  force(a_beta)
  force(b_beta)
  force(a_eta)
  force(b_eta)
  function(theta, data) {
    logpost_weibull13_cens(
      theta,
      data,
      a_beta = a_beta,
      b_beta = b_beta,
      a_eta = a_eta,
      b_eta = b_eta,
      theta3_mean = theta3_mean,
      theta3_sd = theta3_sd
    )
  }
}
# -----
# 4. Starting values
# -----
get_start_cens <- function(data, epsilon = PAD) {
  fail_times <- data$time[data$status == 1]
  t1 <- min(data$time)
  deltas <- exp(seq(
    log(1e-5),
    log(max(0.8 * sd(fail_times), 1e-3)),
    length.out = 25
  ))
  best <- NULL
  best_ll <- -Inf
  for (dlt in deltas) {
    g <- t1 - dlt - PAD
    y_fail <- fail_times - g

    if (any(y_fail <= 0)) {
      next
    }
    fit <- try(
      suppressWarnings(MASS::fitdistr(y_fail, "weibull")),
      silent = TRUE
    )
    if (inherits(fit, "try-error")) {
      next
    }
    beta0 <- as.numeric(fit$estimate["shape"])
    eta0 <- as.numeric(fit$estimate["scale"])
    theta0 <- c(log(beta0), log(eta0), log(dlt))
    lp <- logpost_weibull13_cens(theta0, data)
    if (is.finite(lp) && lp > best_ll) {
      best_ll <- lp
      best <- c(beta0, eta0, dlt)
    }
  }
  if (is.null(best)) {
    beta0 <- 1.3
    eta0 <- sd(fail_times)
    dlt <- 1
    best <- c(beta0, eta0, dlt)
  }
  c(log(best[1]), log(best[2]), log(best[3]))
}

```

```

}
# -----
# 5. Random-walk Metropolis helper functions
# -----
regularize_covariance <- function(Sigma, eps = 1e-8) {
  Sigma <- as.matrix(Sigma)
  Sigma <- (Sigma + t(Sigma)) / 2
  ok <- tryCatch({
    chol(Sigma)
    TRUE
  }, error = function(e) FALSE)
  if (ok) {
    return(Sigma)
  }
  Sigma + diag(eps, nrow(Sigma))
}
rw_pilot <- function(logpost, start, data, iters = 8000,
  scale = 1.0) {
  fitlap <- try(
    LearnBayes::laplace(logpost, start, data),
    silent = TRUE
  )
  if (inherits(fitlap, "try-error")) {
    fitlap <- list(mode = start, var = diag(0.2, 3))
  }
  fitlap$var <- regularize_covariance(fitlap$var)
  out <- LearnBayes::rwmetrop(
    logpost,
    list(var = fitlap$var, scale = scale),
    fitlap$mode,
    iters,
    data
  )
  list(
    mode = colMeans(out$par),
    var = regularize_covariance(cov(out$par))
  )
}
# Independence-sampler
# draws are not retained or reported.
check_laplace <- function(fitlap, label = "") {
  ok <- TRUE
  cond_num <- tryCatch(kappa(fitlap$var),
    error = function(e) Inf)
  chol_ok <- tryCatch({
    chol(fitlap$var)
    TRUE
  }, error = function(e) FALSE)
  if (!all(is.finite(fitlap$var)) ||
    !is.finite(cond_num) ||
    cond_num > 1e6 ||
    !chol_ok) {
    ok <- FALSE
  }
  ok
}
dnorm_mv <- function(x, mean, Sigma, log = TRUE) {
  x <- as.numeric(x)
  mean <- as.numeric(mean)
  k <- length(mean)
  R <- chol(Sigma)
  z <- backsolve(R, x - mean, transpose = TRUE)
  quad <- sum(z * z)
}

```

```

logdet <- sum(log(diag(R)))
lp <- -0.5 * (k * log(2 * pi) + 2 * logdet + quad)
if (log) lp else exp(lp)
}
dt_mv <- function(x, mean, Sigma, df, log = TRUE) {
  x <- as.numeric(x)
  mean <- as.numeric(mean)
  k <- length(mean)
  R <- chol(Sigma)
  z <- backsolve(R, x - mean, transpose = TRUE)
  quad <- sum(z * z)
  logdet <- sum(log(diag(R)))
  lp <- lgamma((df + k) / 2) -
    lgamma(df / 2) -
    0.5 * k * log(df * pi) -
    logdet -
    ((df + k) / 2) * loglp(quad / df)
  if (log) lp else exp(lp)
}
rmvt1 <- function(mean, Sigma, df) {
  mean <- as.numeric(mean)
  k <- length(mean)
  R <- chol(Sigma)
  z <- rnorm(k)
  chi2 <- rchisq(1, df = df)
  as.numeric(mean + (R %*% z) * sqrt(df / chi2))
}
rprop_mix1 <- function(prop) {
  if (runif(1) < prop$mix_p) {
    rmvt1(prop$mu, prop$var, df = prop$df)
  } else {
    as.numeric(MASS::mvrnorm(1, prop$mu, prop$var))
  }
}
dprop_mix1 <- function(x, prop, log = TRUE) {
  lp_norm <- dnorm_mv(x, prop$mu, prop$var, log = TRUE)
  lp_t <- dt_mv(x, prop$mu, prop$var, df = prop$df, log = TRUE)
  a <- max(lp_norm, lp_t)
  lse <- a + log(
    (1 - prop$mix_p) * exp(lp_norm - a) +
    prop$mix_p * exp(lp_t - a)
  )
  if (log) lse else exp(lse)
}
indepmetrop_mix <- function(logpost, start, data, iters, prop) {
  th <- as.numeric(start)
  lcur <- logpost(th, data)
  qcur <- dprop_mix1(th, prop, log = TRUE)
  out <- matrix(NA_real_, nrow = iters, ncol = length(th))
  acc <- 0L
  for (t in seq_len(iters)) {
    can <- rprop_mix1(prop)
    lcan <- logpost(can, data)
    if (is.finite(lcan)) {
      qcan <- dprop_mix1(can, prop, log = TRUE)
      log_alpha <- (lcan - lcur) + (qcur - qcan)

      if (log(runif(1)) < log_alpha) {
        th <- can
        lcur <- lcan
        qcur <- qcan
        acc <- acc + 1L
      }
    }
  }
}

```

```

    }
    out[t, ] <- th
  }
  list(par = out, acc = acc / iters)
}
trim_thin <- function(m) {
  m <- m[-seq_len(burn_in), , drop = FALSE]
  m[seq(1, nrow(m), by = thin), , drop = FALSE]
}
# -----
# 6. Multiple-chain random-walk MCMC for each population
# -----
results <- vector("list", K)
names(results) <- pop_names
for (i in seq_len(K)) {
  pop_i <- pop_names[i]
  data_i <- datasets[[pop_i]]
  logpost_i <- make_logpost_cens(theta3_sd = main_theta3_sd)
  start0 <- get_start_cens(data_i)
  chain_results <- vector("list", n_chains)
  for (ch in seq_len(n_chains)) {
    start_ch <- if (ch == 1) {
      start0
    } else {
      start0 + rnorm(3, 0, 0.3)
    }
    # Pilot random-walk run used to estimate the proposal
    # covariance.
    pilot <- tryCatch(
      rw_pilot(
        logpost_i,
        start_ch,
        data_i,
        iters = 8000,
        scale = 1.0
      ),
      error = function(e) {
        list(mode = start_ch, var = diag(0.2, 3))
      }
    )
    Sigma_pilot <- regularize_covariance(pilot$var)
    # Main random-walk Metropolis run.
    r.walk <- LearnBayes::rwmetrop(
      logpost_i,
      list(var = Sigma_pilot, scale = rw_scale_main),
      pilot$mode,
      iters_long,
      data_i
    )
    # -----
    # Preserve the dissertation RNG stream
    # -----
    # The original program ran this independence sampler here.
    # Its output is intentionally discarded; retaining the run
    # ensures
    # that every later random-walk chain starts from the same RNG
    # state
    # as in the dissertation analysis.
    fitlap <- tryCatch(
      LearnBayes::laplace(logpost_i, pilot$mode, data_i),
      error = function(e) {
        list(mode = pilot$mode, var = Sigma_pilot)
      }
    )
  }
}

```

```

)
fitlap$var <- regularize_covariance(fitlap$var)
laplace_ok <- check_laplace(
  fitlap,
  paste0("Population_", pop_i, ",_chain_", ch)
)
Sigma_prop <- regularize_covariance(
  if (laplace_ok) fitlap$var else Sigma_pilot
)
mu_prop <- if (laplace_ok) fitlap$mode else pilot$mode
prop <- list(
  mu = as.numeric(mu_prop),
  var = Sigma_prop * indep_inflate,
  df = mixture_df,
  mix_p = mixture_p
)
rng_compatibility_run <- indepmetrop_mix(
  logpost = logpost_i,
  start = mu_prop,
  data = data_i,
  iters = iters_long,
  prop = prop
)
rm(rng_compatibility_run, fitlap, prop)
acc_rw <- if (!is.null(r.walk$accept)) {
  r.walk$accept
} else {
  mean(apply(
    diff(r.walk$par),
    1,
    function(x) any(x != 0)
  ))
}
chain_results[[ch]] <- list(
  rw = trim_thin(r.walk$par),
  acc_rw = acc_rw
)
}
results[[pop_i]] <- list(
  chains = chain_results,
  data = data_i
)
}
# -----
# 7. Chain-aware convergence and sampling-efficiency diagnostics
# -----
# The four retained random-walk chains remain separate for all
# diagnostics. The reported R-hat is the maximum of the rank-
# normalized
# split and folded-split R-hat values. Bulk ESS and tail ESS are
# computed
# jointly across the four chains while preserving chain identity.
get_chain_quantity_matrix <- function(
  pop_i,
  quantity = c("gamma", "eta", "beta", "delta", "percentile"),
  p = NULL) {
  quantity <- match.arg(quantity)
  data_i <- datasets[[pop_i]]
  t1 <- min(data_i$time)
  chain_values <- lapply(results[[pop_i]]$chains, function(ch) {
    draws <- ch$rw
    beta <- exp(draws[, 1])
    eta <- exp(draws[, 2])
  })

```

```

delta <- exp(draws[, 3])
gamma <- t1 - delta - PAD
switch(
  quantity,
  gamma = gamma,
  eta = eta,
  beta = beta,
  delta = delta,
  percentile = {
    if (is.null(p) || length(p) != 1L || !is.finite(p) ||
        p <= 0 || p >= 1) {
      stop("For quantity = 'percentile', p must be a scalar in_
            (0, 1).")
    }
    gamma + eta * ((-log(1 - p))^(1 / beta))
  }
)
})
chain_lengths <- vapply(chain_values, length, integer(1))
if (length(unique(chain_lengths)) != 1L) {
  stop("All retained chains must have the same number of draws.")
}
out <- do.call(cbind, chain_values)
colnames(out) <- paste0("chain_", seq_len(ncol(out)))
out
}
safe_chain_diagnostic <- function(x, diagnostic_fun) {
  x <- as.matrix(x)
  if (nrow(x) < 4L || ncol(x) < 2L || any(!is.finite(x))) {
    return(NA_real_)
  }
  chain_sds <- apply(x, 2, stats::sd)
  if (any(!is.finite(chain_sds)) || any(chain_sds == 0)) {
    return(NA_real_)
  }
  tryCatch(
    as.numeric(diagnostic_fun(x)),
    error = function(e) NA_real_
  )
}
diagnose_chain_matrix <- function(x) {
  data.frame(
    Rhat = safe_chain_diagnostic(x, posterior::rhat),
    Bulk_ESS = safe_chain_diagnostic(x, posterior::ess_bulk),
    Tail_ESS = safe_chain_diagnostic(x, posterior::ess_tail),
    check.names = FALSE
  )
}
parameter_specs <- list(
  Gamma = list(quantity = "gamma", p = NULL),
  Eta = list(quantity = "eta", p = NULL),
  Beta = list(quantity = "beta", p = NULL),
  Delta = list(quantity = "delta", p = NULL)
)
percentile_specs <- list(
  t10 = list(quantity = "percentile", p = 0.10),
  t50 = list(quantity = "percentile", p = 0.50),
  t90 = list(quantity = "percentile", p = 0.90)
)
make_diagnostic_rows <- function(pop_i, specs, category) {
  bind_rows(lapply(names(specs), function(quantity_name) {
    spec <- specs[[quantity_name]]
    chain_matrix <- get_chain_quantity_matrix(

```

```

    pop_i = pop_i,
    quantity = spec$quantity,
    p = spec$p
  )
  diag_values <- diagnose_chain_matrix(chain_matrix)
  data.frame(
    Population = pop_i,
    Sampler = "Random_Walk",
    Category = category,
    Quantity = quantity_name,
    Rhat = diag_values$Rhat,
    Bulk_ESS = diag_values$Bulk_ESS,
    Tail_ESS = diag_values$Tail_ESS,
    check.names = FALSE
  )
}))
}
diagnostics_raw <- bind_rows(lapply(pop_names, function(pop_i) {
  bind_rows(
    make_diagnostic_rows(
      pop_i,
      parameter_specs,
      category = "Model_parameter"
    ),
    make_diagnostic_rows(
      pop_i,
      percentile_specs,
      category = "Weibull_percentile"
    )
  )
}))
# Common modern diagnostic guidelines are R-hat <= 1.01 and at
# least
# approximately 100 effective draws per chain. With four chains,
# the
# latter corresponds to a joint bulk/tail ESS of at least 400.
ess_target_total <- 100 * n_chains
diagnostics_detail <- diagnostics_raw %>%
  mutate(
    Rhat_Flag = case_when(
      !is.finite(Rhat) ~ "Check",
      Rhat > 1.01 ~ "Check",
      TRUE ~ "OK"
    ),
    ESS_Flag = case_when(
      !is.finite(Bulk_ESS) | !is.finite(Tail_ESS) ~ "Check",
      Bulk_ESS < ess_target_total | Tail_ESS < ess_target_total ~ "
        Check",
      TRUE ~ "OK"
    ),
    Overall_Flag = ifelse(
      Rhat_Flag == "OK" & ESS_Flag == "OK",
      "OK",
      "Check"
    ),
    Rhat = round(Rhat, 4),
    Bulk_ESS = round(Bulk_ESS, 1),
    Tail_ESS = round(Tail_ESS, 1)
  )
finite_max <- function(x) {
  x <- x[is.finite(x)]
  if (length(x) == 0L) NA_real_ else max(x)
}

```

```

finite_min <- function(x) {
  x <- x[is.finite(x)]
  if (length(x) == 0L) NA_real_ else min(x)
}
diagnostics_overview <- diagnostics_raw %>%
  group_by(Population, Category) %>%
  summarise(
    Max_Rhat = finite_max(Rhat),
    Min_Bulk_ESS = finite_min(Bulk_ESS),
    Min_Tail_ESS = finite_min(Tail_ESS),
    .groups = "drop"
  ) %>%
  mutate(
    Max_Rhat = round(Max_Rhat, 4),
    Min_Bulk_ESS = round(Min_Bulk_ESS, 1),
    Min_Tail_ESS = round(Min_Tail_ESS, 1)
  )
diagnostics_compact_core <- diagnostics_raw %>%
  group_by(Population) %>%
  summarise(
    Maximum_Rhat = finite_max(Rhat),
    Minimum_Parameter_Bulk_ESS = finite_min(
      Bulk_ESS[Category == "Model_parameter"]
    ),
    Minimum_Parameter_Tail_ESS = finite_min(
      Tail_ESS[Category == "Model_parameter"]
    ),
    Minimum_Percentile_Bulk_ESS = finite_min(
      Bulk_ESS[Category == "Weibull_percentile"]
    ),
    Minimum_Percentile_Tail_ESS = finite_min(
      Tail_ESS[Category == "Weibull_percentile"]
    ),
    .groups = "drop"
  )
print(diagnostics_detail)
print(diagnostics_overview)
# -----
# 8. Pool chains and recover parameters on the natural scale
# -----
pool_rw_chains <- function(results_i) {
  do.call(rbind, lapply(results_i$chains, function(ch) ch$rw))
}
recover_params_cens <- function(draws_mat, data_i) {
  t1 <- min(data_i$time)
  beta <- exp(draws_mat[, 1])
  eta <- exp(draws_mat[, 2])
  delta <- exp(draws_mat[, 3])
  gamma <- t1 - delta - PAD
  list(
    gamma = gamma,
    eta = eta,
    beta = beta,
    delta = delta
  )
}
params_rw <- lapply(pop_names, function(pop_i) {
  recover_params_cens(
    pool_rw_chains(results[[pop_i]]),
    datasets[[pop_i]]
  )
})
names(params_rw) <- pop_names

```



```

assemble_mat <- function(params_list, param_name) {
  out <- do.call(cbind, lapply(params_list, `[`, param_name))
  colnames(out) <- pop_names
  out}
g.rw.s <- assemble_mat(params_rw, "gamma")
eta.rw.s <- assemble_mat(params_rw, "eta")
beta.rw.s <- assemble_mat(params_rw, "beta")
delta.rw.s <- assemble_mat(params_rw, "delta")
# -----
# 9. Posterior summaries for the random-walk sampler
# -----
make_summary_rows_rw <- function(mat_rw, parameter_name) {
  data.frame(
    Parameter = parameter_name,
    Population = pop_names,
    RW_Mean = round(colMeans(mat_rw), 4),
    RW_Median = round(apply(mat_rw, 2, median), 4),
    RW_Q2.5 = round(
      apply(mat_rw, 2, quantile, probs = 0.025),
      4
    ),
    RW_Q97.5 = round(
      apply(mat_rw, 2, quantile, probs = 0.975),
      4
    ),
    check.names = FALSE)
}
posterior_summary <- bind_rows(
  make_summary_rows_rw(g.rw.s, "Gamma"),
  make_summary_rows_rw(eta.rw.s, "Eta"),
  make_summary_rows_rw(beta.rw.s, "Beta"),
  make_summary_rows_rw(delta.rw.s, "Delta")
)
# -----
# 10. Posterior probability that each population is best
# -----
sel_prob <- function(mat) {
  idx <- apply(mat, 1, which.max)
  tabulate(idx, nbins = K) / nrow(mat)
}
selection_probs_gamma <- data.frame(
  Population = pop_names,
  RW_P_Best_Gamma = round(sel_prob(g.rw.s), 4),
  check.names = FALSE)
# Posterior probability that each population has the largest eta or
# beta.
selection_probs_eta <- data.frame(
  Population = pop_names,
  RW_P_Best_Eta = round(sel_prob(eta.rw.s), 4),
  check.names = FALSE)
selection_probs_beta <- data.frame(
  Population = pop_names,
  RW_P_Best_Beta = round(sel_prob(beta.rw.s), 4),
  check.names = FALSE)
# Combined table of posterior probabilities of having the largest
# parameter value.
selection_probs_parameters <- bind_rows(
  data.frame(
    Parameter = "Gamma",
    Population = pop_names,
    RW_Probability = round(sel_prob(g.rw.s), 4),
    check.names = FALSE ),
  data.frame(

```

```

    Parameter = "Eta",
    Population = pop_names,
    RW_Probability = round(sel_prob(eta.rw.s), 4),
    check.names = FALSE ),
  data.frame(
    Parameter = "Beta",
    Population = pop_names,
    RW_Probability = round(sel_prob(beta.rw.s), 4),
    check.names = FALSE )
)
# Selected subset with target posterior probability P*.
P_star <- 0.95
selection_order_rw <- selection_probs_gamma %>%
  arrange(desc(RW_P_Best_Gamma)) %>%
  mutate(
    Cumulative_Probability = cumsum(RW_P_Best_Gamma)
  )
cutoff_index <- which(
  selection_order_rw$Cumulative_Probability >= P_star
)[1]
selected_subset_rw <- selection_order_rw[seq_len(cutoff_index), ]
# -----
# 11. Weibull percentiles and percentile selection probabilities
# -----
tp_from_draws <- function(g_mat, eta_mat, beta_mat, p) {
  z <- -log(1 - p)
  g_mat + eta_mat * (z^(1 / beta_mat))
}
tp_rw_list <- lapply(perc, function(p) {
  tp_from_draws(g.rw.s, eta.rw.s, beta.rw.s, p)
})
names(tp_rw_list) <- paste0("t_", perc * 100)
tp_summary <- bind_rows(lapply(seq_along(perc), function(j) {
  make_summary_rows_rw(
    tp_rw_list[[j]],
    paste0("t_", perc[j] * 100, "%")
  )
})))
tp_selection_probs <- bind_rows(lapply(seq_along(perc),
function(j) {
  data.frame(
    Percentile = paste0("t_", perc[j] * 100, "%"),
    Population = pop_names,
    RW_P_Largest = round(sel_prob(tp_rw_list[[j]]), 4),
    check.names = FALSE
  )
})))
joint_sel <- bind_rows(lapply(seq_along(perc), function(j) {
  gamma_winner_rw <- apply(g.rw.s, 1, which.max)
  tp_winner_rw <- apply(tp_rw_list[[j]], 1, which.max)
  data.frame(
    Percentile = paste0("t_", perc[j] * 100, "%"),
    RW_P_Gamma_and_tp_Agree = round(
      mean(gamma_winner_rw == tp_winner_rw),
      4
    ),
    check.names = FALSE
  )
})))
# Print the main posterior summaries.
cat("\n====_Posterior_summaries_for_Weibull_parameters_====\n")
print(posterior_summary, row.names = FALSE)
cat("\n====_Posterior_summaries_for_Weibull_percentiles_====\n")

```

```

print(tp_summary, row.names = FALSE)
# -----
# 12. Acceptance-rate summaries
# -----
acceptance_df <- bind_rows(lapply(pop_names, function(pop_i) {
  rw_acc <- sapply(
    results[[pop_i]]$chains,
    function(ch) ch$acc_rw
  )
  data.frame(
    Population = pop_i,
    RW_Acceptance_Mean = round(mean(rw_acc), 3),
    RW_Acceptance_Min = round(min(rw_acc), 3),
    RW_Acceptance_Max = round(max(rw_acc), 3),
    check.names = FALSE
  )
}))
# -----
# 13. Compact random-walk diagnostic table for the manuscript
# -----
# ESS values below are joint chain-aware ESS values, not arithmetic
# "per-chain ESS" values.
mcmc_diagnostics_table <- acceptance_df %>%
  transmute(
    Population,
    Mean_RW_Acceptance = RW_Acceptance_Mean
  ) %>%
  left_join(diagnostics_compact_core, by = "Population") %>%
  mutate(
    Maximum_Rhat = round(Maximum_Rhat, 4),
    across(
      ends_with("ESS"),
      ~ round(.x, 1)
    )
  )
print(mcmc_diagnostics_table)
# -----
# 14. Posterior predictive checks for censored observations
# -----
rweibull3 <- function(n, beta, eta, gamma) {
  gamma + stats::rweibull(n, shape = beta, scale = eta)
}
ppc_censored <- function(
  data,
  beta_draws,
  eta_draws,
  gamma_draws,
  censor_time = 300,
  n_rep = 1000) {
  n <- nrow(data)
  obs_stats <- c(
    min = min(data$time),
    mean = mean(data$time),
    sd = sd(data$time),
    failures = sum(data$status == 1),
    censored = sum(data$status == 0),
    q10 = as.numeric(quantile(data$time, 0.10)),
    median = median(data$time),
    q90 = as.numeric(quantile(data$time, 0.90))
  )
  idx <- sample(seq_along(beta_draws), n_rep, replace = TRUE)
  sim_stats <- t(sapply(idx, function(s) {
    true_time <- rweibull3(

```

```

      n,
      beta = beta_draws[s],
      eta = eta_draws[s],
      gamma = gamma_draws[s]
    )
    obs_time <- pmin(true_time, censor_time)
    status <- ifelse(true_time <= censor_time, 1, 0)
    c(
      min = min(obs_time),
      mean = mean(obs_time),
      sd = sd(obs_time),
      failures = sum(status == 1),
      censored = sum(status == 0),
      q10 = as.numeric(quantile(obs_time, 0.10)),
      median = median(obs_time),
      q90 = as.numeric(quantile(obs_time, 0.90))
    )
  )))
data.frame(
  Statistic = names(obs_stats),
  Observed = round(as.numeric(obs_stats), 4),
  Sim_Mean = round(colMeans(sim_stats), 4),
  Sim_SD = round(apply(sim_stats, 2, sd), 4),
  Bayes_pval = round(sapply(seq_along(obs_stats), function(j) {
    mean(sim_stats[, j] >= obs_stats[j])
  }), 3),
  check.names = FALSE
)
}
ppc_pvals <- bind_rows(lapply(seq_len(K), function(i) {
  ppc <- ppc_censored(
    data = datasets[[pop_names[i]]],
    beta_draws = beta.rw.s[, i],
    eta_draws = eta.rw.s[, i],
    gamma_draws = g.rw.s[, i],
    censor_time = 300,
    n_rep = ppc_nrep
  )
  ppc$Population <- pop_names[i]
  ppc %>% relocate(Population)
})))
# -----
# 15. Prior-location sensitivity analysis for theta3_mean
# -----
# The prior is  $\log(\delta_i) \sim \text{Normal}(\theta_3\text{mean}, \text{main\_}\theta_3\text{sd}^2)$ 
#
# Vary  $\theta_3\text{mean}$  over  $\log(0.5)$ ,  $\log(1)$ , and  $\log(2)$ , while holding
# the prior standard deviation and all other model settings fixed.
# For each setting, recompute gamma summaries, posterior
# probabilities
# of being best, the complete population ordering, and the selected
# subset at  $P_{\text{star}}$ .
sens_iters <- 30000
sens_burn <- 5000
sensitivity_runs <- lapply(seq_along(theta3_mean_grid), function(k)
{
  mu_val <- theta3_mean_grid[k]
  prior_median_delta <- exp(mu_val)
  cat(
    "\nRunning_theta3_mean_", round(mu_val, 6),
    "(prior_median_delta_", prior_median_delta, ")\n"
  )
})
pop_runs <- lapply(pop_names, function(pop_i) {

```

```

data_i <- datasets[[pop_i]]
logpost_mu <- make_logpost_cens(
  theta3_mean = mu_val,
  theta3_sd = main_theta3_sd
)
start <- get_start_cens(data_i)
pilot <- tryCatch(
  rw_pilot(logpost_mu, start, data_i, iters = 5000),
  error = function(e) {
    warning(
      paste(
        "Pilot_failed_for", pop_i,
        "at_theta3_mean=", round(mu_val, 6),
        ";_using_fallback_covariance."
      )
    )
    list(mode = start, var = diag(0.2, 3))
  }
)
r.walk <- LearnBayes::rwmetrop(
  logpost_mu,
  list(
    var = regularize_covariance(pilot$var),
    scale = rw_scale_main
  ),
  pilot$mode,
  sens_iters,
  data_i
)
post <- r.walk$par[-seq_len(sens_burn), , drop = FALSE]
t1 <- min(data_i$time)
delta_draws <- exp(post[, 3])
gamma_draws <- t1 - delta_draws - PAD
list(
  gamma = gamma_draws,
  delta = delta_draws,
  parameter_summary = data.frame(
    theta3_mean = mu_val,
    Prior_Median_Delta = prior_median_delta,
    Population = pop_i,
    Gamma_Mean = round(mean(gamma_draws), 4),
    Gamma_Q2.5 = round(quantile(gamma_draws, 0.025), 4),
    Gamma_Q97.5 = round(quantile(gamma_draws, 0.975), 4),
    Delta_Mean = round(mean(delta_draws), 4),
    Delta_Q2.5 = round(quantile(delta_draws, 0.025), 4),
    Delta_Q97.5 = round(quantile(delta_draws, 0.975), 4),
    Acceptance_Rate = round(
      if (!is.null(r.walk$accept)) {
        r.walk$accept
      } else {
        mean(apply(diff(r.walk$par), 1, function(x) any(x != 0)
          ))
      },
      3
    ),
    check.names = FALSE
  )
)
}))
names(pop_runs) <- pop_names
# Pair equal-index draws from the independently fitted
# populations to
# obtain draws from the joint product posterior.

```

```

gamma_mat <- do.call(
  cbind,
  lapply(pop_runs, function(x) x$gamma)
)
colnames(gamma_mat) <- pop_names
p_best <- sel_prob(gamma_mat)
order_idx <- order(p_best, decreasing = TRUE)
cumulative_ordered <- cumsum(p_best[order_idx])
cutoff <- which(cumulative_ordered >= P_star)[1]
selected_names <- pop_names[order_idx[seq_len(cutoff)]]
cumulative_by_population <- numeric(K)
cumulative_by_population[order_idx] <- cumulative_ordered
selection_summary <- data.frame(
  theta3_mean = mu_val,
  Prior_Median_Delta = prior_median_delta,
  Population = pop_names,
  P_Best_Gamma = round(p_best, 4),
  Rank = match(seq_len(K), order_idx),
  Cumulative_Probability_at_Rank = round(
    cumulative_by_population,
    4
  ),
  Selected = pop_names %in% selected_names,
  check.names = FALSE
)
subset_summary <- data.frame(
  theta3_mean = mu_val,
  Prior_Median_Delta = prior_median_delta,
  Full_Ordering = paste(pop_names[order_idx], collapse = ">"),
  Selected_Subset = paste(selected_names, collapse = ","),
  Subset_Size = length(selected_names),
  Subset_Cumulative_Probability = round(
    cumulative_ordered[cutoff],
    4
  ),
  check.names = FALSE
)
list(
  parameter_summary = bind_rows(
    lapply(pop_runs, function(x) x$parameter_summary)
  ),
  selection_summary = selection_summary,
  subset_summary = subset_summary
)
}))
sensitivity_results <- bind_rows(
  lapply(sensitivity_runs, `[`, "parameter_summary")
)
sensitivity_selection_results <- bind_rows(
  lapply(sensitivity_runs, `[`, "selection_summary")
)
sensitivity_subset_results <- bind_rows(
  lapply(sensitivity_runs, `[`, "subset_summary")
)
# Compare each prior-location setting with theta3_mean = log(1) =
# 0.
baseline_row <- which.min(
  abs(sensitivity_subset_results$theta3_mean - main_theta3_mean)
)
baseline_ordering <- sensitivity_subset_results$Full_Ordering[
  baseline_row]
baseline_subset <- sensitivity_subset_results$Selected_Subset[
  baseline_row]

```

```

sensitivity_comparison <- sensitivity_subset_results %>%
  mutate(
    Ordering_Unchanged_vs_Baseline =
      Full_Ordering == baseline_ordering,
    Selected_Subset_Unchanged_vs_Baseline =
      Selected_Subset == baseline_subset
  )
sensitivity_probability_ranges <- sensitivity_selection_results %>%
  group_by(Population) %>%
  summarise(
    Min_P_Best_Gamma = min(P_Best_Gamma),
    Max_P_Best_Gamma = max(P_Best_Gamma),
    Range_P_Best_Gamma = round(
      Max_P_Best_Gamma - Min_P_Best_Gamma,
      4
    ),
    .groups = "drop"
  )
print(sensitivity_results)
print(sensitivity_selection_results)
print(sensitivity_subset_results)
print(sensitivity_comparison)
print(sensitivity_probability_ranges)
ordering_unchanged <- all(
  sensitivity_comparison$Ordering_Unchanged_vs_Baseline
)
subset_unchanged <- all(
  sensitivity_comparison$Selected_Subset_Unchanged_vs_Baseline
)
cat("\n====_Prior-location_sensitivity_conclusion_====\n")
if (ordering_unchanged && subset_unchanged) {
  cat(
    "The_population_ordering_and_selected_subset_are_unchanged_",
    "across_",
    "theta3_mean_=_log(0.5),_log(1),_and_log(2). \n",
    sep = ""
  )
} else {
  cat(
    "At_least_one_prior-location_setting_changes_the_population_",
    "ordering_",
    "or_selected_subset._Inspect_the_sensitivity_result_objects.\n",
    sep = ""
  )
}
write.csv(
  sensitivity_results,
  "PriorLocationSensitivity_ParameterSummaries.csv",
  row.names = FALSE)
write.csv(
  sensitivity_selection_results,
  "PriorLocationSensitivity_SelectionProbabilities.csv",
  row.names = FALSE)
write.csv(
  sensitivity_subset_results,
  "PriorLocationSensitivity_SelectedSubsets.csv",
  row.names = FALSE)
write.csv(
  sensitivity_comparison,
  "PriorLocationSensitivity_Comparison.csv",
  row.names = FALSE)
write.csv(
  sensitivity_probability_ranges,

```

```

"PriorLocationSensitivity_ProbabilityRanges.csv",
row.names = FALSE)
# -----
# 16. Posterior densities of gamma for P1-P5
# -----
gamma_range <- range(g.rw.s, finite = TRUE)
gamma_density_df <- bind_rows(lapply(seq_len(K), function(i) {
  dens_i <- stats::density(
    g.rw.s[, i],
    from = gamma_range[1],
    to = gamma_range[2],
    n = 1024,
    cut = 0 )
  data.frame(
    Gamma = dens_i$x,
    Density = dens_i$y,
    Population = factor(
      pop_names[i],
      levels = pop_names
    ),
    check.names = FALSE )
}))
gamma_palette <- c(
  P1 = "#0072B2",
  P2 = "#E69F00",
  P3 = "#009E73",
  P4 = "#CC79A7",
  P5 = "#D55E00")
gamma_density_plot <- ggplot(
  gamma_density_df,
  aes(
    x = Gamma,
    y = Density,
    fill = Population,
    color = Population
  )
) +
  geom_area(
    alpha = 0.42,
    linewidth = 0
  ) +
  geom_line(
    linewidth = 0.8
  ) +
  geom_hline(
    yintercept = 0,
    color = "grey55",
    linewidth = 0.3
  ) +
  facet_grid(
    rows = vars(Population),
    scales = "free_y",
    switch = "y"
  ) +
  scale_fill_manual(values = gamma_palette) +
  scale_color_manual(values = gamma_palette) +
  scale_x_continuous(
    expand = expansion(mult = c(0.01, 0.01))
  ) +
  labs(
    x = expression(gamma[i]),
    y = NULL
  ) +

```



```

theme_classic(base_size = 11) +
theme(
  legend.position = "none",
  strip.placement = "outside",
  strip.background = element_blank(),
  strip.text.y.left = element_text(
    angle = 0,
    face = "bold",
    hjust = 1
  ),
  panel.spacing.y = grid::unit(0.12, "lines"),
  panel.grid = element_blank(),
  axis.text.y = element_blank(),
  axis.ticks.y = element_blank(),
  axis.line.y = element_blank()
)
print(gamma_density_plot)
ggsave(
  filename = "Posterior_Densities_Gamma_P1_to_P5_RW_Stacked.pdf",
  plot = gamma_density_plot,
  width = 7.2,
  height = 7.4,
  units = "in"
)
ggsave(
  filename = "Posterior_Densities_Gamma_P1_to_P5_RW_Stacked.png",
  plot = gamma_density_plot,
  width = 7.2,
  height = 7.4,
  units = "in",
  dpi = 600,
  bg = "white"
)
# -----
# 17. Prior-location sensitivity plot
# -----
# The posterior mean is shown by a solid line,
# and the lower and upper 95% credible limits are shown by dashed
# lines.
pdf(
  "PriorLocationSensitivity_Gamma.pdf",
  width = 12,
  height = 6,
  pointsize = 16)
old_par <- par(no.readonly = TRUE)
par(
  mfrow = c(ceiling(K / 2), 2),
  mar = c(4.2, 4.5, 2.8, 1.0),
  mgp = c(2.6, 0.8, 0),
  oma = c(0.3, 0.3, 0.2, 0.2)
)
mean_col <- "steelblue"
ci_col <- "gray40"
for (i in seq_len(K)) {
  sub <- sensitivity_results %>%
    dplyr::filter(Population == pop_names[i]) %>%
    dplyr::arrange(Prior_Median_Delta)
  y_limits <- range(
    c(sub$Gamma_Q2.5, sub$Gamma_Q97.5),
    finite = TRUE
  )
  y_padding <- 0.04 * diff(y_limits)
  if (!is.finite(y_padding) || y_padding == 0) {

```

```

    y_padding <- 0.5
  }
  plot(
    sub$Prior_Median_Delta,
    sub$Gamma_Mean,
    type = "b",
    pch = 19,
    col = mean_col,
    lwd = 2,
    xaxt = "n",
    xlim = c(0.45, 2.05),
    ylim = y_limits + c(-y_padding, y_padding),
    xlab = expression("Prior_median_of_" * delta[i]),
    ylab = expression("Posterior_mean_of_" * gamma[i]),
    main = paste("Prior-location_sensitivity:", pop_names[i]) )
  axis(
    1,
    at = c(0.5, 1, 2),
    labels = c("0.5", "1.0", "2.0") )
  lines(
    sub$Prior_Median_Delta,
    sub$Gamma_Q2.5,
    lty = 2,
    col = ci_col,
    lwd = 2 )
  lines(
    sub$Prior_Median_Delta,
    sub$Gamma_Q97.5,
    lty = 2,
    col = ci_col,
    lwd = 2 )
  legend(
    "right",
    legend = c("Posterior_mean", "95%_credible_interval"),
    col = c(mean_col, ci_col),
    lty = c(1, 2),
    lwd = 2,
    pch = c(19, NA),
    bty = "n",
    cex = 0.85 )
}
# Leave the sixth panel empty when there are five populations.
if (K %% 2 == 1) {
  plot.new()
}
par(old_par)
dev.off()

```