

SUBSET SELECTION PROCEDURES FOR POPULATIONS FOLLOWING WEIBULL
DISTRIBUTIONS

by

JEZERCA HODAJ

A dissertation submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN APPLIED MATHEMATICAL SCIENCES

2026

Oakland University
Rochester, Michigan

Doctoral Advisory Committee:

Dorin Drignei, Ph.D., Chair
Gary C McDonald, Ph.D., Co-Chair
Hon Yiu So, Ph.D.
Li Li, Ph.D.

© by JEZERCA HODAJ, 2026
All rights reserved

To my parents, with love and gratitude.

ACKNOWLEDGMENTS

I would first like to express my deepest gratitude to my advisor, Dr. Gary C McDonald, for his continuous support throughout my Ph.D. journey. His guidance, patience, and encouragement have helped me grow both academically and personally. I am especially thankful for the confidence he helped me build and for the kindness he has shown throughout this process.

I sincerely thank my committee members, Dr. Dorin Drignei, Dr. Hon Yiu So, and Dr. Li Li, for their time, insight, and helpful suggestions. I am grateful to have learned from each of them as my professors and for their contributions to my academic and professional growth.

I am deeply thankful to Oakland University and the Department of Mathematics and Statistics for the opportunity to pursue my doctoral studies and for their support through my teaching assistantship. I would especially like to thank Professor Anna Maria Spagnuolo for her exceptional care, encouragement, and belief in me from my first days at Oakland University.

Finally, I would like to thank the people closest to my heart. To my parents and family, thank you for your love, sacrifices, and constant support. To my friends, thank you for making this journey more meaningful. You became my family away from home, and I am grateful for all of you more than words can say.

JEZERCA HODAJ

ABSTRACT

SUBSET SELECTION PROCEDURES FOR POPULATIONS FOLLOWING WEIBULL DISTRIBUTIONS

by

JEZERCA HODAJ

Adviser: Gary C McDonald, Ph.D.

In reliability studies, practitioners often need to compare several populations and identify the one with the most favorable lifetime behavior. When sample sizes are limited, populations are similar, or observations are censored, selecting a single best population can be unstable. This dissertation studies subset selection procedures for reliability populations in which the best population is defined as the one with the largest threshold parameter. Since the threshold parameter represents a lower endpoint or minimum lifetime under the assumed model, it provides a meaningful criterion in applications where early failures are especially costly.

The dissertation develops and compares both frequentist and Bayesian subset selection procedures. In the frequentist setting, exponential populations with threshold parameters and a common known scale parameter are considered. Selection rules based on sample minima and sample means are developed and calibrated to satisfy a prescribed probability of correct selection. Their performance is evaluated using operating characteristics such as the probability of correct selection, the probability of incorrect selection, and the expected subset size. Exact operating characteristics are also derived for the two-population case, including comparisons of minimum-based, CLT-based mean, and gamma-based mean rules.

The Bayesian component extends the threshold-based subset selection framework to three-parameter Weibull populations. Posterior simulation is used to estimate the probability that each population has the largest threshold parameter, and the Bayesian selected subset is constructed to achieve a target posterior probability of containing the best population. The method is illustrated using a fully simulated five-population Weibull study and a semi-simulated censored study based on Alloy T7987 fatigue-life data.

Overall, this dissertation provides a unified framework for threshold-based subset selection in reliability settings. The frequentist procedures emphasize calibrated probability guarantees and operating-characteristic performance, while the Bayesian procedure provides a posterior probability interpretation that naturally incorporates parameter uncertainty and censoring.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	iv
ABSTRACT	v
LIST OF TABLES	xix
LIST OF FIGURES	xxv
LIST OF ABBREVIATIONS	xxxi
CHAPTER 1	
INTRODUCTION	1
1.1. Motivation and Overview	1
1.2. Dissertation Contributions	3
1.3. Organization of the Dissertation	6
CHAPTER 2	
BACKGROUND AND PROBLEM FORMULATION	8
2.1. Subset Selection Procedures	8
2.1.1. Subset Selection Objective	8
2.1.2. Why Subset Selection Rather Than Single-Best Selection	9
2.2. Notation and Problem Setup	9
2.3. Classical Subset Selection Procedures	10
2.3.1. Least Favorable Configurations and Calibration	10
2.3.2. Procedure Families	11
2.4. Indifference Zone Procedures	11

TABLE OF CONTENTS—Continued

2.4.1. Indifference Zone Condition and Sample Size Logic	11
2.4.2. Multi-stage and Sequential Methods	12
2.5. Operating Characteristics and Performance Metrics	12
2.6. Bayesian Ranking and Selection and Posterior-Based Subset Rules	13
2.7. Weibull and Exponential Models in Reliability	15
2.8. Concluding Remarks	17
CHAPTER 3	
FREQUENTIST SELECTION RULES FOR THRESHOLD PARAMETERS	19
3.1. Model Setup	19
3.2. Subset Rule Based on Sample Minima (R_1)	21
3.2.1. Definition of the Rule	22
3.2.2. Distribution of the Minimum Order Statistic	22
3.2.3. Probability of Correct Selection (CS) and Calibration of d	23
3.3. Subset Selection Rule Based on Sample Means (R_2)	24
3.3.1. Definition of Rule R_2	25
3.3.2. Approximate Calibration of b Using the Central Limit Theorem	26

TABLE OF CONTENTS—Continued

3.4. A Gamma-Distribution-Based Subset Selection Rule R_3	27
3.4.1. Rule R_3 – Sum Form of the Mean-based Procedure	28
3.4.2. Probability of Correct Selection and Calibration of b	28
3.4.3. Sensitivity to the common-scale assumption	29
3.5. Indifference Zone Procedure	29
3.5.1. Probability of Correct Selection Under the Indifference Zone (IZ) Requirement	30
3.5.2. Least Favorable Configuration and Sample Size Determination	31
3.6. Numerical Illustrations and Implementation Notes	32
3.6.1. Application of the Two Selection Rules	32
3.6.2. Indifference Zone Illustration	35
3.6.3. Implementation Notes	35
3.7. Results and Takeaways	36
CHAPTER 4 OPERATING CHARACTERISTICS AND COMPARATIVE PERFORMANCE	38
4.1. Operating Characteristics and Threshold Configurations	38
4.1.1. Operating Characteristics	38

TABLE OF CONTENTS—Continued

4.1.2. Threshold Configurations	39
4.2. Operating Characteristics of the Selection Rules	40
4.2.1. Simulation-based Operating Characteristics (OCs) Evaluation	40
4.2.2. Slippage Configuration Results	40
4.2.3. Equi-spaced Configuration Results	42
4.2.4. Explaining the Expected Subset Size Expected Subset Size (ESS) Gap via Moments	42
4.3. Operating Characteristics Under Gamma Distribution	45
4.4. Results and Takeaways	46
CHAPTER 5	
EXACT RESULTS FOR TWO POPULATIONS	50
5.1. Exact OCs for $k = 2$ Populations	50
5.2. Exact Computation of Constants d and b and OCs Evaluation	51
5.2.1. Exact OCs for R_1	51
5.2.2. Exact OCs for Rule R_2	52
5.2.3. Exact OCs for Gamma-Based Rule R_3	52
5.3. Comparative Findings Under Slippage and Equi-spaced Configuration	54
5.4. Numerical Verification of Operating Characteristic Comparisons, $k = 2$	57

TABLE OF CONTENTS—Continued

5.4.1. Small-Sample Case: $n = 5$	59
5.4.2. Moderate-Sample Case: $n = 25$	63
5.4.3. Large-Sample Case: $n = 100$	66
5.5. Results and Takeaways	72
 CHAPTER 6	
BAYESIAN SUBSET SELECTION FOR THREE-PARAMETER WEIBULL POPULATIONS	75
6.1. Introduction	75
6.2. Bayesian Subset Selection Framework and Decision Rule	78
6.2.1. Classical Subset Selection Background	78
6.2.2. Definition of the Best Population	79
6.2.3. Posterior Probability of Being Best	79
6.2.4. Posterior Probability of Correct Selection	81
6.2.5. Bayesian Selected Subset	81
6.2.6. Percentile-Based Reliability Comparisons	82
6.3. Bayesian Model Specification	83
6.3.1. Three-Parameter Weibull Model for Exact Failure Times	83
6.3.2. Three-Parameter Weibull Model for Right-Censored Data	84
6.3.3. Transformation Used for Posterior Computation	85
6.4. Prior Specification	86

TABLE OF CONTENTS—Continued

6.5. Posterior Computation	89
6.5.1. Random-Walk Metropolis Sampler	90
6.5.2. Independence Metropolis–Hastings Sampler	91
6.5.3. Implementation Details	93
6.6. Bayesian Subset Selection in a Weibull Simulation Study	94
6.6.1. Data-Generating Configuration	94
6.6.2. Posterior Model for the Generated Failure-Time Data	95
6.6.3. Posterior Estimation of Weibull Parameters	97
6.6.4. Posterior Selection Probabilities	107
6.6.5. Posterior Weibull Percentiles	108
6.6.6. Markov Chain Monte Carlo (MCMC) Diagnostics	112
6.6.7. Prior Sensitivity Analysis	120
6.6.8. Posterior Predictive Checks	124
6.6.9. Summary of the Fully Simulated Study	128
6.7. Alloy T7987 Fatigue Life Data	130
6.7.1. Data Description	130
6.7.2. Maximum Likelihood Fit	131
6.7.3. Graphical Assessment of the Maximum Likelihood Fit	132

TABLE OF CONTENTS—Continued

6.7.4.	Bayesian Fit to the Original Fatigue-Life Data	141
6.7.5.	Posterior Density Plots	143
6.7.6.	Traceplots and MCMC Behavior	145
6.7.7.	Summary of the Original Fatigue-Life Analysis	146
6.8.	Semi-Simulated Censored Study Based on the Alloy Fatigue Data	148
6.8.1.	Construction of the Semi-Simulated Populations	148
6.8.2.	Posterior Summaries for Weibull Parameters	150
6.8.3.	Posterior Selection Probabilities	154
6.8.4.	Posterior Weibull Percentiles	160
6.8.5.	MCMC Diagnostics	166
6.8.6.	Prior Sensitivity Analysis	174
6.8.7.	Posterior Predictive Checks	178
6.8.8.	Summary of the Semi-Simulated Alloy Fatigue Study	184
6.9.	Results and Takeaways	185
CHAPTER 7		
CONCLUSIONS		189
7.1.	Summary of the Dissertation	189
7.2.	Main Findings	190

TABLE OF CONTENTS—Continued

7.3. Practical Implications	192
7.4. Limitations and Future Work	193
7.5. Final Remarks	195

APPENDICES

A. Proofs and Derivations	196
A.1. Deriving the $\Pr(CS)$ $\Pr(ICS)$ for rule R_1 when $k = 2$	197
A.2. Deriving Probability of Incorrect Selection $\Pr(ICS)$ for Rule R_3 when $k = 2$	201
A.2.1. Case 1: $b \geq \delta$.	202
A.2.2. Case 2: $\delta \geq b$.	203

APPENDICES

B. Source Code for Constants and Simulations	205
B.1. Source Code for Constant d	206
B.2. Source Code for Constant b	207
B.3. Source Code for the Sample Statistics and the Selected Populations	208
B.4. Source code for Indifference Zone Example	211
B.5. Source code for OCs under Slippage Configuration	211
B.6. Source code for OCs under Equi-Spaced Configuration	215

TABLE OF CONTENTS—Continued

B.7.	Source Code for Expected Subset Size Plots	218
B.8.	Source Code for CLT Approach of R2	220
B.9.	Source Code for Gamma Distribution Approach of R2	222
B.10.	Source Code for OCs for 5 Populations and Small Sample Size, Slippage Configuration	223
B.11.	Source Code for OCs for 5 Populations and Small Sample Size, Equi-Spaced Configuration	226
B.12.	Source Code for OCs for 2 Populations and Various Sample Sizes	230
B.13.	Source Code for Exploring the Differences of OCs of Rule 1 and Rule 2, when $k = 2$	231
B.14.	Source Code for Exploring the Differences of OCs of Rule 1 and Rule 3, when $k = 2$	234
B.15.	Source Code for Heat Map of Differences of OCs of Rule 1 and Rule 2, when $k = 2$	238
B.16.	Source Code for Heat Map of Differences of OCs of Rule 1 and Rule 3, when $k = 2$	240
B.17.	Source Code for the Fully Simulated Weibull Study	244
B.18.	Source Code for the MLE Fit of Alloy T7987 Data	275

TABLE OF CONTENTS—Continued

B.19.	Source Code for the Bayesian Fit of Alloy T7987 Data	284
B.20.	Source Code for the Bayesian Subset Selection of Semi-Simulated Alloy T7987 Study	294

APPENDICES

C.	Extra Tables and Figures	324
C.1.	Complete Tables for OCs of Rule 1, Rule 2, and Rule 3 when $k = 2$	325
C.1.1.	Complete Tables for OCs of Rule 1 vs Rule 2: $n = 5$, $P^* = 0.95$	325
C.1.2.	Complete Tables for OCs of Rule 1 vs Rule 2: $n = 5$, $P^* = 0.99$	326
C.1.3.	Complete Tables for OCs of Rule 1 vs Rule 3: $n = 5$, $P^* = 0.95$	327
C.1.4.	Complete Tables for OCs of Rule 1 vs Rule 3: $n = 5$, $P^* = 0.99$	328
C.1.5.	Complete Tables for OCs of Rule 1 vs Rule 2: $n = 25$, $P^* = 0.95$	329
C.1.6.	Complete Tables for OCs of Rule 1 vs Rule 2: $n = 25$, $P^* = 0.99$	330
C.1.7.	Complete Tables for OCs of Rule 1 vs Rule 3: $n = 25$, $P^* = 0.95$	331
C.1.8.	Complete Tables for OCs of Rule 1 vs Rule 3: $n = 25$, $P^* = 0.99$	332

TABLE OF CONTENTS—Continued

C.1.9.	Complete Tables for OCs of Rule 1 vs Rule 2: $n = 100$, $P^* = 0.95$	333
C.1.10.	Complete Tables for OCs of Rule 1 vs Rule 2: $n = 100$, $P^* = 0.99$	334
C.1.11.	Complete Tables for OCs of Rule 1 vs Rule 3: $n = 100$, $P^* = 0.95$	335
C.1.12.	Complete Tables for OCs of Rule 1 vs Rule 3: $n = 100$, $P^* = 0.99$	336
C.2.	Heat Maps for OCs of Rule 1, Rule 2, and Rule 3 when $k = 2$	337
C.3.	Additional Fully Simulated Weibull Study Trace Plots	340
C.3.1.	Additional Random-Walk Trace Plots for the Fully Simulated Weibull Study	340
C.3.2.	Additional Independence-Sampler Trace Plots for the Fully Simulated Weibull Study	343
C.4.	Additional Fully Simulated Study Autocorrelation Functions	347
C.5.	Additional Fully Simulated Study Posterior Predictive Check Plots	350
C.6.	Additional Semi-Simulated Alloy Study Trace Plots	354
C.6.1.	Additional Random-Walk Trace Plots for the Semi-Simulated Alloy Study	354

TABLE OF CONTENTS—Continued

C.6.2.	Additional Independence-Sampler Trace Plots for the Semi-Simulated Alloy Study	357
C.7.	Additional Semi-Simulated Alloy Study Autocorrelation Functions	361
C.8.	Additional Semi-Simulated Alloy Study Posterior Predictive Check Plots	364
REFERENCES		368

LIST OF TABLES

Table 3.1	P^* and d Values, for $k = 10, n = 25$	33
Table 3.2	P^* and b Values, for $k = 10, n = 25$	33
Table 3.3	Minimums and Means for Each of the Populations, for $k = 10, n = 25$	34
Table 3.4	P^* and the Selected Populations Using Rules R_1 and R_2	34
Table 3.5	Sample Size n and the Corresponding P^* Values, $k = 5$	36
Table 4.1	OCs for R_1 and R_2 Under the Slippage Configuration	41
Table 4.2	OCs for R_1 and R_2 Under the Equi-spaced Configuration	43
Table 4.3	Moments of Sample Statistics Underlying Selection Rules R_1 and R_2	45
Table 4.4	Comparing the OCs for R_2 and R_3 . Simulation (blue), Central Limit Theorem (CLT) (black), and Gamma Distribution (purple)	46
Table 4.5	OCs for R_1 and R_2 Under the Slippage Configuration	48
Table 4.6	OCs for R_1 and R_2 Under the Equi-spaced Configuration	49
Table 5.1	OCs for R_1 and R_2 for Two Populations, $n = 25$	55
Table 5.2	OCs for R_1 and R_2 for Two Populations, $n = 10$	56
Table 5.3	OCs for R_1 and R_2 for Two Populations, $n = 50$	56

LIST OF TABLES—Continued

Table 5.4	Selected Probability of Correct Selection (PCS)-violation rows for R_1 versus the CLT-based rule R_2 , $n = 5$.	60
Table 5.5	Selected ESS-violation rows for R_1 versus the Gamma-based rule R_3 , $n = 100$, $P^* = 0.95$.	68
Table 5.6	Cases with PCS violations for R_1 versus the CLT-based rule R_2 under the strict tolerance 10^{-10} .	71
Table 5.7	Selected PCS-violation rows for R_1 vs R_2 , $n = 5$, $P^* = 0.99$, $\text{tol} = 10^{-5}$.	72
Table 6.1	Generating parameters for the five Weibull populations	95
Table 6.2	Weibull percentiles computed from the generating parameters	95
Table 6.3	Posterior means of γ , η , and β for the fully simulated study	98
Table 6.4	Posterior medians of γ , η , and β for the fully simulated study.	100
Table 6.5	Posterior estimates compared with true parameter values.	101
Table 6.6	Posterior means and 95% credible intervals for Weibull parameters.	103
Table 6.7	Posterior means and 95% credible intervals for Weibull percentiles.	103
Table 6.8	Posterior probabilities of having the largest parameter value.	108
Table 6.9	Posterior means of Weibull percentiles.	110

LIST OF TABLES—Continued

Table 6.10	Posterior probabilities of having the largest Weibull percentile.	111
Table 6.11	Joint agreement between threshold-based and percentile-based selection.	111
Table 6.12	Acceptance rates across chains.	117
Table 6.13	Gelman–Rubin $\hat{R}$ diagnostics for the transformed Weibull parameters.	119
Table 6.14	Effective sample sizes for Weibull parameters.	120
Table 6.15	Effective sample sizes for Weibull percentiles.	121
Table 6.16	Prior sensitivity analysis for the prior standard deviation of $\log(\delta)$.	123
Table 6.17	Posterior predictive checks for the fully simulated study.	126
Table 6.18	Maximum likelihood estimates for the three-parameter Weibull fit to the alloy fatigue data.	131
Table 6.19	Fitted Weibull quantiles from the maximum likelihood fit.	132
Table 6.20	Bayesian posterior summary for the three-parameter Weibull fit to the alloy fatigue data.	142
Table 6.21	Maximum likelihood and Bayesian posterior comparison for the alloy fatigue data.	143
Table 6.22	Parameters used for the semi-simulated alloy fatigue study.	149
Table 6.23	Weibull percentiles computed from the parameters used for the semi-simulated alloy fatigue study.	149

LIST OF TABLES—Continued

Table 6.24	Censoring summary for the semi-simulated alloy fatigue study.	150
Table 6.25	Posterior summaries for Weibull parameters in the semi-simulated alloy fatigue study.	151
Table 6.26	Posterior probabilities of having the largest parameter value in the semi-simulated alloy fatigue study.	159
Table 6.27	Posterior selection probabilities and cumulative probabilities for γ in the semi-simulated alloy fatigue study.	159
Table 6.28	Posterior summaries for Weibull percentiles in the semi-simulated alloy fatigue study.	162
Table 6.29	Ranked posterior selection probabilities and cumulative probabilities for largest Weibull percentiles in the semi-simulated alloy fatigue study.	163
Table 6.30	Joint agreement between threshold-based and percentile-based selection in the semi-simulated alloy fatigue study.	165
Table 6.31	Acceptance rates across chains for the semi-simulated alloy fatigue study.	170
Table 6.32	Gelman–Rubin $\hat{R}$ diagnostics for the transformed Weibull parameters in the semi-simulated alloy fatigue study.	172
Table 6.33	Effective sample sizes for Weibull parameters in the semi-simulated alloy fatigue study.	173
Table 6.34	Effective sample sizes for Weibull percentiles in the semi-simulated alloy fatigue study.	175

LIST OF TABLES—Continued

Table 6.35	Prior sensitivity analysis for the prior standard deviation of $\log(\delta)$ in the semi-simulated alloy fatigue study.	177
Table 6.36	Posterior predictive checks for the semi-simulated alloy fatigue study.	179
Table C.1	Inequality-check table for R_1 versus the CLT-based rule R_2 , $n = 5$, $P^* = 0.95$.	325
Table C.2	Inequality-check table for R_1 versus the CLT-based rule R_2 , $n = 5$, $P^* = 0.99$.	326
Table C.3	Inequality-check table for R_1 versus the Gamma-based rule R_3 , $n = 5$, $P^* = 0.95$.	327
Table C.4	Inequality-check table for R_1 versus the Gamma-based rule R_3 , $n = 5$, $P^* = 0.99$.	328
Table C.5	Inequality-check table for R_1 versus the CLT-based rule R_2 , $n = 25$, $P^* = 0.95$.	329
Table C.6	Inequality-check table for R_1 versus the CLT-based rule R_2 , $n = 25$, $P^* = 0.99$.	330
Table C.7	Inequality-check table for R_1 versus the Gamma-based rule R_3 , $n = 25$, $P^* = 0.95$.	331
Table C.8	Inequality-check table for R_1 versus the Gamma-based rule R_3 , $n = 25$, $P^* = 0.99$.	332
Table C.9	Inequality-check table for R_1 versus the CLT-based rule R_2 , $n = 100$, $P^* = 0.95$.	333
Table C.10	Inequality-check table for R_1 versus the CLT-based rule R_2 , $n = 100$, $P^* = 0.99$.	334

LIST OF TABLES—Continued

Table C.11	Inequality-check table for R_1 versus the Gamma-based rule R_3 , $n = 100$, $P^* = 0.95$, using the Gamma-quantile upper integration limit.	335
Table C.12	Inequality-check table for R_1 versus the Gamma-based rule R_3 , $n = 100$, $P^* = 0.99$.	336

LIST OF FIGURES

Figure 2.1	Effect of Weibull shape β and threshold γ on reliability behavior.	18
Figure 4.1	ESS with data from Table 4.1 and Table 4.2. $k = 10, n = 25, P^* = 0.95, N = 200,000$	44
Figure 5.1	Difference plots for R_1 versus R_2 $k = 2, n = 5$, $P^* = 0.95$ and $P^* = 0.99$	61
Figure 5.2	Difference plots for R_1 versus R_3 , $k = 2, n = 5$, $P^* = 0.95, P^* = 0.99$.	62
Figure 5.3	Difference plots for R_1 versus R_2 , $k = 2, n = 25$, $P^* = 0.95$, and $P^* = 0.99$.	64
Figure 5.4	Difference plots for R_1 versus R_3 , $k = 2, n = 25$, $P^* = 0.95$, and $P^* = 0.99$.	65
Figure 5.5	Difference plots for R_1 versus R_2 , $k = 2, n = 100$, $P^* = 0.95$, and $P^* = 0.99$.	67
Figure 5.6	Difference plots for R_1 versus R_3 , $k = 2, n = 100$, $P^* = 0.95$, and $P^* = 0.99$.	69
Figure 6.1	Posterior densities of γ, η , and β for Population P1.	104
Figure 6.2	Posterior densities of γ, η , and β for Population P2.	105
Figure 6.3	Posterior densities of γ, η , and β for Population P3.	105

LIST OF FIGURES—Continued

Figure 6.4	Posterior densities of γ , η , and β for Population P4.	106
Figure 6.5	Posterior densities of γ , η , and β for Population P5.	106
Figure 6.6	Random-walk trace plots for Population P1	113
Figure 6.7	Random-walk trace plots for Population P5	114
Figure 6.8	Independent-sampler trace plots for Population P5	115
Figure 6.9	Autocorrelation functions for γ , η , and β for Population P5	116
Figure 6.10	Prior sensitivity analysis for the fully simulated Weibull study. Each panel shows the posterior mean and 95% credible interval for γ under different prior standard deviations for $\log(\delta)$.	124
Figure 6.11	Representative posterior predictive checks for Population P5	129
Figure 6.12	Histogram of observed Alloy T7987 failure times with the fitted three-parameter Weibull density.	133
Figure 6.13	Empirical distribution function of the Alloy T7987 failure times with the fitted three-parameter Weibull distribution function.	134
Figure 6.14	Kaplan–Meier survival estimate for the Alloy T7987 data with the fitted three-parameter Weibull survival curve.	136
Figure 6.15	Fitted three-parameter Weibull hazard function for the Alloy T7987 fatigue-life data.	137

LIST OF FIGURES—Continued

Figure 6.16	Weibull probability plot for the Alloy T7987 data with the fitted three-parameter Weibull line.	141
Figure 6.17	Posterior density plots for γ , η , β , and δ from the Bayesian three-parameter Weibull fit to the Alloy T7987 fatigue-life data.	145
Figure 6.18	Traceplots for γ , η , β , and δ from the Bayesian three-parameter Weibull fit to the Alloy T7987 fatigue-life data.	147
Figure 6.19	Posterior densities of γ , η , and β for P1 in the semi-simulated alloy fatigue study.	154
Figure 6.20	Posterior densities of γ , η , and β for P2 in the semi-simulated alloy fatigue study.	155
Figure 6.21	Posterior densities of γ , η , and β for P3 in the semi-simulated alloy fatigue study.	156
Figure 6.22	Posterior densities of γ , η , and β for P4 in the semi-simulated alloy fatigue study.	157
Figure 6.23	Posterior densities of γ , η , and β for P5 in the semi-simulated alloy fatigue study.	158
Figure 6.24	Random-walk trace plots for Population P1	167
Figure 6.25	Random-walk trace plots for Population P5	168
Figure 6.26	Independence-sampler trace plots for Population P5	169
Figure 6.27	Autocorrelation functions for γ , η , and β for Population P5	170

LIST OF FIGURES—Continued

Figure 6.28	Prior sensitivity analysis for the semi-simulated alloy fatigue study. Each panel shows the posterior mean and 95% credible interval for γ under different prior standard deviations for $\log(\delta)$.	178
Figure 6.29	Representative posterior predictive checks for Population P5	183
Figure C.1	Heat Map of $\Delta_{PCS,2}(\delta)$ for R_1 versus the CLT-based rule R_2 , with $P^* = 0.99$.	337
Figure C.2	Heat Map of $\Delta_{PICS,2}(\delta)$ for R_1 versus the CLT-based rule R_2 , with $P^* = 0.99$.	337
Figure C.3	Heat Map of $\Delta_{ESS,2}(\delta)$ for R_1 versus the CLT-based rule R_2 , with $P^* = 0.99$.	338
Figure C.4	Heat Map of $\Delta_{PCS,3}(\delta)$ for R_1 versus the Gamma-based rule R_3 , with $P^* = 0.99$.	338
Figure C.5	Heat Map of $\Delta_{PICS,3}(\delta)$ for R_1 versus the Gamma-based rule R_3 , with $P^* = 0.99$.	339
Figure C.6	Heat Map of $\Delta_{ESS,3}(\delta)$ for R_1 versus the Gamma-based rule R_3 , with $P^* = 0.99$.	339
Figure C.7	Random-walk trace plots for Population P2	340
Figure C.8	Random-walk trace plots for Population P3	341
Figure C.9	Random-walk trace plots for Population P4	342
Figure C.10	Independence-sampler trace plots for Population P1	343
Figure C.11	Independence-sampler trace plots for Population P2	344

LIST OF FIGURES—Continued

Figure C.12	Independence-sampler trace plots for Population P3	345
Figure C.13	Independence-sampler trace plots for Population P4	346
Figure C.14	Autocorrelation functions for γ , η , and β for Population P1	347
Figure C.15	Autocorrelation functions for γ , η , and β for Population P2	348
Figure C.16	Autocorrelation functions for γ , η , and β for Population P3	348
Figure C.17	Autocorrelation functions for γ , η , and β for Population P4	349
Figure C.18	Posterior predictive checks for Population P1	350
Figure C.19	Posterior predictive checks for Population P2	351
Figure C.20	Posterior predictive checks for Population P3	352
Figure C.21	Posterior predictive checks for Population P4	353
Figure C.22	Random-walk trace plots for Population P2	354
Figure C.23	Random-walk trace plots for Population P3	355
Figure C.24	Random-walk trace plots for Population P4	356
Figure C.25	Independence-sampler trace plots for Population P1	357
Figure C.26	Independence-sampler trace plots for Population P2	358

LIST OF FIGURES—Continued

Figure C.27	Independence-sampler trace plots for Population P3	359
Figure C.28	Independence-sampler trace plots for Population P4	360
Figure C.29	Autocorrelation functions for γ , η , and β for Population P1	361
Figure C.30	Autocorrelation functions for γ , η , and β for Population P2	362
Figure C.31	Autocorrelation functions for γ , η , and β for Population P3	362
Figure C.32	Autocorrelation functions for γ , η , and β for Population P4	363
Figure C.33	Posterior predictive checks for Population P1	364
Figure C.34	Posterior predictive checks for Population P2	365
Figure C.35	Posterior predictive checks for Population P3	366
Figure C.36	Posterior predictive checks for Population P4.	367

LIST OF ABBREVIATIONS

ACF	Autocorrelation Function
CDF	Cumulative Distribution Function
CLT	Central Limit Theorem
CS	Correct Selection
ESS	Expected Subset Size
ICS	Incorrect Selection
iid	Independent and Identically Distributed
IZ	Indifference Zone
LFC	Least Favorable Configuration
MCMC	Markov Chain Monte Carlo
MLE	Maximum Likelihood Estimation
OC	Operating Characteristic
OCs	Operating Characteristics
PCS	Probability of Correct Selection
PDF	Probability Density Function
PICS	Probability of Incorrect Selection

CHAPTER 1

INTRODUCTION

1.1 Motivation and Overview

In many scientific, industrial, and biomedical settings, practitioners must compare multiple populations or systems and determine which one is best according to a performance criterion. Examples include selecting a manufacturing process with superior component lifetime, comparing treatment groups in time-to-event studies, and evaluating competing system designs where early failures are especially costly. In such problems, available data are often limited and may be censored, so uncertainty about population-specific performance can remain substantial even after sampling.

A natural objective in these settings is to identify a single best population. However, when populations are similar or sample sizes are moderate, procedures that attempt to declare a single winner can be unstable. This motivates subset selection, where the decision output is a set of promising populations rather than a single choice. Subset selection procedures are designed to produce a subset $S \subset \{1, \dots, k\}$ that contains the best population with probability of at least a prespecified value P^* , while also seeking to control the size of the subset for practical usefulness. This objective differs from the IZ framework, which targets correct selection of the single best population under the condition that the best is separated from the others by at least a practically meaningful amount. The subset-selection and IZ perspectives provide complementary decision tools

and highlight the fundamental tradeoffs between strong probability guarantees and efficiency measures such as expected subset size and required sample sizes.

This dissertation studies subset selection procedures motivated by reliability modeling, where the performance criterion is naturally linked to the lower tail of a lifetime distribution. For exponential and Weibull lifetime models, a threshold (location) parameter γ_i can be interpreted as a minimum lifetime prior to which failures are not expected under the model. In applications where early failures have high cost, selecting the population with the largest threshold parameter is therefore a meaningful decision goal. The emphasis on threshold parameters also motivates procedures based on statistics sensitive to early-failure behavior and requires careful evaluation of operating characteristics across parameter configurations.

The dissertation consists of two main components. The first component develops and studies frequentist subset selection procedures for exponential populations with threshold parameters. Selection rules based on sample minima and sample means are considered, along with an IZ-based procedure for comparison. For these procedures, calibration constants are derived to meet probability guarantees, and operating characteristics such as the probability of correct selection and the expected subset size are evaluated under standard configuration settings (including slippage and equally spaced configurations). Exact results and comparisons for the two-population case are also developed, providing additional insight and serving as a benchmark for numerical findings.

The second component develops Bayesian subset selection procedures for Weibull and exponential populations. Bayesian methods naturally quantify uncertainty through the posterior distribution of the parameters and enable subset construction based on posterior probabilities of being best. Using posterior simulation, this dissertation constructs subsets

that aim to achieve a target inclusion probability for the best population and compares Bayesian procedures to frequentist rules in terms of operating characteristics and practical performance. Together, these components provide a unified study of subset selection for threshold-driven reliability criteria under both frequentist and Bayesian paradigms.

1.2 Dissertation Contributions

This dissertation contributes to ranking-and-selection methodology for reliability populations by developing, analyzing, and comparing subset selection procedures for threshold-based lifetime models. The main contributions are as follows.

First, the dissertation formulates a threshold-parameter subset selection problem for reliability populations. In this formulation, the best population is defined as the population with the largest threshold parameter, γ_i , which represents a lower endpoint or minimum lifetime under the assumed lifetime model. This provides a reliability-motivated selection criterion that is especially relevant when early failures are costly and when the lower tail of the lifetime distribution is of primary interest.

Second, the dissertation develops frequentist subset selection procedures for k independent exponential threshold populations with a common known scale parameter. Two main subset selection rules are considered: a minimum-based rule, R_1 , and a mean-based rule, R_2 . The rule R_1 uses the sample minimum from each population and is directly connected to the threshold structure of the exponential model. The rule R_2 uses the sample mean and is motivated by the fact that, under a common scale parameter, the population means are ordered in the same way as the threshold parameters. A gamma-based formulation, R_3 , is also introduced for the mean-based procedure in order to evaluate finite-sample behavior without relying only on the central limit theorem

approximation. The calibration constants for these rules are chosen so that the selected subset contains the best population with probability at least P^* .

Third, the dissertation develops an indifference-zone procedure for the same exponential threshold setting. While the subset selection procedures return a set of promising populations, the indifference-zone procedure targets selection of a single best population when the best threshold parameter is separated from the others by at least a practically meaningful amount. This provides a complementary frequentist framework and allows comparison between subset-based selection and single-best selection through the role of the probability requirement and the required sample size.

Fourth, the dissertation evaluates the operating characteristics of the proposed frequentist rules under standard threshold configurations. The operating characteristics include the probability of correct selection, the probability of incorrect selection, and the expected subset size. These quantities are studied under slippage and equi-spaced configurations, allowing the procedures to be compared across both difficult and more separated parameter settings. The numerical results show that procedures calibrated to the same probability of correct selection requirement can behave quite differently in terms of expected subset size. In particular, the minimum-based rule generally produces smaller selected subsets than the mean-based rule once the best population is separated from the competing populations.

Fifth, the dissertation derives exact operating characteristics for the two-population case. For $k = 2$, closed-form or exact integral expressions are obtained for the probability of correct selection, the probability of incorrect selection, and the expected subset size for the minimum-based rule, the CLT based mean rule, and the gamma-based mean rule. These exact results clarify the finite-sample behavior of the procedures and provide a benchmark for the numerical comparisons. They also show that the main practical

distinction among the rules is how quickly each procedure removes the inferior population from the selected subset as the threshold separation increases.

Sixth, the dissertation develops a Bayesian subset selection procedure for independent three-parameter Weibull populations. In the Bayesian framework, posterior simulation is used to estimate the probability that each population has the largest threshold parameter. The selected subset is then constructed as the smallest subset whose cumulative posterior probability of containing the best population is at least the target value P^* . This provides a post-data probability interpretation of subset selection and extends the threshold-based selection framework to a more flexible Weibull reliability model.

Seventh, the Bayesian procedure is applied to both fully simulated and semi-simulated reliability studies. The fully simulated five-population Weibull study provides a controlled setting in which the true best population is known. The semi-simulated alloy fatigue study incorporates right-censored lifetime data and demonstrates how the Bayesian selected subset changes when posterior uncertainty is larger. In addition to threshold-based selection, posterior comparisons of Weibull percentiles are used to provide supplementary reliability interpretations. These examples illustrate how Bayesian subset selection can balance decisiveness and uncertainty in both complete-data and censored-data reliability settings.

Together, these contributions provide a unified study of threshold-based subset selection under frequentist and Bayesian frameworks. The frequentist component emphasizes calibrated probability guarantees and operating characteristics, while the Bayesian component emphasizes posterior uncertainty and probability-based subset construction. Both perspectives support the broader goal of selecting reliable populations while accounting for uncertainty, finite-sample behavior, and practical subset size.

1.3 Organization of the Dissertation

The remainder of this dissertation is organized as follows.

Chapter 2 provides the background and problem formulation. It reviews subset selection procedures, indifference-zone procedures, and operating characteristics used to evaluate ranking-and-selection rules. The chapter also introduces the reliability modeling framework used throughout the dissertation, with emphasis on exponential and Weibull lifetime models containing a threshold parameter. The threshold parameter is identified as the primary performance measure because it represents a lower endpoint or minimum lifetime under the model.

Chapter 3 develops frequentist selection rules for exponential threshold populations. The chapter defines the model for k independent exponential populations with a common known scale parameter and threshold parameters $\gamma_1, \dots, \gamma_k$. It then develops the minimum-based subset selection rule R_1 , the mean-based subset selection rule R_2 , and the gamma-based formulation R_3 . Calibration of the constants used in these procedures is discussed, and an indifference-zone procedure is introduced for selecting a single best population when a specified separation condition holds. Numerical illustrations show how the procedures are implemented and how the selected subsets depend on the target probability level.

Chapter 4 studies the operating characteristics of the frequentist subset selection rules. The chapter defines and evaluates the probability of correct selection, the probability of incorrect selection, and the expected subset size. These operating characteristics are examined under slippage and equi-spaced threshold configurations. The comparisons show how the rules behave as the separation between populations increases and demonstrate that the minimum-based rule often yields smaller expected subset sizes than the mean-based rule while maintaining the required probability guarantee.

Chapter 5 specializes the analysis to the two-population case. In this setting, exact operating characteristics are derived for the minimum-based rule R_1 , the CLT-based mean rule R_2 , and the gamma-based mean rule R_3 . The chapter presents explicit formulae or exact integral expressions for the probability of correct selection, the probability of incorrect selection, and the expected subset size as functions of the separation parameter $\delta = \gamma_{[2]} - \gamma_{[1]}$. Numerical comparisons and inequality checks are then used to study how the rules differ across sample sizes and target probability levels.

Chapter 6 develops the Bayesian subset selection framework for three-parameter Weibull populations. The chapter defines the Bayesian posterior probability that each population is best, constructs the Bayesian selected subset using cumulative posterior probabilities, and presents the Weibull model for exact and right-censored lifetime data. Prior specification, posterior computation using Metropolis–Hastings simulation, convergence diagnostics, prior sensitivity analysis, posterior predictive checks, and percentile-based reliability comparisons are also discussed. The method is applied to a fully simulated five-population Weibull study and to a semi-simulated censored study based on the Alloy T7987 fatigue-life data. The chapter concludes by summarizing the main Bayesian results, practical takeaways, and possible extensions of the threshold-based subset selection framework.

Chapter 7 summarizes the main findings, practical implications, limitations, and directions for future work.

CHAPTER 2

BACKGROUND AND PROBLEM FORMULATION

2.1 Subset Selection Procedures

Subset selection studies procedures for comparing multiple populations and selecting the best (or a set containing the best) with a prescribed probability guarantee, see [1, 2, 3]. Early work formalized this as a multiple-decision problem with guarantees stated in terms of the PCS [1, 4]. Subset selection procedures are particularly attractive when sample sizes are limited or populations are close in performance, because selecting a small subset can be more stable than declaring a single winner while still meeting a user-specified PCS target.

2.1.1 Subset Selection Objective

Let T_i denote the performance measure for population i , and define the best population as

$$i^* = \arg \max_{1 \leq i \leq k} T_i.$$

A subset selection procedure returns a random subset $S \subset \{1, \dots, k\}$. The main requirement studied in this dissertation is

$$\Pr(i^* \in S) \geq P^*,$$

where P^* is prespecified (e.g., 0.90 or 0.95). In addition to meeting the probability requirement, a useful procedure should keep the average subset size small.

2.1.2 Why Subset Selection Rather Than Single-Best Selection

When populations are similar, a single-best decision can be sensitive to sampling variability. Subset selection offers a compromise: it yields an informative decision by narrowing the candidate set while maintaining a probability guarantee of including the best population. This perspective is particularly relevant in reliability settings, where data are costly to obtain, and uncertainty remains substantial even with careful sampling designs.

2.2 Notation and Problem Setup

Consider k independent populations indexed by $i = 1, \dots, k$. From population i we observe a random sample $X_{i1}, \dots, X_{in_i}$, where n_i denotes the sample size (often $n_i = n$ for all i). Let θ_i denote the parameter vector for population i under the assumed lifetime model.

The focus of this dissertation is on Exponential and Weibull families used in reliability analysis. In settings where a threshold (location) parameter is included, we denote it by γ_i and interpret it as a minimum lifetime prior to which failures are not expected under the model. In that case, the target performance measure is taken to be $T_i = \gamma_i$, so the best population is the one with the largest threshold parameter. Other model parameters (e.g., scale and shape) are introduced as needed in later chapters.

Throughout the dissertation, we evaluate procedures using PCS-type criteria and additional operating characteristics (defined in Section 4.1). We also consider standard parameter configurations used in the ranking-and-selection literature to assess procedure performance under both favorable and difficult scenarios.

2.3 Classical Subset Selection Procedures

A central frequentist approach to subset selection guarantees that the selected subset contains the best population with probability at least P^* , uniformly over a broad class of parameter configurations. Gupta's subset-selection formulation in [1] is a key reference point and led to many extensions and performance analyses, including attention to least favorable configurations and expected subset size, see also Gibbons et al. [3].

2.3.1 Least Favorable Configurations and Calibration

To obtain a uniform PCS guarantee, many procedures calibrate critical constants under configurations that minimize PCS subject to the model assumptions, see Gupta [1, 2] and Gibbons et al. [3]. This calibration ensures that, regardless of the true parameter values, the probability requirement is met. However, such worst-case calibration can lead to conservative behavior, motivating comparisons of operating characteristics across representative configurations.

In more formal terms, let Θ denote the collection of parameter configurations under which the guarantee is desired. A least favorable configuration Least Favorable Configuration (LFC) for a procedure is a configuration $\theta_{\text{LFC}} \in \Theta$ at which the PCS is minimized, i.e.,

$$\Pr_{\theta_{\text{LFC}}}(i^* \in S) = \inf_{\theta \in \Theta} \Pr_{\theta}(i^* \in S).$$

Calibrating critical constants using θ_{LFC} yields a uniform guarantee of the form $\inf_{\theta \in \Theta} \Pr_{\theta}(i^* \in S) \geq P^*$. Intuitively, the LFC corresponds to a "hard" scenario in which the best population is only marginally better than its competitors, making the data least informative for discrimination. As a consequence, calibration at the LFC can be conservative for configurations in which the best population is more clearly separated,

motivating the evaluation of operating characteristics such as PCS, and ESS, [3] across a range of representative configurations.

2.3.2 Procedure Families

Classical procedures include single-stage rules based on summary statistics and multi-stage or sequential rules that adapt sampling effort. The procedures studied in this dissertation emphasize rules that are simple to implement and calibrate, while still allowing meaningful comparisons through PCS and expected subset size.

2.4 Indifference Zone Procedures

IZ procedures, originating in the classical ranking-and-selection literature, address a different but related goal: guaranteeing correct selection when the best population is separated from the rest by at least a practically meaningful margin [4]. The IZ framework introduces a separation parameter, often denoted by δ , and motivates procedures designed to achieve a PCS guarantee when the IZ condition holds.

2.4.1 Indifference Zone Condition and Sample Size Logic

Under an IZ formulation, the guarantee is required for parameter configurations in which the best population exceeds all others by at least δ (under the chosen performance measure). Sample size determination (or two-stage sampling) is then chosen so that PCS meets or exceeds P^* in the least favorable configuration within the IZ set. This leads to procedures that can be conservative when δ is small or when populations are difficult to distinguish, which motivates subset selection as an alternative decision objective.

2.4.2 Multi-stage and Sequential Methods

Sequential and multi-stage procedures have been studied extensively as a way to reduce sampling effort while maintaining PCS guarantees, see [5, 6]. These methods highlight the tradeoff between conservatism (strong guarantees) and efficiency (sample size, subset size, and computational complexity). In this dissertation, IZ ideas serve primarily as a benchmark for comparison against subset selection rules targeting threshold parameters.

2.5 Operating Characteristics and Performance Metrics

To compare ranking-and-selection procedures, we use operating characteristics (OCs) that quantify both probability guarantees and practical efficiency. For subset selection procedures that return a random subset $S \subset \{1, \dots, k\}$, the primary measure is the probability of including the best population,

$$\text{PCS} = \Pr(i^* \in S),$$

which is compared to the target requirement P^* . A second key Operating Characteristic (OC) is the expected subset size, ESS, which reflects the average decisiveness of the procedure: smaller values correspond to more informative selections.

Additional summaries used in the literature include the distribution of $|S|$, which is the size of the selected subset, and the probability of selecting exactly one population, $\Pr(|S| = 1)$, as well as the complement $1 - \Pr(i^* \in S)$, which represents the probability of failing to include the best population. Since OCs depend on the underlying parameter configuration, it is common to evaluate procedures under standard configuration settings such as slippage-type configurations (one population is best and the remaining populations are equal) and equally spaced configurations (performance measures differ by fixed

increments). Detailed definitions of these configurations and comparative OC results for the procedures studied in this dissertation are presented in Chapter 4.

2.6 Bayesian Ranking and Selection and Posterior-Based Subset Rules

Bayesian approaches to ranking and selection represent uncertainty about unknown population performance through the posterior distribution and translate that uncertainty into selection decisions. Let T_i denote the target measure for population i (e.g., a threshold parameter or a reliability functional) and define the best population index by $i^* = \arg \max_{1 \leq i \leq k} T_i$. In this dissertation, the primary target is the threshold parameter $T_i = \gamma_i$, motivated by early-failure protection in reliability applications. A natural Bayesian summary is the posterior probability that population i is best,

$$p_i = \Pr(i = i^* \mid \text{data}),$$

which can be approximated using posterior simulation by the proportion of posterior draws in which population i attains the largest value of T_i . Posterior-based summaries such as $\{p_i\}_{i=1}^k$ provide an interpretable ranking of populations and support direct probability statements about the selection decision.

Bayesian selection rules are often motivated by decision theory: the selected action (a single index or a subset) is chosen to minimize posterior expected loss (equivalently, posterior risk). Under this view, different loss functions lead to different selection rules. For example, a loss that heavily penalizes failing to include the best population leads naturally to subset-selection rules that control the posterior probability of containing the best. Conversely, a loss that penalizes large subset sizes more strongly produces smaller subsets and emphasizes decisiveness. Bayesian ranking-and-selection has also been studied in the context of sampling design and allocation, where one chooses

how to distribute sampling effort across populations to reduce posterior uncertainty and improve downstream selection performance.

For Bayesian subset selection, one widely used construction is to select the smallest subset whose cumulative posterior probability of containing the best meets or exceeds a target value. A typical implementation ranks populations by p_i and chooses the smallest m such that the sum of the m largest posterior probabilities is at least P^* , see [7, 8, 9].

Let $p_{(1)} \geq p_{(2)} \geq \cdots \geq p_{(k)}$ denote the posterior probabilities sorted in decreasing order, and

$$m = \min \left\{ r \in \{1, \dots, k\} : \sum_{\ell=1}^r p_{(\ell)} \geq P^* \right\}.$$

The selected subset consists of the populations,

$$S = \{(1), (2), \dots, (m)\}.$$

Variants include selecting all populations with p_i above a fixed cutoff, or selecting populations whose posterior credible intervals for T_i remain competitive relative to the best. These posterior-based subset rules are attractive because they can be computed directly from posterior draws even when analytic expressions are unavailable.

Bayesian formulations are particularly relevant in reliability and life-data settings. Censoring complicates classical ranking-and-selection because the information content differs across populations and the sampling distributions of key statistics may be difficult to characterize; review work has discussed ranking and subset selection procedures under censored sampling schemes, see [10]. In Weibull models with threshold (location) parameters, likelihood irregularity and sensitivity near the endpoint can further challenge purely analytic approaches, making simulation-based Bayesian procedures appealing.

Recent Bayesian subset selection developments also include nonparametric or semi-parametric constructions with Weibull components, which provide additional flexibility for modeling heterogeneous populations.

The Bayesian component of this dissertation adopts posterior simulation to construct subsets based on posterior probabilities of being best under Weibull/Exponential models and compares these Bayesian rules to frequentist procedures in terms of inclusion probability and subset size behavior.

2.7 Weibull and Exponential Models in Reliability

The Weibull distribution is among the most widely used models in reliability and life-data analysis because it provides a flexible, interpretable family for describing a broad range of failure-time behaviors [11, 12, 13]. In its three-parameter form, the Weibull model is commonly written with a threshold (or location/shift) parameter γ , a scale parameter $\eta > 0$, and a shape parameter $\beta > 0$ (see, e.g., [13, 14]). In this parameterization, γ represents a minimum lifetime (or delayed start) so that failures are modeled only for times $x \geq \gamma$, making γ a natural quantity for describing early-failure protection or a guaranteed minimum performance level [11, 13].

The density of the three-parameter Weibull model can be written as

$$f(x | \gamma, \eta, \beta) = \frac{\beta}{\eta} \left(\frac{x - \gamma}{\eta} \right)^{\beta-1} \exp \left[- \left(\frac{x - \gamma}{\eta} \right)^{\beta} \right], \quad \{x \geq \gamma\}$$

with survival function

$$S(x | \gamma, \eta, \beta) = \begin{cases} 1, & x < \gamma, \\ \exp \left[- \left(\frac{x - \gamma}{\eta} \right)^{\beta} \right], & x \geq \gamma. \end{cases}$$

The two-parameter Weibull model is the special case when the threshold parameter $\gamma = 0$.

A major practical advantage of the Weibull model is that its hazard function can capture decreasing, constant, or increasing failure rates depending on the value of the shape parameter β [11, 13, 14]. When $0 < \beta < 1$, the hazard rate decreases over time, which is often used to represent "infant mortality" behavior where early failures are more likely due to manufacturing defects. When $\beta = 1$, the hazard is constant and the model reduces to the exponential distribution, corresponding to a memoryless failure mechanism with a constant failure rate [11, 13]. When $\beta > 1$, the hazard increases, representing aging and wear-out phenomena in mechanical systems, where the chance of failure grows as components accumulate wear [11, 13]. These qualitative regimes make the Weibull family a convenient default model in reliability applications, where the observed failure mechanism may vary across products, environments, or stress levels [12, 13].

Percentiles of the Weibull distribution are also widely used as engineering summaries, since they identify the time by which a specified proportion of units are expected to fail [11, 13]. In particular, in the two-parameter Weibull model (with $\gamma = 0$), the scale parameter η equals the 63.2% percentile, and more generally percentiles can be expressed in closed form as functions of (γ, η, β) [13, 14]. This is one reason why Weibull-based criteria often align naturally with practical reliability goals such as warranty design, maintenance scheduling, and comparison of competing designs at specific reliability targets [12, 13].

The Exponential model arises as the special case $\beta = 1$ of the three-parameter Weibull distribution and is frequently used when a constant failure rate is a reasonable approximation [11, 13]. In the exponential threshold setting, the lifetime distribution has both a scale parameter η and a threshold parameter γ , and the threshold enters additively into the mean and into percentiles. Consequently, when the scale is common across populations, ranking populations by the threshold parameter is equivalent to ranking by

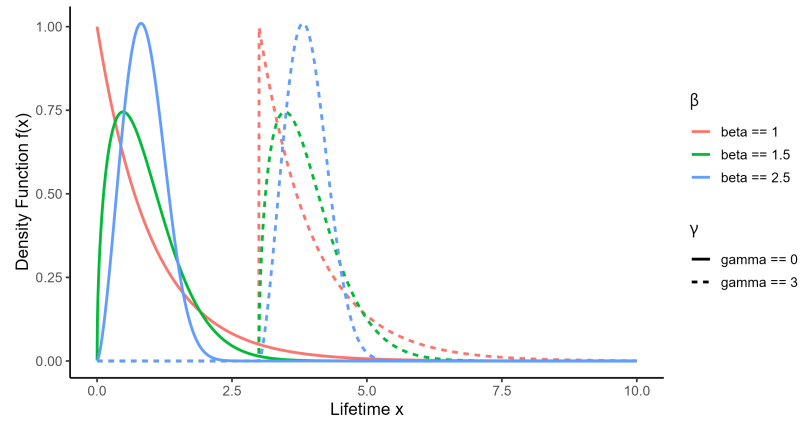
the mean or by any fixed percentile, making γ a particularly interpretable target for selection in early-failure-focused applications [11, 13].

From an inferential standpoint, incorporating a threshold parameter can introduce additional challenges. In three-parameter Weibull models, estimation of the endpoint/threshold can exhibit non-regular behavior, and likelihood-based methods can be sensitive near the boundary implied by γ (since the support depends on γ) [11, 13]. In reliability studies, these challenges are often compounded by censoring, where some units have not failed by the end of observation, so the data contain incomplete information about the lower tail and about early-failure behavior [11, 12]. These features motivate careful procedure design in the frequentist setting and make Bayesian approaches attractive, since posterior modeling and simulation propagate uncertainty in γ (and other parameters) in a principled way under incomplete observation.

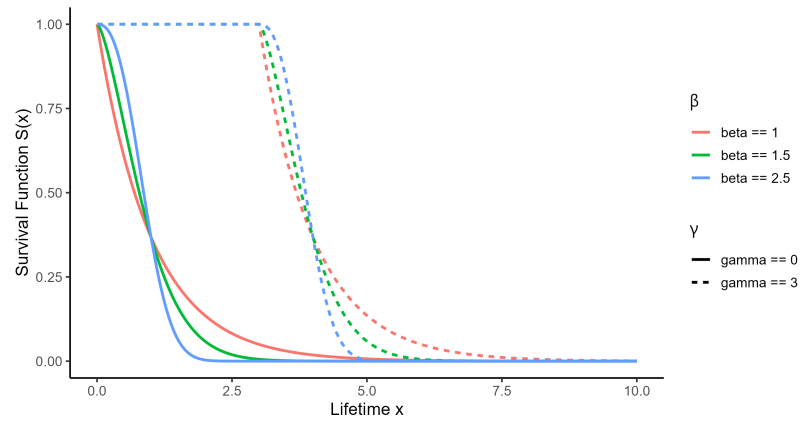
In this dissertation, these reliability considerations motivate defining the "best" population in terms of early-failure protection and focusing on selection procedures that target threshold (location) parameters in exponential and Weibull models.

2.8 Concluding Remarks

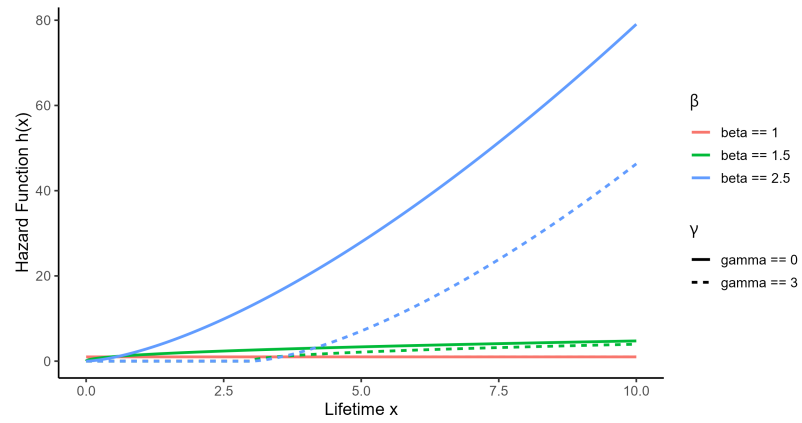
Existing R&S literature provides a strong foundation for PCS-guaranteed subset selection and IZ-based procedures, and Bayesian work offers posterior-probability-driven decision rules. However, there is still room to develop and compare procedures that specifically target threshold/shift parameters within Exponential/Weibull reliability settings, and to connect frequentist operating characteristics with Bayesian posterior subset selection under common experimental designs.



(a) Density function for $\eta = 1$ with $\beta \in \{1, 1.5, 2.5\}$ and $\gamma \in \{0, 3\}$.



(b) Survival function for $\eta = 1$ with $\beta \in \{1, 1.5, 2.5\}$ and $\gamma \in \{0, 3\}$.



(c) Hazard function for $\eta = 1$ with $\beta \in \{1, 1.5, 2.5\}$ and $\gamma \in \{0, 3\}$.

Figure 2.1: Effect of Weibull shape β and threshold γ on reliability behavior.

CHAPTER 3

FREQUENTIST SELECTION RULES FOR THRESHOLD PARAMETERS

This chapter develops frequentist selection procedures for k exponential populations that differ only in their threshold parameters, under the assumption of a common known scale parameter and equal sample sizes. The best population is defined as the one with the largest threshold parameter. We present two subset selection rules, one based on sample minima (R_1) and one based on sample means (R_2), each calibrated to ensure inclusion of the best population with probability at least P^* . We also describe a small-sample calibration approach based on the gamma distribution and develop an indifference-zone procedure that selects a single best population when the best threshold exceeds the others by at least a specified margin. Numerical examples and implementation notes are provided to illustrate calibration and practical use [15, 16].

3.1 Model Setup

Let $\Pi_1, \Pi_2, \dots, \Pi_k$ denote k independent populations. From population Π_i we observe n Independent and Identically Distributed (iid) lifetimes

$$X_{i1}, X_{i2}, \dots, X_{in}, \quad i = 1, 2, \dots, k,$$

and we assume independence both within and across populations.

The populations are modeled using the special case of the three-parameter Weibull distribution with shape parameter $\beta = 1$ [11, 13]. This reduces to an exponential distribution with a threshold (location/shift) parameter γ_i and a common scale parameter $\eta > 0$ (assumed known and common across all populations). In particular, for $x \geq \gamma_i$,

$$f_i(x; \gamma_i, \eta) = \frac{1}{\eta} \exp\left(-\frac{x - \gamma_i}{\eta}\right), \quad \text{and} \quad f_i(x; \gamma_i, \eta) = 0 \text{ for } x < \gamma_i.$$

Thus, the populations differ only through their threshold parameters γ_i .

Let

$$\gamma_{[1]} \leq \gamma_{[2]} \leq \cdots \leq \gamma_{[k]}$$

denote the ordered threshold parameters, so that the best population is the one with the largest threshold $\gamma_{[k]}$. Equivalently, defining

$$i^* = \arg \max_{1 \leq i \leq k} \gamma_i,$$

population Π_{i^*} is the best population. In case more than one population share the largest threshold, one of these is tagged at random and denoted the best.

A subset selection procedure returns a subset $S \subset \{1, 2, \dots, k\}$. The objective is to construct S such that the best population is included with probability at least P^* [1, 2, 4]:

$$\Pr(i^* \in S) \geq P^*.$$

Two statistics used throughout this chapter are the sample minimum and the sample mean from each population:

$$Y_i = \min_{1 \leq j \leq n} X_{ij}, \quad \bar{X}_i = \frac{1}{n} \sum_{j=1}^n X_{ij}, \quad i = 1, 2, \dots, k.$$

Since γ_i is a threshold parameter that shifts the distribution, both Y_i and $\bar{X}_i$ contain information about γ_i . Rule R_1 is based on $\{Y_i\}_{i=1}^k$, while rule R_2 is based on $\{\bar{X}_i\}_{i=1}^k$. In particular, Y_i is directly sensitive to the lower tail and is therefore naturally connected to estimation and selection based on threshold behavior, while $\bar{X}_i$ provides a complementary summary that is convenient for calibration and approximation arguments. These quantities form the basis of subset selection rules R_1 and R_2 developed in the subsequent sections.

In this chapter, calibration refers to choosing the critical constants d (for rule R_1) and b (for rule R_2) so that the resulting subset selection procedure satisfies the required

probability guarantee,

$$\Pr(i^* \in S) \geq P^*,$$

typically under least favorable parameter configurations, see [1, 2, 3]. Intuitively, larger values of d or b produce more conservative selections (larger subsets) and increase $\Pr(i^* \in S)$, while smaller values produce more aggressive selections (smaller subsets) and may reduce $\Pr(i^* \in S)$. Related selection procedures for exponential populations and "good population" selection have also been studied in the literature, see [17, 18].

3.2 Subset Rule Based on Sample Minima (R_1)

The subset selection procedure developed in this chapter follows the seminal framework of Gupta [1]. Let X_{ij} denote n independent and identically distributed iid random variables sampled from population Π_i , for $i = 1, 2, \dots, k$ and $j = 1, 2, \dots, n$. Each population is assumed to follow the exponential threshold model with threshold parameter γ_i (and common known scale parameter η as described in Section 3.2.3.1). Since observations satisfy $x \geq \gamma$ under the threshold model, the minimum order statistic is a natural estimator for γ .

Define the sample minimum from the population with threshold parameter γ_i as

$$Y_i = \min_{1 \leq j \leq n} X_{ij}, \quad i = 1, 2, \dots, k. \quad (3.1)$$

Accordingly, define $Y_{(i)}$ to be the sample minimum of the population with threshold parameter $\gamma_{[i]}$.

3.2.1 Definition of the Rule

Let $d \geq 0$ be a calibration constant. The subset selection rule R_1 selects population Π_i if its sample minimum is within d units of the largest observed sample minimum:

$$R_1 : \text{Select } \Pi_i \text{ iff } Y_i \geq \max_{1 \leq j \leq k} Y_j - d, \quad d \geq 0. \quad (3.2)$$

Equivalently, letting $Y_{[k]} = \max_{1 \leq j \leq k} Y_j$, the selected subset is

$$S = \{i \in \{1, \dots, k\} : Y_i \geq Y_{[k]} - d\}.$$

3.2.2 Distribution of the Minimum Order Statistic

Suppose $X_1, \dots, X_n$ are iid from an exponential threshold model with threshold γ and (for the moment) unit scale. Let $Y = \min(X_1, \dots, X_n)$. For $y \geq \gamma$,

$$\begin{aligned} \Pr(Y \leq y) &= 1 - \Pr(Y > y) \\ &= 1 - \Pr(X_1 > y, \dots, X_n > y) \\ &= 1 - \prod_{i=1}^n \Pr(X_i > y) \\ &= 1 - \prod_{i=1}^n [1 - \Pr(X_i \leq y)] \end{aligned} \quad (3.3)$$

$$= 1 - \prod_{i=1}^n \left\{ 1 - \left[1 - e^{-(y-\gamma)} \right] \right\} \quad (3.4)$$

$$\begin{aligned} &= 1 - \left[e^{-(y-\gamma)} \right]^n \\ &= 1 - e^{-n(y-\gamma)}, \quad y \geq \gamma. \end{aligned} \quad (3.5)$$

Differentiating (3.5) yields the density of the minimum [19]:

$$f_Y(y) = ne^{-n(y-\gamma)}, \quad y \geq \gamma. \quad (3.6)$$

3.2.3 Probability of CS and Calibration of d

Let CS denote the event that the best population (with threshold $\gamma_{[k]}$) is included in the selected subset. Under R_1 ,

$$\begin{aligned}\Pr(\text{CS}) &= \Pr\left(Y_{(k)} \geq \max_{1 \leq j \leq k-1} Y_{(j)} - d\right) \\ &= \Pr\left(\max_{1 \leq j \leq k-1} Y_{(j)} \leq Y_{(k)} + d\right) \\ &= \int_{\gamma_{[k]}}^{\infty} \left\{ \prod_{j=1}^{k-1} \Pr(Y_{(j)} \leq y + d) \right\} dF_{Y_{(k)}}(y).\end{aligned}\tag{3.7}$$

Using (3.6) for the density of $Y_{(k)}$ in the unit-scale case, (3.7) becomes

$$\Pr(\text{CS}) = \int_{\gamma_{[k]}}^{\infty} \left\{ \prod_{j=1}^{k-1} \Pr(Y_{(j)} \leq y + d) \right\} n e^{-n(y - \gamma_{[k]})} dy.$$

Let $x = y - \gamma_{[k]}$ so that $y = x + \gamma_{[k]}$. Since $\gamma_{[k]} - \gamma_{[j]} \geq 0$ for $j = 1, \dots, k-1$, we obtain the bound

$$\begin{aligned}\Pr(\text{CS}) &= n \int_0^{\infty} \left\{ \prod_{j=1}^{k-1} \Pr(Y_{(j)} - \gamma_{[j]} \leq x + \gamma_{[k]} - \gamma_{[j]} + d) \right\} e^{-nx} dx \\ &\geq n \int_0^{\infty} \left\{ \prod_{j=1}^{k-1} \Pr(Y_{(j)} - \gamma_{[j]} \leq x + d) \right\} e^{-nx} dx.\end{aligned}\tag{3.8}$$

The configuration $\gamma_1 = \gamma_2 = \dots = \gamma_k$ is the least favorable configuration LFC because it yields the minimum value of $\Pr(\text{CS})$. Setting the LFC expression in (3.8) equal to a specified value P^* (with $k^{-1} < P^* < 1$) determines the calibrated value of d . With this choice, rule R_1 guarantees

$$\Pr(\text{CS}) \geq P^*$$

for all configurations of the threshold parameters.

3.2.3.1 Common and Known Scale Parameter $\eta \neq 1$. We assume that the scale parameter is common across populations and is known, but it need not equal 1. This does not change the form of rule R_1 . If $X \sim \text{Exp}(\eta, \gamma)$ under the exponential threshold model, then for $x \geq \gamma/\eta$,

$$\begin{aligned} \Pr\left(\frac{X}{\eta} \leq x\right) &= \Pr(X \leq \eta x) \\ &= 1 - e^{\left(-\frac{\eta x - \gamma}{\eta}\right)} \\ &= 1 - e^{-(x - \gamma/\eta)} \\ &= 1 - e^{-(x - \gamma')}, \end{aligned} \tag{3.9}$$

where $\gamma' = \gamma/\eta$. Hence, rescaling by η reduces the model to the unit-scale case, and the same selection rule R_1 applies (with calibration depending on d/η).

3.3 Subset Selection Rule Based on Sample Means (R_2)

In the exponential threshold setting (equivalently, a three-parameter Weibull model with $\beta = 1$), the mean is a simple function of the threshold parameter. In particular, if $X \sim \text{Weibull}(\gamma, \eta, \beta = 1)$, then [13, 14]

$$E(X) = \eta\Gamma(2) + \gamma = \eta + \gamma, \quad \text{Var}(X) = \eta^2 \left[\Gamma(3) - \Gamma^2(2) \right] = \eta^2,$$

where $\Gamma(z)$ is the gamma function, defined as

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt.$$

Therefore, when the scale parameter η is common across populations, the population means are ordered in the same way as the threshold parameters γ_i . This motivates a subset selection rule based on sample means.

Assume $\Pi_i, i = 1, 2, \dots, k$, are k independent populations with

$$\Pi_i \sim \text{Weibull}(\gamma_i, \eta, \beta = 1),$$

where $\eta > 0$ is common and known. Let X_{ij} be an independent random sample from Π_i , for $j = 1, 2, \dots, n$. Our goal is to select a subset of populations such that the population having the largest threshold parameter is included with a specified probability P^* , where $k^{-1} < P^* < 1$. As previously noted, let

$$\gamma_{[1]} \leq \gamma_{[2]} \leq \dots \leq \gamma_{[k]}$$

denote the ordered threshold parameters. As shown in Section 3.2, the minimum-based rule R_1 targets γ directly. Since the means are also ordered by γ under common η , we now consider a complementary mean-based rule.

3.3.1 Definition of Rule R_2

Let $\bar{X}_i$ denote the sample mean from population Π_i ,

$$\bar{X}_i = \frac{1}{n} \sum_{j=1}^n X_{ij}, \quad i = 1, 2, \dots, k.$$

Define $b = b(k, n, P^*) \geq 0$ to be a calibration constant. The subset selection rule R_2 is

$$R_2 : \text{Select } \Pi_i \text{ iff } \bar{X}_i \geq \max_{1 \leq j \leq k} \bar{X}_j - b. \quad (3.10)$$

Equivalently, if $\bar{X}_{[k]} = \max_{1 \leq j \leq k} \bar{X}_j$, then the selected subset is

$$S = \{i \in \{1, \dots, k\} : \bar{X}_i \geq \bar{X}_{[k]} - b\}.$$

The constant b is chosen so that the probability of correct selection (inclusion of the best population) satisfies the requirement

$$\min_{\Omega} \Pr(\text{CS}) \geq P^*, \quad (3.11)$$

where $\Omega = (\gamma_1, \gamma_2, \dots, \gamma_k)$ denotes the threshold-parameter configuration and CS is the event that the best population is included in the selected subset.

3.3.2 Approximate Calibration of b Using the Central Limit Theorem

Let $\bar{X}_{(i)}$ be the sample mean from the population with threshold $\gamma_{[i]}$. Under $\beta = 1$ and common η , we have

$$E(\bar{X}_{(i)}) = \eta + \gamma_{[i]}, \quad \text{Var}(\bar{X}_{(i)}) = \frac{\eta^2}{n}.$$

For large n , the CLT [19] implies that $\bar{X}_{(i)}$ is approximately normal. Let

$$Z_i = \frac{\sqrt{n}}{\eta} \left(\bar{X}_{(i)} - (\eta + \gamma_{[i]}) \right), \quad i = 1, 2, \dots, k,$$

so that $Z_1, \dots, Z_k$ are approximately independent standard normal random variables. Then

$$\begin{aligned} \Pr(\text{CS}) &= \Pr\left(\bar{X}_{(k)} \geq \max_{1 \leq j \leq k-1} \bar{X}_{(j)} - b\right) \\ &= \Pr\left(Z_k \geq Z_j + (\gamma_{[j]} - \gamma_{[k]}) - \frac{\sqrt{n}}{\eta} b, \quad j = 1, 2, \dots, k-1\right) \\ &\geq \int_{-\infty}^{\infty} \Phi^{k-1}(x + cb) \phi(x) dx, \end{aligned} \quad (3.12)$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are the standard normal Cumulative Distribution Function (CDF) and Probability Density Function (PDF), respectively, and $c = \sqrt{n}/\eta$. Hence, an approximate

calibrated value $b = b(k, n, P^*)$ can be obtained by setting the right-hand side of (3.12) equal to P^* and solving numerically for b .

3.4 A Gamma-Distribution-Based Subset Selection Rule R_3

For small values of n , calibration of the mean-based procedure can be carried out using the exact gamma distribution of sums of exponential random variables, see [16]. This approach avoids reliance on the normal approximation used in the CLT calibration and can provide a more accurate implementation constant when sample sizes are small to moderate.

Assume that X_{ij} is an independent random sample from population Π_i , $j = 1, 2, \dots, n$, where

$$X_{ij} \sim \exp(\gamma_i, \eta), \quad i = 1, 2, \dots, k,$$

and that the scale parameter η is common across populations and known. Without loss of generality, we take $\eta = 1$ (the general case follows by rescaling as in Section 3.2.3.1).

Define the sum of sample values from population Π_i by

$$S_i = \sum_{j=1}^n X_{ij}, \quad \text{so that} \quad \bar{X}_i = \frac{S_i}{n}.$$

Because $X_{ij} - \gamma_i$ has an exponential distribution with threshold 0 and unit scale, the sum of shifted observations satisfies

$$\sum_{j=1}^n (X_{ij} - \gamma_i) = S_i - n\gamma_i \sim \text{Gamma}(n, 1), \quad (3.13)$$

with probability density function

$$g(x | n, 1) = \frac{x^{n-1} e^{-x}}{(n-1)!}, \quad x \geq 0, \quad (3.14)$$

and corresponding cumulative distribution function

$$G(x) = \int_0^x \frac{t^{n-1} e^{-t}}{(n-1)!} dt$$

3.4.1 Rule R_3 – Sum Form of the Mean-based Procedure

Multiplying the mean-based rule R_2 by n yields an equivalent selection rule expressed in terms of the sums. Specifically,

$$R_3 : \text{Select } \Pi_i \text{ iff } S_i \geq \max_{1 \leq j \leq k} S_j - nb, \quad b = b(k, n, P^*) \geq 0. \quad (3.15)$$

Thus, R_3 selects all populations whose sums are within nb of the largest observed sum, and it is algebraically equivalent to R_2 [16, 19].

3.4.2 Probability of Correct Selection and Calibration of b

Let CS denote the event that the best population (with threshold $\gamma_{[k]}$) is included in the selected subset. Under R_3 ,

$$\begin{aligned} \Pr(\text{CS}) &= \Pr\left(S_{(k)} \geq \max_{1 \leq j \leq k-1} S_{(j)} - nb\right) \\ &= \int_0^\infty \left\{ \prod_{j=1}^{k-1} G\left[x + n(\gamma_{[k]} - \gamma_{[j]}) + nb\right] \right\} g(x) dx \\ &\geq \int_0^\infty G^{k-1}(x + nb) g(x) dx, \end{aligned} \quad (3.16)$$

where the inequality uses $\gamma_{[k]} - \gamma_{[j]} \geq 0$ for $j = 1, \dots, k-1$. The calibration constant $b = b(k, n, P^*)$ can therefore be obtained by solving the one-dimensional equation

$$\int_0^\infty G^{k-1}(x + nb) g(x) dx = P^*, \quad (3.17)$$

for $b \geq 0$. In practice, (3.17) is solved numerically (e.g., by bisection or other root-finding routines) with the integral evaluated by numerical integration, and Monte Carlo integration can be used when convenient.

This gamma-based calibration is most useful when n is small to moderate, where the CLT approximation for sample means may be inaccurate. In later chapters, this exact small- n formulation also supports operating-characteristic calculations under standard threshold configurations.

3.4.3 Sensitivity to the common-scale assumption

To assess sensitivity of the selected subsets to the assumption of a common known scale parameter, one may specify a plausible interval $L < \eta < U$ and discretize it into m values $L = v_1 < \dots < v_m = U$. For each v_r , treat $\eta = v_r$ as known, calibrate the critical constants, and compute the resulting selected subset. Comparing the m subsets provides an empirical assessment of how uncertainty in η affects the selection decision.

3.5 Indifference Zone Procedure

As a benchmark to subset selection, we also consider an IZ procedure in the sense of Bechhofer [4]. In the IZ framework, the goal is to make a *single* assertion of the best population, rather than selecting a subset. Specifically, the population producing the largest observed minimum order statistic is asserted to be the best. The indifference-zone procedure is primarily a sample size determination tool. Given a desired probability requirement P^* and a practically meaningful separation d^* between the best population and its nearest competitor, the goal is to choose the smallest sample size $n = n(k, P^*, d^*)$ such that the probability of correctly asserting a single best population is at least P^* whenever the separation condition $\gamma_{[k]} - \gamma_{[k-1]} \geq d^*$ holds. Thus, d^* plays the role of a user-specified margin: larger values of d^* correspond to easier discrimination and typically require smaller sample sizes, while smaller values of d^* correspond to more difficult discrimination and require larger n to achieve the same P^* level. In this dissertation, the IZ approach is used as a benchmark for single-best selection and as a

practical reference for how the desired probability requirement and margin translate into sample size, complementing the subset selection procedures that target inclusion of the best population rather than a single assertion.

Let Y_i denote the sample minimum from population Π_i as defined in (3.1). The IZ objective is to ensure that the probability of correctly asserting the best population is at least P^* over a restricted parameter space in which the best threshold is meaningfully separated from its closest competitor. Let $d^* > 0$ denote a user-specified meaningful difference, and consider the parameter space

$$\gamma_{[k]} - \gamma_{[k-1]} \geq d^*,$$

where $\gamma_{[1]} \leq \dots \leq \gamma_{[k]}$ are the ordered thresholds. Within this IZ region, the sample size $n = n(k, P^*, d^*)$ is chosen so that the probability of CS satisfies

$$\Pr(\text{CS}) \geq P^*.$$

Outside this region (i.e., when the best population is not separated by at least d^*), the IZ framework does not require the probability guarantee.

3.5.1 Probability of Correct Selection Under the IZ Requirement

Under the IZ decision rule, correct selection occurs when the best population (with threshold $\gamma_{[k]}$) produces the largest sample minimum:

$$\text{CS} = \left\{ Y_{(k)} \geq \max_{1 \leq j \leq k-1} Y_{(j)} \right\}.$$

Therefore,

$$\begin{aligned}
\Pr(\text{CS}) &= \Pr\left(Y_{(k)} \geq \max_{1 \leq j \leq k-1} Y_{(j)}\right) \\
&= \Pr\left(Y_{(j)} \leq Y_{(k)}, \quad j = 1, 2, \dots, k-1\right) \\
&= \Pr\left(Y_{(j)} - \gamma_{[j]} \leq Y_{(k)} - \gamma_{[k]} + (\gamma_{[k]} - \gamma_{[j]}), \quad j = 1, \dots, k-1\right) \\
&\geq \Pr\left(Y_{(j)} - \gamma_{[j]} \leq Y_{(k)} - \gamma_{[k]} + d^*, \quad j = 1, \dots, k-1\right). \tag{3.18}
\end{aligned}$$

The inequality follows because $\gamma_{[k]} - \gamma_{[j]} \geq \gamma_{[k]} - \gamma_{[k-1]} \geq d^*$ for all $j = 1, \dots, k-1$ in the IZ parameter space.

In what follows, d^* is treated as fixed and user-specified, and we determine the required sample size $n = n(k, P^*, d^*)$ by solving for the smallest n such that the resulting lower bound on $\Pr(\text{CS})$ is at least P^* . Using the exponential-threshold model with unit scale (without loss of generality), the random variables $Y_{(j)} - \gamma_{[j]}$ and $Y_{(k)} - \gamma_{[k]}$ are independent and each has an exponential distribution with rate n . Conditioning on $X = Y_{(k)} - \gamma_{[k]}$ yields

$$\begin{aligned}
\Pr(\text{CS}) &\geq \int_0^\infty \Pr\left(Y_{(j)} - \gamma_{[j]} \leq x + d^*, \quad j = 1, \dots, k-1\right) n e^{-nx} dx \\
&= n \int_0^\infty \left[1 - e^{-n(x+d^*)}\right]^{k-1} e^{-nx} dx. \tag{3.19}
\end{aligned}$$

The right-hand side of (3.19) is the least-favorable value within the IZ region.

3.5.2 Least Favorable Configuration and Sample Size Determination

Within the IZ parameter space, the LFC is

$$\gamma_{[1]} = \gamma_{[2]} = \dots = \gamma_{[k-1]} = \gamma_{[k]} - d^*,$$

since it makes the best population as difficult as possible to distinguish from its closest competitor while still satisfying the IZ separation. The required sample size

$n = n(k, P^*, d^*)$ is obtained by finding the smallest integer n such that the bound in (3.19) meets the target:

$$n \int_0^\infty \left[1 - e^{-n(x+d^*)} \right]^{k-1} e^{-nx} dx \geq P^*.$$

In practice, for specified values of k , P^* , and d^* , the required sample size is determined numerically by finding the smallest integer n for which the above inequality holds.

3.6 Numerical Illustrations and Implementation Notes

This section provides numerical illustrations of the subset selection procedures developed earlier in the chapter. The examples show how the calibration constants are obtained, how the rules are applied to simulated exponential-threshold data, and how the indifference-zone formulation can be used for sample size determination, see [15, 16]. All computations were carried out in R, and the corresponding code is provided in the Appendix B.

3.6.1 Application of the Two Selection Rules

To illustrate rules R_1 and R_2 , consider $k = 10$ populations with common sample size $n = 25$. The threshold parameters are taken to follow an equally spaced configuration over the interval $[0, 2]$, and the simulation size is set to $N = 200,000$. This setting is used to demonstrate the calibration of the constants for the two rules and the corresponding selected subsets.

3.6.1.1 Application of Rule R_1 . For the minimum-based rule R_1 , the constant d is calculated for several values of the required probability of correct selection, P^* . The resulting values are shown in Table 3.1 and the source code for calculating this constant is found in Appendix B.1. As expected, larger values of P^* require larger values of d ,

reflecting the more conservative selection needed to guarantee inclusion of the best population.

Table 3.1: P^* and d Values, for $k = 10, n = 25$

P^*	0.75	0.90	0.95	0.975	0.99
d	0.1088	0.1500	0.1791	0.2074	0.2432

3.6.1.2 Application of Rule R_2 . For the sample-means rule R_2 , the calibration constant b was computed for the same setting of $k = 10$ and $n = 25$. The resulting values are shown in Table 3.2. The code to calculate the constant b is found in Appendix B.2.

Table 3.2: P^* and b Values, for $k = 10, n = 25$

P^*	0.75	0.90	0.95	0.975	0.99
b	0.4816	0.6274	0.7144	0.7906	0.8810

Using this same threshold configuration, random samples of size 25 were generated from each of the 10 populations. For each sample, the minimum and the sample mean were recorded. These values are reported in Table 3.3.

Table 3.3: Minimums and Means for Each of the Populations, for $k = 10$, $n = 25$

Population	Minimum	Mean
1	0.0525	0.8292
2	0.2121	1.4786
3	0.4222	1.3685
4	0.6225	1.8170
5	0.8339	1.5279
6	1.0557	2.1759
7	1.2351	2.0611
8	1.4078	2.3922
9	1.6406	2.7331
10	1.9485	2.7801

Applying rule R_1 and rule R_2 to these simulated data yields the subsets reported in Table 3.4, which compares the subsets selected by the two rules across the five values of the prescribed confidence level, P^* .

Table 3.4: P^* and the Selected Populations Using Rules R_1 and R_2

P^*	Min Procedure, R_1	Means Procedure, R_2
0.75	10	8, 9, 10
0.90	10	6, 8, 9, 10
0.95	10	6, 8, 9, 10
0.975	10	6, 7, 8, 9, 10
0.99	10	6, 7, 8, 9, 10

For this particular simulation, rule R_1 produces smaller selected subsets than rule R_2 at each of the probability levels considered. In particular, the minimum-based procedure selects only population 10 for all these probabilities: $P^* = 0.75, 0.90, 0.95, 0.975, 0.99$. In contrast, the mean-based procedure selects from three to five populations over the same range of probability requirements. Thus, for this dataset, rule R_1 would be preferred on the basis of subset size, while still retaining the best population. More broadly, the comparison shows that the relative efficiency of the two procedures may vary with the realized sample summaries and the required level of protection. The R-code for Table 3.3 and Table 3.4 can be found in Appendix B.3.

3.6.2 Indifference Zone Illustration

The indifference-zone formulation provides a practical way to determine the minimum sample size required to achieve a prescribed probability of correct selection when the best population is separated from the second-best by at least a meaningful amount. To illustrate this, consider the case of $k = 5$ populations with target probability $P^* = 0.90$ and indifference-zone parameter $d^* = 0.25$.

Table 3.5 shows evaluations of the lower bound in (3.19) for several candidate values of n . The requirement is first satisfied at $n = 12$, so the minimum sample size needed in this setting is 12 observations per population. The results of this table can be obtained from the R-code in Appendix B.4.

3.6.3 Implementation Notes

The numerical calculations in this section were carried out in R. In particular, numerical integration and simulation were used to calibrate the constants defining rules R_1 , R_2 , and the indifference-zone sample size requirement. Once these constants are

Table 3.5: Sample Size n and the Corresponding P^* Values, $k = 5$

n	P^*
10	0.8488
11	0.8801
12	0.9053
13	0.9254
14	0.9414

available, the procedures are straightforward to implement: rule R_1 depends only on the sample minima, rule R_2 depends only on the sample means, and the indifference-zone calculation determines the minimum sample size n required to achieve a specified probability of correct selection P^* for a given indifference parameter d^* .

These examples demonstrate that the proposed procedures are computationally simple and practically interpretable. They also show how the probability requirement P^* influences the size of the selected subset under the subset selection rules and the required sample size under the indifference-zone framework.

3.7 Results and Takeaways

This chapter developed frequentist selection procedures for comparing k exponential populations that differ only in their threshold parameters. The first procedure, rule R_1 , is based on sample minima and directly targets the threshold structure of the model. The second procedure, rule R_2 , is based on sample means and uses the fact that, under a common known scale parameter, the population means are ordered in the same way as the thresholds. For small sample sizes, the equivalent sum-based formulation R_3 allows calibration through the exact gamma distribution, thereby avoiding reliance on the large-sample normal approximation, [15, 16].

For the subset selection problem, the calibration constants were chosen so that the resulting procedure includes the best population with probability at least P^* . The numerical example showed that both R_1 and R_2 satisfy this objective, but they may differ substantially in the size of the selected subset. In the revised example considered here, the minimum-based rule R_1 produced smaller subsets than the mean-based rule R_2 across all values of P^* examined. In particular, rule R_1 selected only population 10 for all of the probability levels considered, while rule R_2 selected larger subsets containing three or four populations. For this dataset, the minimum-based rule therefore appeared more efficient in terms of subset size while still retaining the best population.

The chapter also introduced an indifference-zone procedure for selecting a single best population when the best threshold exceeds the second-best by at least a specified margin d^* . In that setting, the main objective is sample size determination rather than subset formation. The numerical illustration showed how the required sample size can be obtained by finding the smallest integer n such that the lower bound on the probability of correct selection is at least the desired value P^* .

Taken together, these results provide a frequentist framework for both subset selection and single-best selection in exponential threshold models. The procedures developed here illustrate how the desired probability requirement P^* affects both the conservativeness of the selected subset and, under the indifference-zone formulation, the required sample size. They also show that the relative efficiency of rules R_1 and R_2 may depend on the realized sample summaries, even though both procedures are calibrated to satisfy the same probability guarantee.

CHAPTER 4

OPERATING CHARACTERISTICS AND COMPARATIVE PERFORMANCE

This chapter develops Operating Characteristics of the subset selection rules developed in Chapter 3. The analyses focus on the probability of including the best population PCS and on the typical size of the selected subset ESS, under standard threshold configurations such as slippage and equi-spaced settings.

4.1 Operating Characteristics and Threshold Configurations

4.1.1 Operating Characteristics

Let $S \subset \{1, \dots, k\}$ denote the random subset returned by a selection rule, and let

$$i^* = \arg \max_{1 \leq i \leq k} \gamma_i$$

be the index of the best population, the one with the largest threshold parameter, $\gamma_{[k]}$. The primary OC for subset selection is the probability of correct selection PCS,

$$\text{PCS} = \Pr(i^* \in S).$$

These OC definitions are standard in the ranking-and-selection literature [3].

An Incorrect Selection (ICS) occurs when a non-best population is included in the selected subset; for exchangeable non-best populations under a given configuration, we may speak of $\Pr(\text{ICS})$ as the probability that a representative non-best population is selected.

Define the indicator function,

$$S[i] = \begin{cases} 1 & \text{if } \Pi_i \in S \\ 0 & \text{otherwise} \end{cases} \quad (4.1)$$

Then, a second key OC is the ESS,

$$\text{ESS} = \mathbb{E}(|S|) = \mathbb{E} \left[\sum_{i=1}^k S[i] \right] = \sum_{i=1}^k \Pr(\Pi_i \in S), \quad (4.2)$$

a common efficiency metric for subset selection procedures [2]. Here $|S|$ is the size of the selected subset yielded by any of the selection rules.

4.1.2 Threshold Configurations

Operating characteristics depend on the configuration of thresholds $(\gamma_1, \dots, \gamma_k)$.

Following [16], to compare procedures systematically, we evaluate OCs under the following standard configurations [3].

4.1.2.1 Slippage Configuration. A single population is best, while the remaining $k - 1$ populations share a common threshold, and the best population is separated by $\delta \geq 0$.

With unit scale, the configuration may be written as

$$\gamma_{[1]} = \gamma_{[2]} = \dots = \gamma_{[k-1]} = \gamma, \quad \gamma_{[k]} = \gamma + \delta.$$

Without loss of generality, one may take $\gamma = 0$. This configuration is useful because it represents a difficult scenario in which the best population is only weakly separated when δ is small.

4.1.2.2 Equi-spaced Configuration. Thresholds are equally spaced by a fixed increment $\delta > 0$. A convenient parameterization for k populations is

$$\gamma_i = \gamma_1 + (i - 1)\delta, \quad i = 1, 2, \dots, k,$$

so that adjacent thresholds differ by δ : $\gamma_{i+1} - \gamma_i = \delta$ for all i . Under this configuration, $\Pr(\Pi_i \in S)$ varies systematically with i , and ESS is computed as the sum of all selection probabilities [16]. Under both threshold configurations, increasing δ makes the populations easier to distinguish. For fixed calibration targets, larger δ typically leads to smaller selected subsets, while PCS remains at or above the target.

4.2 Operating Characteristics of the Selection Rules

4.2.1 Simulation-based OCs Evaluation

For general (k, n, P^*) and a specified threshold configuration, OCs are evaluated by Monte Carlo simulation as follows. For each replication, random samples of size n are drawn independently from each population under the exponential threshold model, the selection rule is applied, and selection indicators are recorded [20]. Repeating this process N times yields empirical estimates of $\Pr(\Pi_i \in S)$, from which $\Pr(\text{CS})$ and ESS are computed [16]. In the OCs study reported here, we use large values of N (e.g., $N = 200,000$) to stabilize estimates. The R-code for evaluating the OCs is given in Appendix B.

4.2.2 Slippage Configuration Results

Under slippage configurations, $\Pr(\text{CS})$ for R_1 and R_2 are typically close because both rules are calibrated to satisfy the same PCS requirement at the least favorable configuration. However, the procedures can differ greatly in ESS. For example, for

$k = 10$, $n = 25$, and $P^* = 0.95$, Table 4.1 shows that while both rules achieve $\Pr(\text{CS}) \approx 0.95$ at $\delta = 0$, the minimum-based rule R_1 yields substantially smaller expected subset sizes than the mean-based rule R_2 for $\delta > 0$. This pattern, as it can be observed in Table 4.1, persists for smaller settings such as $k = 5$, $n = 10$. See also [16]. In the slippage configuration, all $k - 1$ non-best populations are exchangeable, and we summarize incorrect inclusion by the sum of non-best selection probabilities,

$$\text{ICS}_\Sigma = \sum_{i \neq i^*} \Pr(\Pi_i \in S).$$

This quantity equals the expected number of incorrectly included populations and therefore need not be bounded by 1. With this convention,

$$\text{ESS} = \Pr(\text{CS}) + \text{ICS}_\Sigma,$$

since $\text{ESS} = \sum_{i=1}^k \Pr(\Pi_i \in S)$. In the OCs tables, ICS_Σ is reported in the row labeled $\Pr(\text{ICS})$. The R-code for obtaining these results is in Appendix B.5

Table 4.1: OCs for Rules R_1 (red) and R_2 (blue): Slippage Configuration
 $k = 10$, $n = 25$, $P^* = 0.95$, $d = 0.1791$, $b = 0.7144$, $N = 200,000$

δ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$\Pr(\text{CS})$	0.9499	0.9960	0.9997	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
$\Pr(\text{CS})$	0.9500	0.9801	0.9930	0.9976	0.9993	0.9998	0.9999	1.0000	1.0000	1.0000	1.0000
$\Pr(\text{ICS})$	8.5523	8.0159	2.6248	0.2189	0.018	0.0013	0.0001	0.0000	0.0000	0.0000	0.0000
$\Pr(\text{ICS})$	8.5435	8.4776	8.3342	8.0539	7.5709	6.8290	5.8173	4.6201	3.3822	2.2658	1.3806
ESS	9.5022	9.0118	3.6245	1.2189	1.0180	1.0013	1.0001	1.0000	1.0000	1.0000	1.0000
ESS	9.4935	9.4577	9.3272	9.0515	8.5702	7.8288	6.8172	5.6201	4.3822	3.2658	2.3806

4.2.3 Equi-spaced Configuration Results

A similar conclusion holds under equi-spaced configurations. When $k = 10$, $n = 25$, and $P^* = 0.95$, the OCs results indicate that R_1 again produces markedly smaller ESS than R_2 for all $\delta > 0$ [16]. In this configuration, the full set of estimated selection probabilities $\Pr(\Pi_i \in S)$ provides additional insight into where each procedure "spends" its subset size: mean-based selection tends to include more near-best populations for a wider range of separations. Table 4.2 gives a complete picture of the results obtained under this configuration and Appendix B.6 gives the R-code to obtain these results.

4.2.4 Explaining the Expected Subset Size ESS Gap via Moments

An intuitive explanation for the observed ESS gap comes from comparing the variability of the statistics underlying the two rules. Under unit scale, the sample minimum satisfies $\text{Var}(Y_i) = 1/n^2$, whereas the sample mean satisfies $\text{Var}(\bar{X}_i) = 1/n$; see Table 4.3 and [16]. These moment expressions are standard for exponential samples (e.g., [11]). Moreover, we have $E(Y_i) = \gamma_i + 1/n$ and $E(\bar{X}_i) = \gamma_i + 1$, so the centered statistics $Y_i - 1/n$ and $\bar{X}_i - 1$ are both unbiased estimators of the threshold parameter γ_i . Defining

$$\hat{\gamma}_i^{(1)} = Y_i - \frac{1}{n}, \quad \hat{\gamma}_i^{(2)} = \bar{X}_i - 1,$$

the two selection rules can be written in a parallel form:

$$R_1 : \Pi_i \in S \iff \hat{\gamma}_i^{(1)} \geq \max_j \hat{\gamma}_j^{(1)} - d, \quad R_2 : \Pi_i \in S \iff \hat{\gamma}_i^{(2)} \geq \max_j \hat{\gamma}_j^{(2)} - b.$$

Thus, both procedures apply the same "within a margin of the best" decision template to estimators of γ_i , but the estimator underlying R_1 is far less variable. This efficiency gap translates into more decisive subset selection (smaller ESS) for R_1 once the thresholds separate.

Table 4.2: OCs for Rules R_1 (red) and R_2 (blue): Equi-Spaced Configuration
 $k = 10$, $n = 25$, $P^* = 0.95$, $d = 0.1791$, $b = 0.7144$, $N = 200,000$

δ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$\text{Pr}(\pi_1)$	0.9505	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_1)$	0.9490	0.1294	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_2)$	0.9500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_2)$	0.9498	0.2258	0.0004	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_3)$	0.9502	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_3)$	0.9487	0.3594	0.0041	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_4)$	0.9506	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_4)$	0.9496	0.5118	0.0247	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_5)$	0.9498	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_5)$	0.9498	0.6671	0.1060	0.0022	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_6)$	0.9502	0.0020	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_6)$	0.9496	0.7961	0.3074	0.0323	0.0009	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_7)$	0.9503	0.0237	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_7)$	0.9490	0.8899	0.5999	0.2211	0.0373	0.0027	0.0001	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_8)$	0.9504	0.2881	0.0021	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_8)$	0.9488	0.9486	0.8451	0.6326	0.3620	0.1468	0.0418	0.0083	0.0012	0.0002	0.0000
$\text{Pr}(\pi_9)$	0.9504	0.9312	0.2967	0.0244	0.0020	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_9)$	0.9493	0.9810	0.9643	0.9290	0.8691	0.7786	0.6587	0.5189	0.3768	0.2521	0.1529
$\text{Pr}(\text{CS})$	0.9499	0.9996	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
$\text{Pr}(\text{CS})$	0.9500	0.9963	0.9991	0.9997	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
ESS	9.5022	2.2448	1.2987	1.0244	1.0020	1.0001	1.0000	1.0000	1.0000	1.0000	1.0000
ESS	9.4935	6.5053	3.8512	2.8169	2.2692	1.9281	1.7006	1.5272	1.3780	1.2522	1.1529

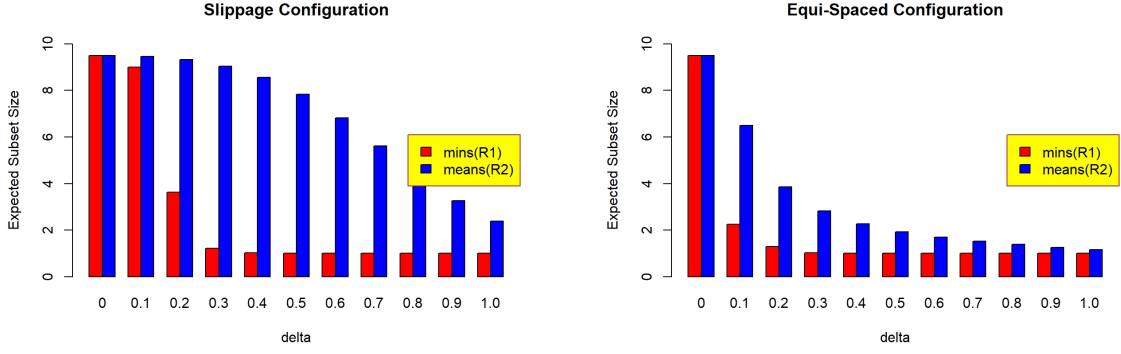


Figure 4.1: ESS with data from Table 4.1 and Table 4.2.
 $k = 10, n = 25, P^* = 0.95, N = 200,000$

Figure 4.1 provides a visual summary of the ESS patterns underlying the comparisons in Tables 4.1 and 4.2. In both the slippage and equi-spaced configurations, the expected subset size decreases as the separation parameter δ increases, reflecting that larger threshold separation makes the best population easier to identify. However, the minimum-based rule R_1 becomes decisive much faster: ESS drops rapidly toward 1 once δ is moderately positive, indicating that the rule is selecting essentially a single population. By contrast, the mean-based rule R_2 remains substantially more conservative, especially under slippage, where ESS decreases much more slowly and stays large for a wide range of δ . These visual trends are consistent with the moment comparison in Table 4.3: under unit scale, the sample minimum has variance $1/n^2$ while the sample mean has variance $1/n$, so the statistic used by R_1 is considerably less variable and therefore yields sharper discrimination once the thresholds separate. In short, the variance gap between minima and means translates directly into a decisiveness gap in expected subset size. The R-code for reproducing these plots is found on Appendix B.7.

Table 4.3: Moments of Sample Statistics Underlying Selection Rules R_1 and R_2

Statistic	Sample Minimum	Sample Mean
Expected Value	$\gamma + 1/n$	$\gamma + 1$
Variance	$1/n^2$	$1/n$

4.3 Operating Characteristics Under Gamma Distribution

When n is small, the mean-based procedure can be implemented via the sum $S_i = \sum_{j=1}^n X_{ij}$, with $\bar{X}_i = S_i/n$. In the unit-scale exponential threshold model,(see [19])

$$S_i - n\gamma_i = \sum_{j=1}^n (X_{ij} - \gamma_i) \sim \text{Gamma}(n, 1),$$

which yields exact integral expressions for $\text{Pr}(\text{CS})$, $\text{Pr}(\text{ICS})$, and ESS for the sum-based implementation of the mean rule. For the mean-based rule, this gamma representation is particularly useful because it provides an exact small- n benchmark for operating characteristics that are otherwise computed by simulation or approximated via the CLT.

Table 4.4 summarizes the OCs for R_2 when $k = 10$, $n = 25$, and $P^* = 0.95$, comparing simulation estimates (blue), the CLT approximation (black), and the exact gamma-based evaluation (purple). All three methods agree closely across the full range of separations. In particular, $\text{Pr}(\text{CS})$ matches to at least three decimal places throughout and quickly approaches 1 as δ increases. The same close agreement is seen in $P(\text{ICS})$ and ESS, indicating that the CLT approximation is already accurate for $n = 25$ and that the simulation study is well-calibrated. The table also highlights the practical behavior of R_2 as separation increases. At $\delta = 0$, calibration enforces $\text{Pr}(\text{CS}) = 0.95$, but the procedure

Table 4.4: Comparing the OCs for R_2 and R_3 .
Simulation (blue), CLT (black), and Gamma Distribution (purple)

δ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Pr(CS)	0.9500	0.9801	0.9930	0.9976	0.9993	0.9998	0.9999	1.0000	1.0000	1.0000	1.0000
Pr(ICS)	8.5434	8.4747	8.3340	8.0537	7.5706	6.8285	5.8168	4.6195	3.3817	2.2654	1.3802
ESS	9.4934	9.4575	9.3269	9.0513	8.5699	7.8283	6.8168	5.6195	4.3817	3.2654	2.3802
Pr(CS)	0.9500	0.9802	0.9932	0.9980	0.9995	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000
Pr(ICS)	8.5499	8.4781	8.3115	7.9806	7.4141	6.5741	5.4894	4.2648	3.0518	1.9950	1.1836
ESS	9.4998	9.4583	9.3047	8.9785	8.4136	7.5740	6.4894	5.2648	4.0518	2.9950	2.1836
Pr(CS)	0.9500	0.9796	0.9925	0.9975	0.9992	0.9998	0.9999	1.0000	1.0000	1.0000	1.0000
Pr(ICS)	8.5500	8.4841	8.3417	8.0647	7.5832	6.8413	5.8353	4.6405	3.4016	2.2800	1.3913
ESS	9.5000	9.4637	9.3343	9.0622	8.5824	7.8411	6.8352	5.6405	4.4016	3.2800	2.3913

remains conservative: $P(\text{ICS}) \approx 8.55$ and $\text{ESS} \approx 9.5$, meaning that on average nearly all populations are retained in the selected subset. As δ grows, incorrect inclusion declines steadily and the expected subset size shrinks. For example, under the exact gamma evaluation, ESS decreases from 9.50 at $\delta = 0$ to about 7.84 at $\delta = 0.5$ and to about 2.39 at $\delta = 1.0$, reflecting increasing decisiveness as the best threshold becomes more separated. The results in the Table 4.4 can be reproduced using the code in Appendies B.8 and B.9.

4.4 Results and Takeaways

Across both slippage and equi-spaced configurations, the operating characteristic study shows that the minimum-based rule R_1 consistently yields smaller expected subset sizes than the mean-based rule R_2 for $\delta > 0$, while maintaining the required PCS guarantee [16]. The gamma-based formulation for the mean rule provides an exact

small- n alternative that agrees closely with simulation-based OC estimates, lending support to the simulation methodology for practical use.

For the mean-based rule in the small-to-moderate sample setting, the gamma representation provides a useful validation check. Table 4.4 reports OCs for R_2 when $k = 10$, $n = 25$, and $P^* = 0.95$, comparing simulation results to both the CLT approximation and the exact gamma-based formulation. Across the full range of separations, the three methods agree closely: PCS is essentially identical (and quickly approaches 1 as δ increases), and the corresponding ESS values track each other very closely. In this setting, the gamma-based results indicate that the CLT approximation is already quite accurate, while also supporting the reliability of the simulation-based OCs estimates used elsewhere in this chapter.

The OCs comparisons highlight that procedures calibrated to the same PCS requirement can behave quite differently in terms of subset size. In particular, under slippage configurations (a difficult scenario), subset sizes can remain large for small δ , while under equi-spaced configurations subsets tend to shrink faster as separation increases. These OCs patterns motivate the exact $k = 2$ analysis in Chapter 5, which provides closed-form expressions explaining the mechanisms behind the observed differences.

To illustrate the magnitude of the subset-size differences, consider first the slippage configuration with $k = 10$, $n = 25$, and $P^* = 0.95$. At $\delta = 0.30$, the expected subset size for the minimum-based rule drops to $ESS = 1.2189$, while the mean-based rule remains highly conservative with $ESS = 9.0515$; at $\delta = 0.50$, the contrast persists with $ESS = 1.0013$ for R_1 versus $ESS = 7.8288$ for R_2 . A similar pattern appears in the small- n setting $k = 5$, $n = 10$, $P^* = 0.95$: under slippage, ESS decreases from 4.7508 at $\delta = 0$ to 2.3896 by $\delta = 0.40$ for R_1 , while R_2 remains much larger at $ESS = 4.5461$; by

$\delta = 0.70$, R_1 is already near single-population selection with $\text{ESS} = 1.0698$, whereas R_2 still averages nearly four populations with $\text{ESS} = 3.9898$. Under the equi-spaced configuration for $k = 5$, $n = 10$, $P^* = 0.95$, the same decisiveness gap is visible: at $\delta = 0.50$, R_1 has $\text{ESS} = 1.1308$ while R_2 remains at $\text{ESS} = 2.4906$, and this gap persists as δ increases. For a complete picture on the results for these parameters see Table 4.5 and Table 4.6. The R-code for reproducing these tables is in Appendix B.10 and B.11.

These numerical contrasts are consistent with the fact that the sample minimum is much less variable than the sample mean under the exponential-threshold model (for unit scale, $\text{Var}(Y_i) = 1/n^2$ versus $\text{Var}(\bar{X}_i) = 1/n$), so once thresholds separate, R_1 eliminates non-best populations more quickly and yields substantially smaller subsets.

Table 4.5: OCs for Rules R_1 (red) and R_2 (blue): Slippage Configuration
 $k = 5$, $n = 10$, $P^* = 0.95$, $d = 0.3652$, $b = 1.0160$, $N = 200,000$

δ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Pr(CS)	0.9500	0.9810	0.9930	0.9974	0.9990	0.9997	0.9999	1.0000	1.0000	1.0000	1.0000
Pr(CS)	0.9500	0.9693	0.9818	0.9895	0.9940	0.9968	0.9983	0.9991	0.9995	0.9998	0.9999
Pr(ICS)	3.8008	3.7151	3.4866	2.8621	1.3905	0.5166	0.1909	0.0699	0.026	0.0096	0.0036
Pr(ICS)	3.8008	3.7703	3.7239	3.6531	3.5521	3.4138	3.2268	2.9907	2.705	2.3799	2.0267
ESS	4.7508	4.6962	4.4796	3.8595	2.3895	1.5163	1.1908	1.0698	1.0259	1.0096	1.0036
ESS	4.7508	4.7396	4.7057	4.6426	4.5461	4.4106	4.2251	3.9898	3.7045	3.3797	3.0266

Table 4.6: OCs for Rules R_1 (red) and R_2 (blue): Equi-spaced Configuration
 $k = 5$, $n = 10$, $P^* = 0.95$, $d = 0.3652$, $b = 1.0160$, $N = 200,000$

δ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$\text{Pr}(\pi_1)$	0.9506	0.2930	0.0062	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_1)$	0.9504	0.8375	0.5765	0.2596	0.0708	0.0122	0.0015	0.0001	0.0000	0.0000	0.0000
$\text{Pr}(\pi_2)$	0.9498	0.6552	0.0448	0.0024	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_2)$	0.9499	0.8930	0.7559	0.5302	0.2861	0.1153	0.0349	0.0085	0.0017	0.0003	0.0000
$\text{Pr}(\pi_3)$	0.9502	0.8690	0.3383	0.0469	0.0065	0.0009	0.0001	0.0000	0.0000	0.0000	0.0000
$\text{Pr}(\pi_3)$	0.9506	0.9358	0.8863	0.7937	0.6547	0.4844	0.3145	0.1789	0.0888	0.0391	0.0152
$\text{Pr}(\pi_4)$	0.9503	0.9581	0.9024	0.7397	0.3540	0.1299	0.0480	0.0176	0.0066	0.0024	0.0010
$\text{Pr}(\pi_4)$	0.9499	0.9638	0.9594	0.9431	0.9165	0.8794	0.8300	0.7665	0.6908	0.6054	0.5134
$\text{Pr}(\pi_5)$	0.9500	0.9926	0.9978	0.9993	0.9998	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000
$\text{Pr}(\pi_5)$	0.9500	0.9832	0.9932	0.9969	0.9984	0.9992	0.9996	0.9998	0.9999	0.9999	1.0000
ESS	4.7508	3.7680	2.2896	1.7883	1.3603	1.1308	1.0481	1.0176	1.0066	1.0024	1.0010
ESS	4.7508	4.6133	4.1713	3.5234	2.9265	2.4906	2.1804	1.9538	1.7812	1.6448	1.5286

CHAPTER 5

EXACT RESULTS FOR TWO POPULATIONS

Ranking and selection procedures are commonly developed to identify the best population, or a subset containing the best population, while controlling the probability of correct selection [1, 2, 3, 4, 21]. This chapter specializes to $k = 2$ exponential threshold populations and derives exact operating characteristics for the minimum-based rule R_1 , the CLT-based mean rule R_2 , and the Gamma-based mean rule R_3 [16, 22]. The two-population case permits closed-form expressions for the probability of correct selection PCS, the probability of incorrect selection Probability of Incorrect Selection (PICS), and the expected subset size ESS, as well as explicit formulae for the calibrated constants d and b . This chapter specializes to $k = 2$ populations and derives exact operating characteristics for the minimum-based rule R_1 and the mean-based rule R_2 [16, 22].

5.1 Exact OCs for $k = 2$ Populations

Assume two exponential threshold populations with thresholds $\gamma_{[1]} \leq \gamma_{[2]}$. For OCs calculations under slippage, set

$$\gamma_{[1]} = 0, \quad \gamma_{[2]} = \delta, \quad \delta \geq 0,$$

without loss of generality [22]. Let $Y_{(i)}$ denote the minimum order statistic from the population with threshold $\gamma_{[i]}$, and let $\bar{X}_{(i)}$ denote the corresponding sample mean.

The use of a random sized selected subset is central to the subset-selection approach. In this setting, the goal is not only to include the best population with the prescribed probability, but also to avoid unnecessarily large selected subsets. For $k = 2$,

the expected subset size simplifies to

$$\text{ESS} = \mathbb{E}(|S|) = \mathbb{E}(S_{[1]}) + \mathbb{E}(S_{[2]}) = \Pr(\text{ICS}) + \Pr(\text{CS}) \quad (5.1)$$

since the subset can contain population 2 (correct inclusion) and/or population 1 (incorrect inclusion).

5.2 Exact Computation of Constants d and b and OCs Evaluation

5.2.1 Exact OCs for R_1

Let $Y_{(1)}$ and $Y_{(2)}$ denote the sample minima from the two populations. Under rule R_1 , population i is selected iff $Y_i \geq \max(Y_1, Y_2) - d$ for $d \geq 0$.

For the slippage configuration $(0, \delta)$, the probability of correct selection (including the best population), under R_1 is

$$\Pr(\text{CS}_1) = \Pr(Y_{(2)} \geq Y_{(1)} - d) = 1 - \frac{1}{2}e^{(-n(d+\delta))}. \quad (5.2)$$

This PCS is minimized at $\delta = 0$. Therefore, to achieve $\Pr(\text{CS}_1) \geq P^*$ for all $\delta \geq 0$, the calibrated constant is

$$d = -\frac{\ln(2(1 - P^*))}{n}. \quad (5.3)$$

The probability of incorrectly selecting the non-best population under R_1 satisfies

$$\Pr(\text{ICS}_1) = \Pr(Y_{(1)} \geq Y_{(2)} - d) = \begin{cases} \frac{1}{2}e^{(-n(\delta-d))}, & \delta \geq d, \\ 1 - \frac{1}{2}e^{(-n(d-\delta))}, & d \geq \delta. \end{cases} \quad (5.4)$$

Finally, the expected subset size is

$$\text{ESS}_1 = \Pr(\text{CS}_1) + \Pr(\text{ICS}_1). \quad (5.5)$$

5.2.2 Exact OCs for Rule R_2

For large n , the sample means are approximately normal, $\bar{X}_i \sim N(\gamma_i + 1, 1/n)$ [19]. Under unit scale, $k = 2$ and slippage $(0, \delta)$, the probability of correct selection for R_2 satisfies

$$\Pr(\text{CS}_2) = \Pr(\bar{X}_{(2)} \geq \bar{X}_{(1)} - b) = \Phi\left(\frac{b + \delta}{\sqrt{2/n}}\right), \quad (5.6)$$

where $\Phi(\cdot)$ is the standard normal CDF [16]. Since $\delta \geq 0$ implies $\Pr(\text{CS}_2) \geq \Phi\left(\frac{b}{\sqrt{2/n}}\right)$, the calibrated constant satisfying $\Pr(\text{CS}_2) \geq P^*$ for all $\delta \geq 0$ is

$$b = \sqrt{\frac{2}{n}} \Phi^{-1}(P^*). \quad (5.7)$$

Similarly, the incorrect selection probability is

$$\Pr(\text{ICS}_2) = \Pr(\bar{X}_{(1)} \geq \bar{X}_{(2)} - b) = \Phi\left(\frac{b - \delta}{\sqrt{2/n}}\right), \quad (5.8)$$

and

$$\text{ESS}_2 = \Pr(\text{CS}_2) + \Pr(\text{ICS}_2). \quad (5.9)$$

5.2.3 Exact OCs for Gamma-Based Rule R_3

The purpose of including R_3 in the present chapter is to separate the effect of using sample means from the effect of using the CLT approximation. Since R_2 and R_3 are both mean-based rules, differences between them reflect the use of the normal approximation in R_2 versus the exact finite-sample Gamma calculation in R_3 [19, 23].

For rule R_3 , the selection rule is written in terms of the sample sums

$$S_i = \sum_{j=1}^n X_{ij}, \quad i = 1, 2.$$

Using the notation from Section 3.4, population Π_i is selected iff

$$S_i \geq \max(S_1, S_2) - nb, \quad (5.10)$$

where $b = b(k, n, P^*)$ is calibrated using the exact gamma distribution.

Under the slippage configuration $(0, \delta)$, population Π_2 is the best population.

Hence

$$S_1 = G_1, \quad S_2 = n\delta + G_2,$$

where G_1 and G_2 are independent $\text{Gamma}(n, 1)$ random variables with density $g(\cdot)$ and CDF $G(\cdot)$.

Conditioning on G_2 , and averaging over its density, the probability of correct selection is

$$\begin{aligned} \Pr(\text{CS}_3) &= \Pr(S_2 \geq S_1 - nb) \\ &= \Pr(G_1 \leq G_2 + n(\delta + b)) \\ &= \int_0^\infty G(x + n(\delta + b)) g(x) dx. \end{aligned} \quad (5.11)$$

This probability is minimized at $\delta = 0$. Therefore, for $k = 2$, the gamma-based calibration constant b is obtained by solving

$$\int_0^\infty G(x + nb) g(x) dx = P^*. \quad (5.12)$$

The probability of incorrectly selecting the non-best population is

$$\begin{aligned} \Pr(\text{ICS}_3) &= \Pr(S_1 \geq S_2 - nb) \\ &= \Pr(G_1 \geq G_2 + n(\delta - b)). \end{aligned} \quad (5.13)$$

Thus,

$$\Pr(\text{ICS}_3) = \begin{cases} \int_0^\infty G(x + n(b - \delta)) g(x) dx, & b \geq \delta, \\ \int_0^\infty \{1 - G(x + n(\delta - b))\} g(x) dx, & \delta \geq b. \end{cases} \quad (5.14)$$

For a more detailed derivation on Equation (5.14) see Appendix A.2.

Finally, the expected subset size is

$$\text{ESS}_3 = \Pr(\text{CS}_3) + \Pr(\text{ICS}_3). \quad (5.15)$$

For $k = 2$, these expressions give the exact operating characteristics of the gamma-based rule. Unlike the CLT-based rule R_2 , which replaces the distribution of the sample means by a normal approximation, rule R_3 evaluates the operating characteristics using the exact finite-sample gamma distribution of the sums.

5.3 Comparative Findings Under Slippage and Equi-spaced Configuration

When $k = 2$, the equi-spaced configuration reduces to the same one-parameter family $(0, \delta)$ as the slippage configuration. Therefore, OCs comparisons can be presented as functions of δ for fixed (n, P^*) . In particular, comparing (5.2)–(5.8) shows that both procedures satisfy the required PCS guarantee by construction, but R_1 typically yields much smaller PICS (and thus smaller ESS) once δ exceeds the calibration threshold d [16, 22]. In particular, one compares $\Pr(\text{CS}_1)$ versus $\Pr(\text{CS}_2)$, $\Pr(\text{ICS}_1)$ versus $\Pr(\text{ICS}_2)$, and ESS_1 versus ESS_2 as δ varies.

For $k = 2$ with $n = 25$ and $P^* = 0.95$, Table 5.1 shows that both rules attain the required PCS at $\delta = 0$ by construction. As δ increases, the PCS values under R_1 rise to essentially one much faster than under R_2 . The more substantial difference appears in the incorrect-selection probability: under R_1 , PICS drops sharply once δ exceeds the calibrated constant d , and ESS quickly approaches 1, indicating near single-population

Table 5.1: OCs for Rules R_1 (red) and R_2 (blue): $\gamma_{[2]} - \gamma_{[1]} = \delta$
 $k = 2$, $n = 25$, $d = 0.0917$, $b = 0.4647$, $P^* = 0.95$, $N = 300,000$

δ	0	0.05	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55
Pr(CS)	0.9500	0.9854	0.9989	0.9997	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Pr(CS)	0.9500	0.9654	0.9847	0.9902	0.9937	0.9960	0.9975	0.9986	0.9992	0.9995	0.9997
Pr(ICS)	0.9495	0.8246	0.1162	0.0333	0.0097	0.0026	0.0007	0.0002	0.0001	0.0000	0.0000
Pr(ICS)	0.9502	0.9296	0.8694	0.8279	0.7793	0.7228	0.6591	0.5919	0.5205	0.4486	0.3786
ESS	1.8995	1.8100	1.1151	1.0330	1.0096	1.0026	1.0007	1.0002	1.0001	1.0000	1.0000
ESS	1.9002	1.8951	1.8542	1.8181	1.7731	1.7188	1.6567	1.5906	1.5196	1.4481	1.3784

selection. By contrast, under R_2 the decay of PICS is much slower, and ESS remains noticeably larger than 1 even for moderate separation values, reflecting continued inclusion of the non-best population with non-negligible probability.

Table 5.2 exhibits the same qualitative behavior in the smaller-sample setting $n = 10$ with $P^* = 0.90$. While both procedures satisfy the PCS requirement at $\delta = 0$, R_1 again becomes decisive more quickly: ESS decreases rapidly toward 1 as δ increases. In contrast, R_2 remains conservative across the range of separations shown, with a comparatively slow reduction in Pr(ICS) and correspondingly larger ESS values.

For $n = 50$ and the more strict requirement $P^* = 0.99$ (Table 5.3), both rules again achieve the target PCS at $\delta = 0$. As separation increases, the contrast in decisiveness becomes even more pronounced: R_1 drives Pr(ICS) to near zero at relatively small values of δ , producing ESS values essentially equal to 1, while R_2 continues to include the non-best population with substantial probability over the same range, keeping ESS well

Table 5.2: OCs for Rules R_1 (red) and R_2 (blue): $\gamma_{[2]} - \gamma_{[1]} = \delta$
 $k = 2, \quad n = 10, \quad d = 0.1601, \quad b = 0.5651, \quad P^* = 0.90, \quad N = 300,000$

δ	0	0.05	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55
Pr(CS)	0.9000	0.9397	0.9778	0.9866	0.9918	0.9949	0.9969	0.9981	0.9989	0.9994	0.9996
Pr(CS)	0.9000	0.9176	0.9455	0.9561	0.9647	0.9719	0.9779	0.9827	0.9864	0.9893	0.9918
Pr(ICS)	0.8992	0.8338	0.5462	0.3343	0.2026	0.1223	0.0741	0.0447	0.0272	0.0165	0.0100
Pr(ICS)	0.9000	0.8793	0.8288	0.7984	0.7653	0.7294	0.6905	0.6485	0.6044	0.5593	0.5131
ESS	1.7992	1.7734	1.5240	1.3209	1.1945	1.1172	1.0710	1.0428	1.0262	1.0158	1.0096
ESS	1.8000	1.7968	1.7743	1.7545	1.7300	1.7013	1.6683	1.6312	1.5908	1.5486	1.5049

Table 5.3: OCs for Rules R_1 (red) and R_2 (blue): $\gamma_{[2]} - \gamma_{[1]} = \delta$
 $k = 2, \quad n = 50, \quad d = 0.0785, \quad b = 0.4688, \quad P^* = 0.99, \quad N = 300,000$

δ	0	0.05	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55
Pr(CS)	0.9900	0.9993	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Pr(CS)	0.9900	0.9949	0.9988	0.9995	0.9997	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000
Pr(ICS)	0.9903	0.8798	0.0142	0.0011	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Pr(ICS)	0.9898	0.9814	0.9444	0.9111	0.8648	0.8030	0.7261	0.6352	0.5377	0.4382	0.3432
ESS	1.9803	1.8791	1.0142	1.0011	1.0001	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
ESS	1.9798	1.9763	1.9432	1.9106	1.8645	1.8029	1.7261	1.6352	1.5377	1.4382	1.3432

above 1. This illustrates that even under high PCS requirements, the minimum-based rule

can remain highly decisive once the best population is separated. The R-code to replicate Tables 5.2–5.1 can be found in Appendix B.12.

5.4 Numerical Verification of Operating Characteristic Comparisons, $k = 2$

The preceding sections derived exact operating characteristics for the minimum-based rule R_1 , the CLT-based mean rule R_2 , and the Gamma-based mean rule R_3 in the two-population case. This section numerically compares these operating characteristics over selected values of the sample size n , the target probability P^* , and the separation parameter

$$\delta = \gamma_{[2]} - \gamma_{[1]}.$$

The purpose is not to give a formal analytic proof of a uniform ordering. Instead, the goal is to examine how the probability of correct selection, the probability of incorrect selection, and the expected subset size behave across representative finite-sample settings.

The numerical verification is carried out for

$$n = 5, 25, 100$$

and

$$P^* = 0.95, 0.99.$$

The value $P^* = 0.95$ provides continuity with the operating-characteristic comparisons considered previously in the thesis, while $P^* = 0.99$ represents a more stringent probability-of-correct-selection requirement. For each pair (n, P^*) , the rules are compared over the grid

$$\delta = 0, 0.1, 0.2, \dots, 2.0.$$

For $j = 2, 3$, define the component-wise differences

$$\Delta_{\text{PCS},j}(\delta) = \Pr(\text{CS}_1) - \Pr(\text{CS}_j),$$

$$\Delta_{\text{PICS},j}(\delta) = \Pr(\text{ICS}_1) - \Pr(\text{ICS}_j),$$

and

$$\Delta_{\text{ESS},j}(\delta) = \text{ESS}_1 - \text{ESS}_j.$$

The numerical comparisons are assessed using the sign conditions

$$\Delta_{\text{PCS},j}(\delta) \geq 0, \quad \Delta_{\text{PICS},j}(\delta) \leq 0, \quad \Delta_{\text{ESS},j}(\delta) \leq 0.$$

Together, these inequalities describe the desired comparison of R_1 with rule R_j in terms of the three operating characteristics. Specifically, the first inequality requires that R_1 have probability of correct selection at least as large as that of R_j . The second requires that R_1 have probability of incorrect selection no larger than that of R_j . The third requires that R_1 have expected subset size no larger than that of R_j .

The expected subset size comparison is checked directly because

$$\Delta_{\text{ESS},j}(\delta) = \Delta_{\text{PCS},j}(\delta) + \Delta_{\text{PICS},j}(\delta).$$

Thus, a positive PCS difference and a negative PICS difference contribute in opposite directions. In many cases, the negative PICS difference dominates, producing a smaller expected subset size for R_1 . However, if both incorrect-selection probabilities are essentially zero, the PICS difference contributes very little, and the ESS difference can be driven by very small PCS differences. This is why the ESS comparison is interpreted as part of the broader operating-characteristic comparison rather than as a uniform analytic inequality.

A strict numerical tolerance of 10^{-10} was used in the inequality checks to avoid classifying floating-point round-off error as a substantive violation. Thus, the numerical indicators were computed as

$$\Delta_{\text{PCS},j}(\delta) \geq -10^{-10},$$

$$\Delta_{\text{PICS},j}(\delta) \leq 10^{-10}$$

and

$$\Delta_{\text{ESS},j}(\delta) \leq 10^{-10}$$

5.4.1 Small-Sample Case: $n = 5$

The case $n = 5$ is useful for illustrating the finite-sample distinction between the CLT-based rule R_2 and the Gamma-based rule R_3 . Since R_2 relies on a normal approximation to the distribution of the sample means, its operating-characteristic comparisons may be less stable when the sample size is small. In contrast, R_3 evaluates the operating characteristics using the exact Gamma distribution of the sample sums.

5.4.1.1 Comparison of R_1 and the CLT-Based Rule R_2 For $P^* = 0.95$, the comparison between R_1 and R_2 shows the main pattern that R_1 has substantially smaller incorrect-selection probability and expected subset size for most positive values of δ . However, the PCS difference becomes slightly negative for some larger values of δ . This indicates that the desired PCS ordering,

$$\Pr(\text{CS}_1) \geq \Pr(\text{CS}_2),$$

does not hold uniformly under the CLT approximation in this small-sample case. The corresponding PICS and ESS inequalities continue to hold over the grid because the reduction in incorrect-selection probability under R_1 dominates the small PCS differences.

For $P^* = 0.99$, the same finite-sample issue is more visible. The stricter PCS requirement produces a more extreme CLT cutoff, and the PCS difference for R_1 versus R_2 becomes negative over a wider range of larger δ values. Nevertheless, the PICS and ESS comparisons still favor R_1 , because the CLT-based rule continues to include the non-best population with larger probability than R_1 .

Table 5.4 reports the rows from the inequality-check table where the PCS ordering fails for the CLT-based comparison. For $P^* = 0.95$, the violation occurs only for larger separation values, beginning at $\delta = 1.6$. For $P^* = 0.99$, the violation begins earlier, at $\delta = 0.7$, and continues through the rest of the grid. In both cases, the PICS and ESS inequalities continue to hold. Thus, the small-sample issue is specific to the PCS comparison under the CLT approximation. The complete tables for $P^* = 0.95$ and $P^* = 0.99$ are found in Appendix C.1.1–C.1.2.

Table 5.4: Selected PCS-violation rows for R_1 versus the CLT-based rule R_2 , $n = 5$.

P^*	δ	$\Delta_{PCS,2}$	Ineq $_{PCS}$ holds	$\Delta_{PICS,2}$	Ineq $_{PICS}$ holds	$\Delta_{ESS,2}$	Ineq $_{ESS}$ holds
0.95	1.6	-1.85259×10^{-6}	FALSE	-1.86409×10^{-1}	TRUE	-1.86411×10^{-1}	TRUE
0.95	1.7	-2.81183×10^{-6}	FALSE	-1.47438×10^{-1}	TRUE	-1.47441×10^{-1}	TRUE
0.95	1.8	-2.62441×10^{-6}	FALSE	-1.14220×10^{-1}	TRUE	-1.14223×10^{-1}	TRUE
0.95	1.9	-2.07499×10^{-6}	FALSE	-8.66499×10^{-2}	TRUE	-8.66520×10^{-2}	TRUE
0.95	2.0	-1.50444×10^{-6}	FALSE	-6.43528×10^{-2}	TRUE	-6.43543×10^{-2}	TRUE
0.99	0.7	-3.66243×10^{-6}	FALSE	-2.19837×10^{-1}	TRUE	-2.19841×10^{-1}	TRUE
0.99	0.8	-1.86142×10^{-5}	FALSE	-3.97863×10^{-1}	TRUE	-3.97882×10^{-1}	TRUE
0.99	0.9	-2.24513×10^{-5}	FALSE	-5.39098×10^{-1}	TRUE	-5.39120×10^{-1}	TRUE
0.99	1.0	-2.07489×10^{-5}	FALSE	-6.03479×10^{-1}	TRUE	-6.03499×10^{-1}	TRUE
0.99	1.1	-1.69133×10^{-5}	FALSE	-6.19261×10^{-1}	TRUE	-6.19278×10^{-1}	TRUE
0.99	1.2	-1.27721×10^{-5}	FALSE	-6.04063×10^{-1}	TRUE	-6.04076×10^{-1}	TRUE
0.99	1.3	-9.15002×10^{-6}	FALSE	-5.69168×10^{-1}	TRUE	-5.69177×10^{-1}	TRUE
0.99	1.4	-6.30534×10^{-6}	FALSE	-5.22090×10^{-1}	TRUE	-5.22096×10^{-1}	TRUE
0.99	1.5	-4.21760×10^{-6}	FALSE	-4.68083×10^{-1}	TRUE	-4.68087×10^{-1}	TRUE
0.99	1.6	-2.75624×10^{-6}	FALSE	-4.10996×10^{-1}	TRUE	-4.10998×10^{-1}	TRUE
0.99	1.7	-1.76853×10^{-6}	FALSE	-3.53743×10^{-1}	TRUE	-3.53745×10^{-1}	TRUE
0.99	1.8	-1.11854×10^{-6}	FALSE	-2.98550×10^{-1}	TRUE	-2.98552×10^{-1}	TRUE
0.99	1.9	-6.99550×10^{-7}	FALSE	-2.47073×10^{-1}	TRUE	-2.47074×10^{-1}	TRUE
0.99	2.0	-4.33746×10^{-7}	FALSE	-2.00462×10^{-1}	TRUE	-2.00463×10^{-1}	TRUE

Figure 5.1 display the difference plots for $n = 5$. The CLT-based PCS difference crosses below zero for some larger separation values, while the Gamma-based comparison preserves the desired sign pattern. In both comparisons, the PICS difference is negative, showing that R_1 eliminates the non-best population more rapidly.

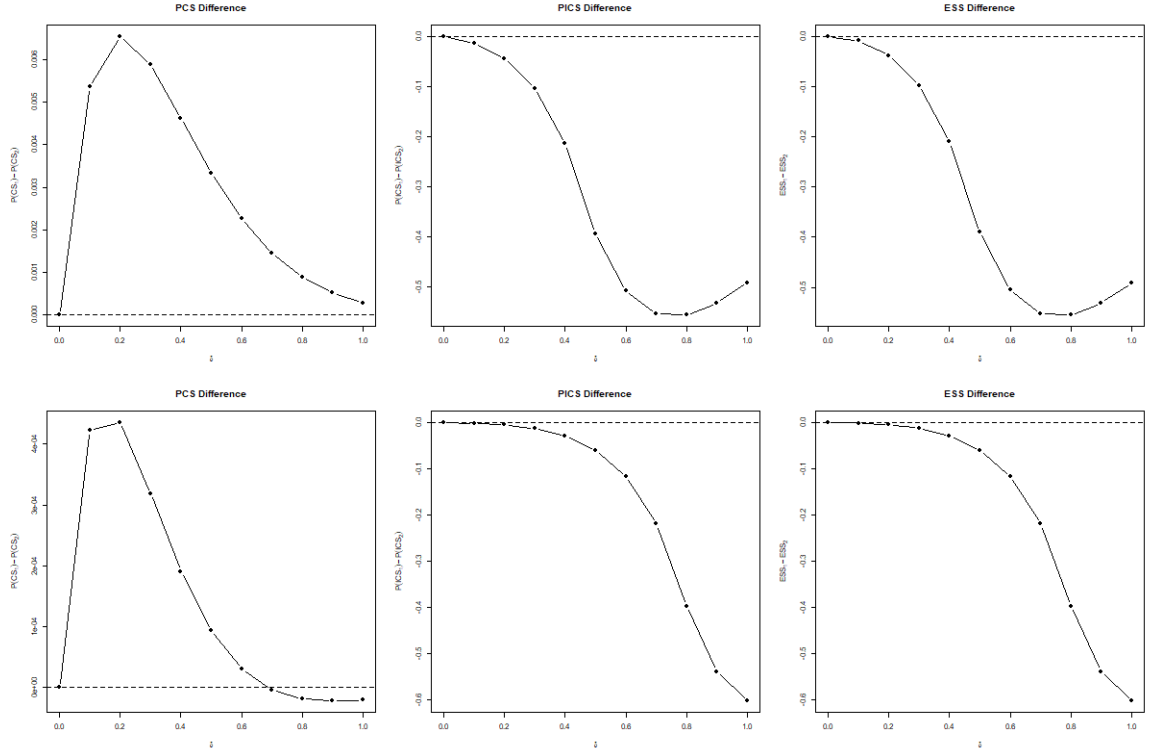


Figure 5.1: Difference plots for R_1 versus R_2 $k = 2$, $n = 5$, $P^* = 0.95$ and $P^* = 0.99$

5.4.1.2 Comparison of R_1 and the Gamma-Based Rule R_3 The Gamma-based comparison is more stable in this small-sample setting. For both $P^* = 0.95$ and $P^* = 0.99$, the numerical checks for R_1 versus R_3 show no violations of the PCS, PICS, or ESS

inequalities over the grid considered. Since no sign violations occur in these two cases, only the difference plots are shown in the main text, while the complete inequality-check tables are provided in Appendix C.1.3–C.1.4. This supports the use of the Gamma-based operating characteristics as a more appropriate finite-sample comparison than the CLT approximation when n is small.

The difference plots in Figure 5.2 show that the desired sign pattern is preserved for the Gamma-based comparison. In particular, the PCS difference remains nonnegative, while the PICS difference is negative for positive separation values. The ESS difference is also non-positive, reflecting the faster elimination of the non-best population under R_1 .

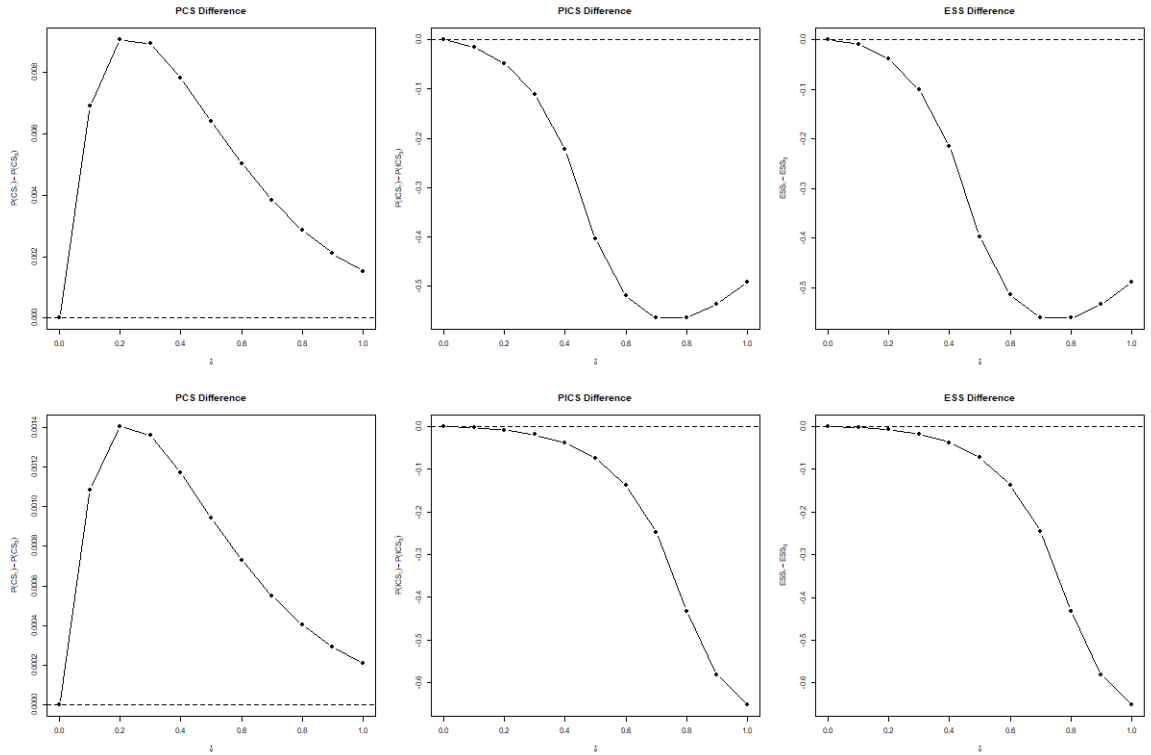


Figure 5.2: Difference plots for R_1 versus R_3 , $k = 2$, $n = 5$, $P^* = 0.95$, $P^* = 0.99$.

5.4.2 Moderate-Sample Case: $n = 25$

The case $n = 25$ gives a moderate-sample comparison. In this setting, the CLT approximation is more stable than in the $n = 5$ case, and both the CLT-based and Gamma-based comparisons show the same main operating-characteristic pattern. The complete inequality-check tables for the cases discussed in this subsection are provided in Appendix C.1.5–C.1.8.

5.4.2.1 Comparison of R_1 and the CLT-Based Rule R_2 For $P^* = 0.95$, the comparison between R_1 and the CLT-based rule R_2 show no violations of the PCS, PICS, or ESS inequalities over the grid considered. As δ increases R_1 reaches PCS close to one quickly, while also reducing the probability of incorrect selection more sharply than the CLT-based mean rule. Consequently, the ESS for R_1 approaches one earlier than the ESS for R_2 .

The same qualitative behavior for $P^* = 0.99$. The stricter PCS requirement increases the calibration constant for the CLT-based rule, but the numerical comparison remains stable. The PCS difference remains nonnegative, while the PICS and ESS differences remain non-positive over the grid. Thus, unlike the small-sample case $n = 5$, the PCS differences do not show the same instability, and the CLT-based comparison does not show PCS sign violations for $n = 25$. The main comparison remains the same: R_1 is more decisive because it has a much smaller probability of including the non-best population once the two thresholds are separated.

Figure 5.3 shows the difference plots for R_1 versus R_2 when $n = 25$. The PCS differences remain nonnegative, while the PICS differences are strongly negative over the most informative part of the grid. The ESS differences are also non-positive, supporting the conclusion that R_1 has comparable or larger PCS and smaller expected subset size than R_2 over the selected grid.

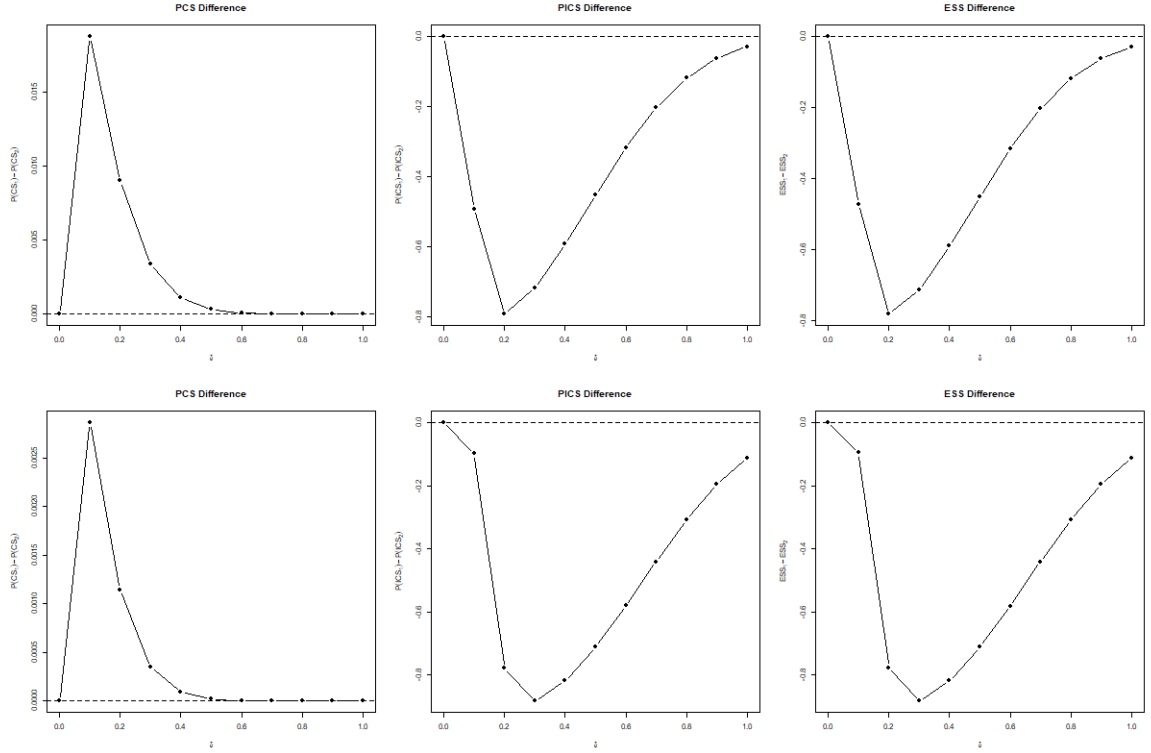


Figure 5.3: Difference plots for R_1 versus R_2 , $k = 2$, $n = 25$, $P^* = 0.95$, and $P^* = 0.99$.

5.4.2.2 Comparison of R_1 and the Gamma-Based Rule R_3 The Gamma-based comparison gives the same main conclusion for $n = 25$. For both $P^* = 0.95$ and $P^* = 0.99$, the numerical checks for R_1 versus R_3 show no violations of the PCS, PICS, or ESS inequalities over the grid considered. This indicates that the exact Gamma-based comparison preserves the desired sign pattern in the moderate-sample setting.

As in the CLT-based comparison, R_1 reduces the probability of including the non-best population more rapidly as the threshold separation increases. The PICS differences are negative over positive values of δ , and the ESS differences are also non-positive. Therefore, the moderate-sample results support the conclusion that R_1 is

more decisive than the Gamma-based mean rule while maintaining at least comparable PCS over the selected grid.

Figure 5.4 shows the corresponding difference plots for R_1 versus R_3 . The plots confirm that the PCS, PICS, and ESS differences preserve the desired signs for both probability requirements.

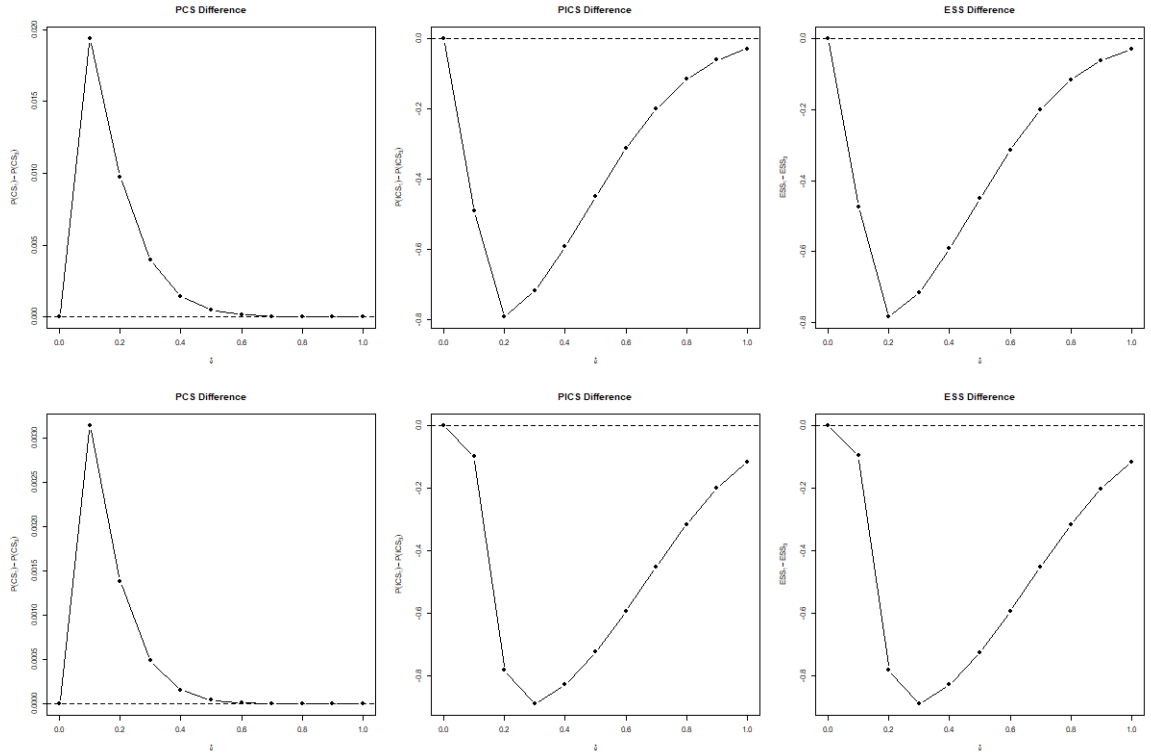


Figure 5.4: Difference plots for R_1 versus R_3 , $k = 2$, $n = 25$, $P^* = 0.95$, and $P^* = 0.99$.

5.4.3 Large-Sample Case: $n = 100$

The case $n = 100$ illustrates the large-sample behavior of the operating-characteristic comparisons. In this setting, all three rules become highly effective at identifying the best population as δ increases. The probability of correct selection rapidly approaches one, and the probability of incorrect selection can become extremely close to zero. The complete inequality-check tables for the cases discussed in this subsection are provided in Appendix C.1.9–C.1.12.

5.4.3.1 Comparison of R_1 and the CLT-Based Rule R_2 For the CLT-based comparison, the numerical results for $P^* = 0.95$ and $P^* = 0.99$ show the same broad pattern as in the moderate-sample case. Rule R_1 has comparable or larger PCS and a much smaller PICS over the main part of the grid. Hence, R_1 reaches an expected subset size close to one earlier than R_2 .

For both probability requirements, the PCS differences remain nonnegative, while the PICS and ESS differences remain non-positive over the grid considered. Thus, the CLT-based comparison shows no violations of the PCS, PICS, or ESS inequalities for $n = 100$. The complete inequality-check tables for R_1 versus R_2 are given in Appendix C.1.9–C.1.10.

Figure 5.5 shows the difference plots for R_1 versus R_2 when $n = 100$. The plots show that the main distinction between the rules is the faster reduction of incorrect selection under R_1 , while the PCS differences are either positive or numerically negligible.

5.4.3.2 Comparison of R_1 and the Gamma-Based Rule R_3 The Gamma-based comparison requires a more careful interpretation in the large-sample case. For $P^* = 0.95$,

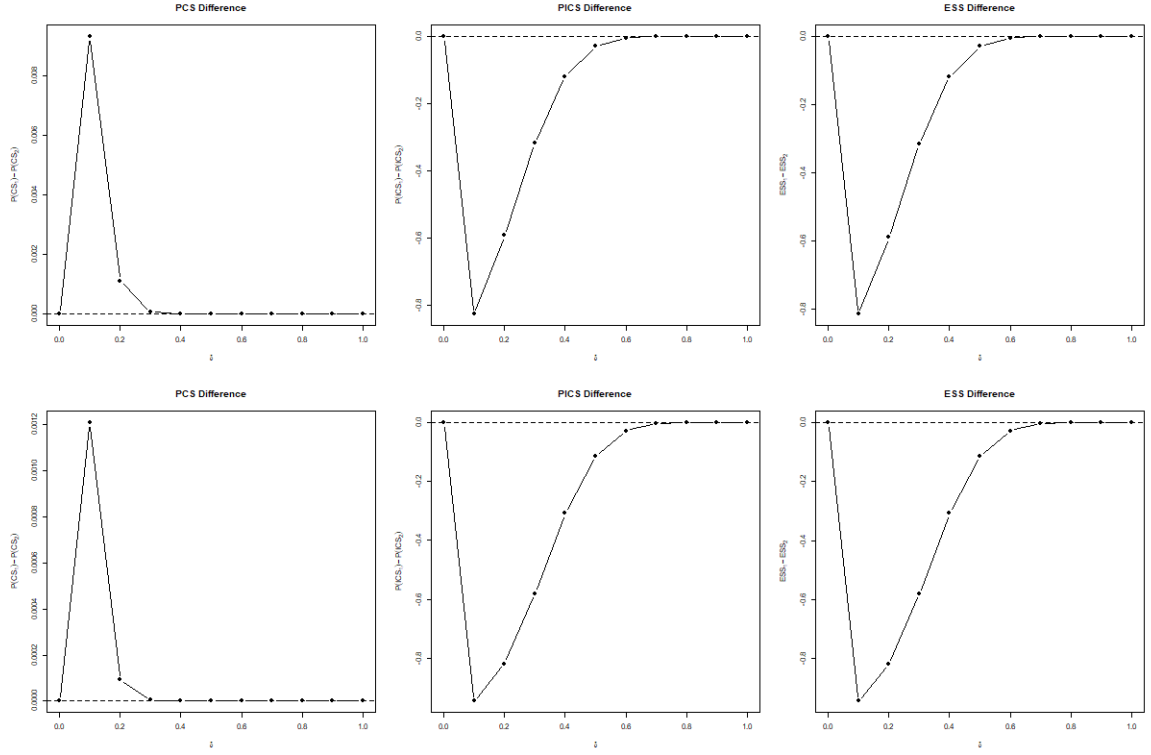


Figure 5.5: Difference plots for R_1 versus R_2 , $k = 2$, $n = 100$, $P^* = 0.95$, and $P^* = 0.99$.

the ESS inequality

$$\text{ESS}_1 \leq \text{ESS}_3$$

does not hold uniformly over the full grid under the numerical tolerance used in the inequality checks. Specifically, local ESS reversals occur at $\delta = 0.6$ and $\delta = 0.7$, as shown in Table 5.5. These reversals occur only after the probability of incorrect selection has essentially vanished for both rules. In this region, the PICS difference is approximately zero,

$$\Delta_{\text{PICS},3}(\delta) \approx 0,$$

so the ESS difference is determined almost entirely by the small positive PCS difference,

$$\Delta_{\text{PCS},3}(\delta) > 0.$$

Since

$$\Delta_{\text{ESS},3}(\delta) = \Delta_{\text{PCS},3}(\delta) + \Delta_{\text{PICS},3}(\delta),$$

the ESS difference can become positive even though the magnitude of the reversal is very small. Thus, the local ESS reversal should not be interpreted as a substantive loss of efficiency for R_1 . Rather, it reflects the fact that, at these separation values, both rules already have PCS close to one and PICS close to zero, so the remaining differences are very small.

Table 5.5: Selected ESS-violation rows for R_1 versus the Gamma-based rule R_3 , $n = 100$, $P^* = 0.95$.

δ	$\Delta_{\text{PCS},3}$	Ineq _{PCS} holds	$\Delta_{\text{PICS},3}$	Ineq _{PICS} holds	$\Delta_{\text{ESS},3}$	Ineq _{ESS} holds
0.6	6.33641×10^{-9}	TRUE	-1.11100×10^{-15}	TRUE	6.33640×10^{-9}	FALSE
0.7	1.28341×10^{-10}	TRUE	-1.34791×10^{-23}	TRUE	1.28341×10^{-10}	FALSE

For $P^* = 0.99$, the Gamma-based comparison does not show substantive violations within the numerical tolerance used. At larger values of δ , the differences are essentially at round-off level because both rules have PCS approximately equal to one and PICS approximately equal to zero. The complete inequality-check table for $P^* = 0.99$ is also provided in Appendix C.1.12.

Figure 5.6 shows the corresponding difference plots for R_1 versus R_3 . These plots illustrate that the most meaningful distinction between the rules is again the faster reduction of incorrect selection under R_1 . The local ESS reversals occur only in regions where both procedures are already effectively selecting a single population.

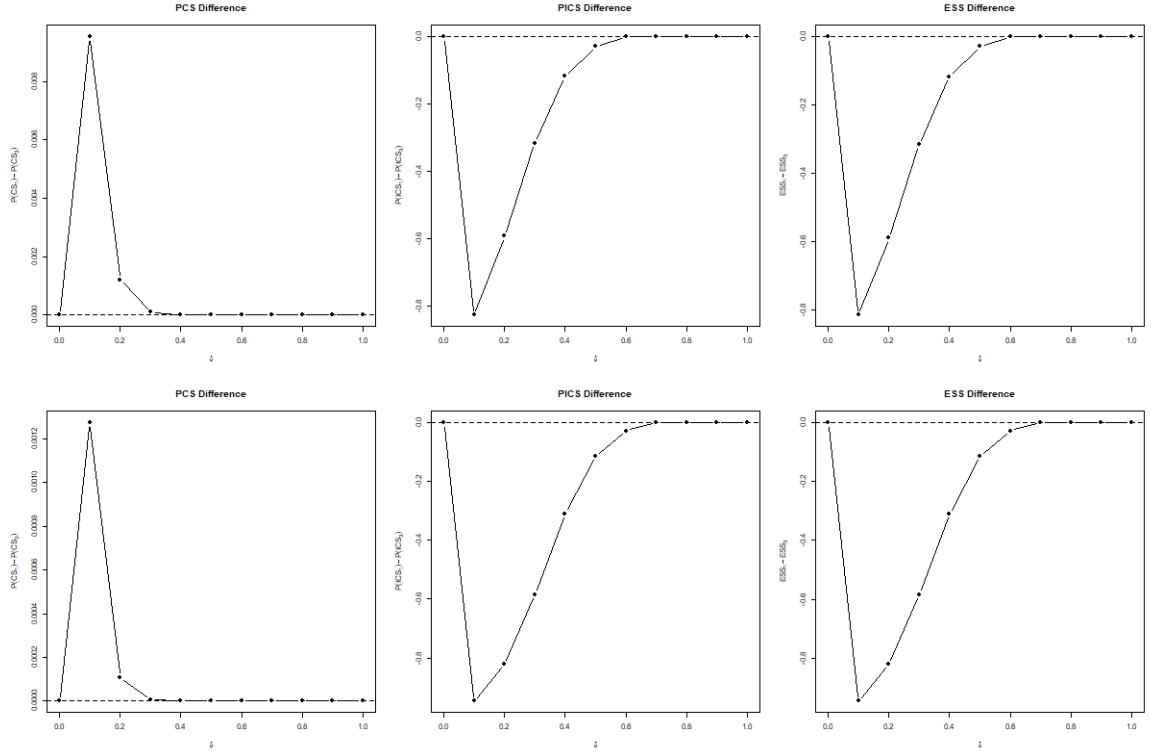


Figure 5.6: Difference plots for R_1 versus R_3 , $k = 2$, $n = 100$, $P^* = 0.95$, and $P^* = 0.99$.

As an additional check, the operating-characteristic differences were also evaluated over a broader grid of sample sizes $n = 5, 10, 15, \dots, 100$ and separation values $0 \leq \delta \leq 0.5$, with $P^* = 0.99$. The corresponding heat maps are provided in Appendix C.2.

These grid-based checks support the same conclusions as the fixed- n comparisons: the CLT-based comparison is most sensitive in the small-sample setting, the Gamma-based comparison is more stable in finite samples, and the main advantage of R_1 is its faster reduction of the probability of incorrect selection.

Overall, the numerical results support the conclusion that R_1 is generally more decisive than the mean-based rules in the two-population setting. The comparisons do not constitute a formal analytic proof of a uniform ordering, but they provide numerical evidence that the minimum-based rule tends to remove the inferior population more quickly while maintaining the calibrated probability of correct selection requirement at the least favorable configuration. The R-code to generate the results in Section 5.4 is found in Appendix B.13–B.16.

5.4.3.3 Numerical Stability and Tolerance Sensitivity Checks. Additional numerical checks were conducted to assess whether the inequality comparisons were sensitive to the numerical integration method and to the tolerance used for flagging violations. For the comparison of R_1 and the Gamma-based rule R_3 , the Gamma integrals were recomputed using a finite upper integration limit rather than direct integration over $[0, \infty)$. Specifically, the upper limit was chosen as

$$U = G^{-1}(1 - \varepsilon),$$

where G denotes the $\text{Gamma}(n, 1)$ distribution function and ε is a small tail probability. This finite upper limit was used because the contribution from the far right tail of the Gamma density is negligible, while direct integration to infinity can introduce unnecessary numerical instability. Since the integrands have the form $h(x)g(x)$, where $g(x)$ is the

Gamma($n, 1$) density and $0 \leq h(x) \leq 1$, the omitted tail is bounded by

$$\int_U^\infty h(x)g(x) dx \leq \int_U^\infty g(x) dx = \varepsilon.$$

Using this finite Gamma-quantile upper limit, no violations were observed for the R_1 versus R_3 comparison across $n = 5, \dots, 100$ and $P^* = 0.75, 0.90, 0.95, 0.99$, under either the strict tolerance 10^{-10} or the more practical tolerance 10^{-5} . Thus, the apparent ESS reversals observed under direct infinite-range integration are best interpreted as numerical integration artifacts rather than substantive violations of the ESS comparison.

For the R_1 versus R_2 comparison, the strict tolerance 10^{-10} flagged additional PCS violations for small values of n . These cases are summarized in Table 5.6. The violations occur only for the CLT-based rule and reflect small-sample limitations of the normal approximation. When the inequality checks for R_1 versus R_2 were repeated using

Table 5.6: Cases with PCS violations for R_1 versus the CLT-based rule R_2 under the strict tolerance 10^{-10} .

P^*	Values of n with PCS violations
0.90	5
0.95	5, 6, 7
0.99	5, 6, 7, 8, 9

the more practical tolerance 10^{-5} , the only remaining violations occurred in the PCS comparison for $n = 5$, $P^* = 0.99$, at

$$\delta = 0.7, 0.8, 0.9, 1.0, 1.1, 1.2.$$

The corresponding rows are shown in Table 5.7. This suggests that most violations flagged under the stricter tolerance are extremely small in magnitude, while the remaining cases identify the strongest small-sample departure of the CLT-based rule R_2 from the exact minimum-based rule R_1 .

Table 5.7: Selected PCS-violation rows for R_1 vs R_2 , $n = 5$, $P^* = 0.99$, $\text{tol} = 10^{-5}$.

δ	$\Delta_{\text{PCS},2}$	Ineq _{PCS} holds	$\Delta_{\text{PICS},2}$	Ineq _{PICS} holds	$\Delta_{\text{ESS},2}$	Ineq _{ESS} holds
0.8	-1.86142×10^{-5}	FALSE	-3.97863×10^{-1}	TRUE	-3.97882×10^{-1}	TRUE
0.9	-2.24513×10^{-5}	FALSE	-5.39098×10^{-1}	TRUE	-5.39120×10^{-1}	TRUE
1.0	-2.07489×10^{-5}	FALSE	-6.03479×10^{-1}	TRUE	-6.03499×10^{-1}	TRUE
1.1	-1.69133×10^{-5}	FALSE	-6.19261×10^{-1}	TRUE	-6.19278×10^{-1}	TRUE
1.2	-1.27721×10^{-5}	FALSE	-6.04063×10^{-1}	TRUE	-6.04076×10^{-1}	TRUE

5.5 Results and Takeaways

This chapter examined the two-population case in detail. The main contribution was to derive exact operating characteristics for the minimum-based rule R_1 , the CLT-based mean rule R_2 , and the Gamma-based mean rule R_3 . For $k = 2$, the probability of correct selection, probability of incorrect selection, and expected subset size can be expressed directly as functions of the separation

$$\delta = \gamma_{[2]} - \gamma_{[1]}.$$

This makes the two-population setting useful for understanding how the different rules behave as the best population becomes increasingly separated from the inferior population.

The exact formulas show that all three rules can be calibrated to attain the required probability of correct selection at the least favorable configuration, where $\delta = 0$. Thus, the

most important practical distinction among the rules is not whether they satisfy the target PCS requirement at the calibration point, but how quickly they remove the inferior population once $\delta > 0$. Across the numerical comparisons, the minimum-based rule R_1 generally has a much smaller probability of incorrect selection than the mean-based rules. Consequently, its expected subset size decreases toward one more rapidly, indicating that R_1 tends to make a more decisive selection once the two threshold parameters are separated.

The comparison between R_1 and the CLT-based rule R_2 also shows the limitations of using a normal approximation in small samples. Under the strict tolerance 10^{-10} , PCS violations were flagged only for small values of n , especially when P^* was large. When the inequality checks were repeated using the more practical tolerance 10^{-5} , the only remaining PCS violations occurred for $n = 5$, $P^* = 0.99$, at a small number of δ values. Even in these cases, R_1 continues to have a smaller probability of incorrect selection and a smaller expected subset size over the grid considered. This indicates that the small-sample issue is mainly associated with the PCS comparison under the CLT approximation, rather than with the overall decisiveness of R_1 .

The Gamma-based rule R_3 provides a more stable finite-sample comparison than the CLT-based rule. Since R_3 uses the exact Gamma distribution of the sample sums, it avoids the small-sample approximation issue observed for R_2 . In the final numerical verification, the Gamma integrals were evaluated using a finite Gamma-quantile upper limit rather than direct integration to infinity. With this numerically stable implementation, no violations were observed for the R_1 versus R_3 comparison across $n = 5, \dots, 100$, $P^* = 0.75, 0.90, 0.95, 0.99$, and both tolerances 10^{-10} and 10^{-5} . Thus, the earlier apparent ESS reversals under direct infinite-range integration are best interpreted as numerical artifacts rather than substantive violations of the ESS comparison.

For $n = 25$, both mean-based comparisons become stable, and R_1 has comparable or larger PCS, smaller PICS, and smaller ESS than both R_2 and R_3 . For $n = 100$, all three rules become highly effective as δ increases, and most differences become numerically negligible once PCS is near one and PICS is near zero. The additional numerical stability checks support the conclusion that the Gamma-based comparison is stable when the tail integration is handled using the finite Gamma-quantile upper limit.

Overall, the results support three main conclusions. First, the minimum-based rule R_1 is generally more decisive than the mean-based rules in the two-population exponential threshold setting. Second, the main advantage of R_1 is its faster reduction of the probability of incorrect selection, which leads to smaller expected subset sizes for most positive values of δ . Third, when a mean-based rule is used in finite samples, the Gamma-based operating characteristics provide a more reliable comparison than the CLT approximation, particularly when n is small.

These findings do not constitute a formal analytic proof of a uniform ordering among the rules over all possible parameter values. Instead, they provide exact and numerical evidence that, in the two-population case, R_1 maintains the calibrated PCS requirement while tending to exclude the inferior population more quickly than the mean-based alternatives. This makes the minimum-based procedure especially attractive when the goal is not only to include the best population with high probability, but also to avoid unnecessarily large selected subsets.

CHAPTER 6

BAYESIAN SUBSET SELECTION FOR THREE-PARAMETER WEIBULL POPULATIONS

6.1 Introduction

The subset selection procedures developed in the previous chapters were constructed from a frequentist point of view. In that framework, the objective is to select a subset of populations that contains the best population with probability at least P^* , where the probability of correct selection is evaluated with respect to repeated sampling. Such procedures are closely connected to the threshold-based ranking and subset selection framework developed in the previous chapters [15, 16].

In this chapter, a Bayesian approach is developed for subset selection among populations modeled by the three-parameter Weibull distribution. Rather than constructing a selection rule only from the sampling distribution of a statistic, the Bayesian approach uses the posterior distribution of the unknown model parameters. This allows uncertainty about the model parameters and uncertainty about the identity of the best population to be quantified directly after observing the data.

For population i , let γ_i denote the threshold parameter of the three-parameter Weibull distribution. The best population is defined as the population with the largest threshold parameter. That is, population i is best if

$$\gamma_i = \max_{1 \leq j \leq k} \gamma_j.$$

This definition is consistent with the threshold-based selection framework considered in the previous chapters. In lifetime and reliability applications, the threshold parameter represents a lower endpoint or minimum lifetime before failures can occur. Therefore, a

larger threshold parameter shifts the lifetime distribution to the right and indicates a larger minimum lifetime.

The three-parameter Weibull distribution is especially useful in reliability analysis because it includes a threshold parameter and can represent different hazard-rate shapes through its shape parameter. When the shape parameter is less than one, the hazard rate is decreasing; when it is equal to one, the hazard rate is constant; and when it is greater than one, the hazard rate is increasing. This flexibility makes the Weibull model appropriate for fatigue-life, degradation, and other reliability settings [11, 14].

The Bayesian procedure in this chapter assigns a posterior probability to the event that each population is best. These posterior probabilities are then used to construct a selected subset whose cumulative posterior probability of containing the best population is at least P^* . In this sense, the Bayesian procedure has a direct post-data interpretation: conditional on the observed data and the assumed prior model, the selected subset contains the best population with posterior probability at least P^* .

Two applications are considered. The first application uses five fully simulated three-parameter Weibull populations. These populations are designed to represent similar degrading materials with ordered threshold parameters, mildly varying scale parameters, and shape parameters greater than one. Thus, all five populations have increasing hazard functions, which is consistent with reliability settings where failure risk increases as the material ages or degrades. The purpose of the study is not to create widely separated populations, but rather to examine the Bayesian subset selection procedure in a setting where the populations are similar while the best population is still determined by the largest threshold parameter. Since the true parameter values are known, this study provides a controlled setting for evaluating the Bayesian subset selection procedure.

The second application uses fatigue-life data from Alloy T7987. This dataset is a right-censored reliability dataset reported by Meeker et al.,[13] and is included in the Statistical Methods for Reliability data, 2nd edition, reliability data collection. The observations represent fatigue lives of alloy specimens, measured in thousands of cycles. In the dataset used here, 67 specimens failed during testing and five additional specimens were tested up to 300 thousand cycles without failure. These five observations are therefore treated as right-censored observations, since their exact failure times are known only to exceed 300 thousand cycles.

First, a three-parameter Weibull model is fit to the original alloy fatigue data using maximum likelihood and Bayesian methods. The fitted model is then used as the baseline for a semi-simulated five-population censored study. In that study, the first population is based on the original fitted alloy model, while the remaining four populations are generated to be similar but with larger threshold parameters. This second application connects the Bayesian subset selection procedure to a more realistic reliability-data setting involving right censoring.

This chapter is organized as follows. Section 6.2 introduces the Bayesian subset selection framework and presents the Bayesian subset selection rule. Section 6.3 presents the three-parameter Weibull model for exact and right-censored lifetime data. Section 6.4 describes the prior distributions. Section 6.5 discusses posterior computation using Metropolis–Hastings sampling. Section 6.6 applies the method to the fully simulated Weibull study. Section 6.7 presents the fitted Weibull analysis for the original alloy fatigue data. Section 6.8 applies the subset selection procedure to the semi-simulated alloy fatigue study. Section 6.9 summarizes the main results and takeaways.

6.2 Bayesian Subset Selection Framework and Decision Rule

6.2.1 Classical Subset Selection Background

Classical subset selection procedures are designed to select a subset of populations that contains the best population with a guaranteed probability of correct selection. In the ranking and selection literature, this guarantee is usually expressed as

$$P(\text{CS}) \geq P^*,$$

where CS denotes the event of correct selection and P^* is a pre-specified probability level.

The procedures developed in the previous chapters follow this frequentist interpretation. In that setting, the probability of correct selection is a pre-data probability computed over repeated samples under fixed parameter values. Foundational work on ranking and selection includes procedures for selecting the best population and for selecting a subset that contains the best population [1, 2, 3, 4].

For Weibull populations, subset selection and best-population selection have been studied in several reliability contexts, including complete samples, censored samples, and Weibull populations ordered according to reliability-related parameters [24, 25]. In contrast to the frequentist approach, the Bayesian procedure developed in this chapter evaluates the probability that each population is best after observing the data.

The Bayesian framework is useful in the present setting because the identity of the best population depends on unknown Weibull threshold parameters. By sampling from the posterior distribution of the model parameters, the posterior probability that each population has the largest threshold parameter can be estimated directly.

6.2.2 Definition of the Best Population

Suppose there are k independent populations. For population i , let

$$\psi_i = (\gamma_i, \eta_i, \beta_i),$$

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

In this chapter, the best population is defined as the population with the largest threshold parameter. Thus, population i is best if

$$\gamma_i = \max_{1 \leq j \leq k} \gamma_j.$$

This choice is motivated by reliability applications in which a larger threshold parameter corresponds to a larger lower endpoint of the lifetime distribution. In other words, a larger value of γ_i shifts the lifetime distribution to the right and indicates a later earliest possible failure time.

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

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

Since the posterior distributions considered in this chapter are continuous, exact ties occur with probability zero in the retained posterior draws.

6.2.3 Posterior Probability of Being Best

Under the Bayesian model, the unknown parameters are treated as random quantities. After observing the data, uncertainty about the model parameters is represented by the posterior distribution

$$\pi(\psi_1, \dots, \psi_k \mid \mathbf{x}),$$

where $\mathbf{x}$ denotes the observed data from all populations.

The posterior probability that population i is best is defined as

$$p_i = P\left(\gamma_i = \max_{1 \leq j \leq k} \gamma_j \mid \mathbf{x}\right).$$

The values $p_1, \dots, p_k$ summarize the posterior evidence that each population is best. Since exactly one population is best at each posterior draw, these probabilities satisfy

$$\sum_{i=1}^k p_i = 1.$$

In practice, the posterior probabilities are estimated from Markov Chain Monte Carlo samples. Suppose that after burn-in and thinning, M posterior draws are retained:

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

Each posterior draw represents one plausible realization of the k threshold parameters from the joint posterior distribution. For the m th draw, the population with the largest sampled threshold parameter is recorded as the best population. Repeating this comparison over all retained posterior draws gives an estimate of how often each population is best under the posterior distribution.

Thus, the posterior probability that population i is best is estimated by

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

where $I(\cdot)$ is the indicator function. The indicator equals one when population i has the largest sampled threshold parameter in draw m , and equals zero otherwise. Therefore, $\hat{p}_i$ is the proportion of retained posterior draws in which population i is selected as best.

6.2.4 Posterior Probability of Correct Selection

Let $\mathcal{S}$ denote a selected subset of populations. The selected subset contains the best population if

$$i^* \in \mathcal{S}.$$

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

$$P(\text{CS} \mid \mathbf{x}) = P(i^* \in \mathcal{S} \mid \mathbf{x}).$$

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

$$P(\text{CS} \mid \mathbf{x}) = \sum_{i \in \mathcal{S}} p_i.$$

This quantity is different from the frequentist probability of correct selection. The frequentist probability of correct selection is a pre-data probability computed over repeated samples under fixed parameter values. In contrast, the Bayesian posterior probability of correct selection is conditional on the observed data and the assumed prior model.

6.2.5 Bayesian Selected Subset

Let P^* be the desired posterior probability level. The Bayesian selected subset is constructed by ordering the estimated posterior probabilities of being best from largest to smallest:

$$\hat{p}_{(1)} \geq \hat{p}_{(2)} \geq \cdots \geq \hat{p}_{(k)}.$$

Here, $\hat{p}_{(r)}$ denotes the r th largest estimated posterior probability, and $i_{(r)}$ denotes the population index associated with that probability.

The Bayesian selected subset is

$$\mathcal{S}_B = \{i_{(1)}, \dots, i_{(m)}\},$$

where m is the smallest integer satisfying

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

Thus, $\mathcal{S}_B$ is the smallest subset, based on posterior probability ordering, whose posterior probability of containing the best population is at least P^* .

This rule has a direct interpretation. If one population has an overwhelming posterior probability of being best, the selected subset may contain only that population. If posterior uncertainty is larger, the selected subset expands to include additional populations until the desired posterior probability level is reached.

6.2.6 Percentile-Based Reliability Comparisons

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

$$t_{p,i} = \gamma_i + \eta_i [-\log(1-p)]^{1/\beta_i}.$$

In this chapter, the percentiles

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

are examined for each population. These quantities provide a reliability interpretation because $t_{p,i}$ represents the time by which a proportion p of units from population i is expected to have failed.

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

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

This calculation is applied to $p = 0.10$, $p = 0.50$, and $p = 0.90$.

These percentile-based comparisons are not used to define the formal Bayesian selected subset, which is based on the threshold parameter γ_i . Instead, they are used as supplementary reliability summaries. They help assess whether the population with the largest threshold parameter also tends to have larger lifetime percentiles under the posterior distribution.

6.3 Bayesian Model Specification

6.3.1 Three-Parameter Weibull Model for Exact Failure Times

For population i , let $X_{i1}, \dots, X_{in}$ denote independent exact lifetime observations from a three-parameter Weibull distribution with threshold γ_i , scale η_i , and shape β_i . The density is

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

where $\eta_i > 0$ and $\beta_i > 0$.

The distribution function is

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

and the survival function is

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

The hazard function is

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

Thus, the hazard rate is decreasing when $\beta_i < 1$, constant when $\beta_i = 1$, and increasing when $\beta_i > 1$.

For exact lifetime data from population i , the likelihood is

$$L_i(\gamma_i, \eta_i, \beta_i | \mathbf{x}_i) = \prod_{j=1}^n f_i(x_{ij} | \gamma_i, \eta_i, \beta_i), \quad (6.1)$$

with the restriction that

$$\gamma_i < x_{i(1)},$$

where

$$x_{i(1)} = \min_{1 \leq j \leq n} x_{ij}.$$

The corresponding log-likelihood is

$$\ell_i(\gamma_i, \eta_i, \beta_i) = \sum_{j=1}^n \left[\log(\beta_i) - \beta_i \log(\eta_i) + (\beta_i - 1) \log(x_{ij} - \gamma_i) - \left(\frac{x_{ij} - \gamma_i}{\eta_i} \right)^{\beta_i} \right]. \quad (6.2)$$

6.3.2 Three-Parameter Weibull Model for Right-Censored Data

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

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

The likelihood contribution for observation j in population i is

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

Therefore, the likelihood for population i is

$$L_i(\gamma_i, \eta_i, \beta_i \mid \mathbf{t}_i, \mathbf{d}_i) = \prod_{j=1}^n [f_i(t_{ij} \mid \gamma_i, \eta_i, \beta_i)]^{d_{ij}} [S_i(t_{ij} \mid \gamma_i, \eta_i, \beta_i)]^{1-d_{ij}}. \quad (6.3)$$

The log-likelihood is

$$\begin{aligned} \ell_i(\gamma_i, \eta_i, \beta_i) &= \sum_{j=1}^n d_{ij} \log f_i(t_{ij} \mid \gamma_i, \eta_i, \beta_i) + \sum_{j=1}^n (1 - d_{ij}) \log S_i(t_{ij} \mid \gamma_i, \eta_i, \beta_i). \\ &= \sum_{j=1}^n d_{ij} [\log(\beta_i) - \beta_i \log(\eta_i) + (\beta_i - 1) \log(t_{ij} - \gamma_i)] - \sum_{j=1}^n \left(\frac{t_{ij} - \gamma_i}{\eta_i} \right)^{\beta_i}. \end{aligned} \quad (6.4)$$

This likelihood reduces to the exact-data likelihood when all $d_{ij} = 1$.

6.3.3 Transformation Used for Posterior Computation

The threshold parameter must satisfy $\gamma_i < x_{i(1)}$ for exact data and $\gamma_i < t_{i(1)}$ for censored data. To write this constraint in a common form for both exact and right-censored data, let y_{ij} denote the observed time for unit j in population i , and define

$$y_{i(1)} = \min_{1 \leq j \leq n} y_{ij}.$$

Thus, the threshold parameter must satisfy

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

Bayesian computation texts commonly recommend transformations of constrained parameters to improve computational behavior and simplify posterior simulation [26, 27]. To enforce the threshold constraint during posterior simulation, the following transformation is used:

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

where

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

and ε is a small positive constant. Equivalently,

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

Because $\exp(\theta_{i3}) > 0$, this parameterization guarantees that $\gamma_i < y_{i(1)}$ for every proposed value of θ_{i3} .

6.4 Prior Specification

Prior specification is an important part of Bayesian Weibull modeling, especially for the three-parameter Weibull distribution. The scale parameter η_i and shape parameter β_i are positive, while the threshold parameter γ_i determines the lower endpoint of the distribution. Because the support of the distribution depends on γ_i , the threshold parameter requires particular care.

Bayesian reliability texts commonly emphasize the importance of incorporating prior information carefully, especially when lifetime data are limited or censored [28, 29]. The use of formal and weakly informative prior distributions has also been widely discussed in Bayesian statistics [30]. In this chapter, the prior distributions are chosen to respect the parameter constraints of the Weibull model while allowing substantial uncertainty in the parameter values.

Prior choices for Weibull models have been studied in several Bayesian and reliability contexts. Noninformative prior choices for Weibull distributions are discussed by Sun [31], and alternative Bayesian prior structures for Weibull estimation have also been considered in earlier reliability work [32].

In this chapter, the prior distributions are chosen to respect the parameter constraints of the Weibull model while allowing substantial uncertainty in the parameter values. Independent priors are assigned to the Weibull parameters for each population. Prior specification for Weibull models requires attention to the positivity constraints on the shape and scale parameters and to the threshold constraint on the location parameter. Because the shape and scale parameters must be positive, gamma priors are used for β_i and η_i . The threshold parameter is handled through the positive distance parameter δ . The prior is assigned to the transformed parameter

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

which places the threshold-distance parameter on an unconstrained real-valued scale.

For the fully simulated Weibull study, the priors are

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

$$\eta_i \sim \text{Gamma}(\alpha_{\eta_i} = 20, \theta_{\eta_i} = 0.05),$$

where the second argument denotes the scale parameter. These priors have prior means

$$E(\beta_i) = \alpha_{\beta_i} \theta_{\beta_i} = 1.5, \quad E(\eta_i) = \alpha_{\eta_i} \theta_{\eta_i} = 1.0.$$

They are centered near the generating values used in the simulation study, while still allowing the likelihood to update the parameter estimates based on the observed data. For the transformed threshold-distance parameter, the prior is

$$\theta_{i3} = \log(\delta_i) \sim N(\mu_{\theta_{i3}} = \log(0.05), \sigma_{\theta_{i3}}^2 = 1.0^2).$$

For the original alloy fatigue data and the semi-simulated alloy fatigue study, time is measured in thousands of cycles, so the scale parameter requires a prior that is

compatible with the larger lifetime scale. For the original Bayesian fit, the priors are

$$\beta \sim \text{Gamma}(\alpha_\beta = 6, \theta_\beta = 0.25),$$

$$\eta \sim \text{Gamma}(\alpha_\eta = 0.01, \theta_\eta = 100),$$

and

$$\theta_3 = \log(\delta) \sim N(\mu_{\theta_3} = \log(20), \sigma_{\theta_3}^2 = 10^2).$$

For the semi-simulated censored study, the priors are

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

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

and

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

The prior assigned to θ_{i3} is especially important because the threshold parameter is closely tied to the smallest observed lifetime. Larger values of δ_i correspond to smaller threshold values γ_i , while smaller values of δ_i place the threshold closer to the minimum observed time. Therefore, the prior on θ_{i3} controls how strongly the model allows the threshold to approach the lower end of the observed data.

For this reason, prior sensitivity analyses are performed by varying the prior standard deviation of θ_{i3} , following the general practice of assessing the stability of Bayesian conclusions under alternative prior specifications [33]. In the fully simulated study, the values

$$\sigma_{\theta_3} = 0.5, 1.0, 1.5$$

are considered. In the semi-simulated censored study, the values

$$\sigma_{\theta_3} = 1.0, 2.0, 3.0$$

are considered. These sensitivity analyses assess whether the posterior conclusions, including posterior summaries and the selected subset of best populations, are stable under different levels of prior uncertainty for the threshold parameter.

6.5 Posterior Computation

Posterior computation is carried out using Metropolis–Hastings simulation [34, 35, 36]. Two samplers are compared: a random-walk Metropolis sampler and an independence Metropolis–Hastings sampler. Both algorithms are implemented on the transformed parameter scale introduced in Section 6.3, so that the positivity constraints on the Weibull parameters and the threshold constraint are satisfied automatically during sampling.

For population i , let

$$\theta_i = (\theta_{i1}, \theta_{i2}, \theta_{i3})^\top = (\log \beta_i, \log \eta_i, \log \delta_i)^\top,$$

where $\beta_i > 0$ is the Weibull shape parameter, $\eta_i > 0$ is the Weibull scale parameter, and $\delta_i > 0$ is a positive gap parameter. If $y_{i(1)}$ denotes the smallest observed time in population i , the threshold parameter is written as

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

where $\varepsilon > 0$ is a small numerical constant. This reparameterization ensures that

$$\gamma_i < y_{i(1)}$$

for every posterior draw.

The posterior distribution is sampled on the transformed scale. After sampling, the draws are converted back to the original Weibull parameter scale using

$$\beta_i = \exp(\theta_{i1}), \quad \eta_i = \exp(\theta_{i2}), \quad \delta_i = \exp(\theta_{i3}), \quad \gamma_i = y_{i(1)} - \exp(\theta_{i3}) - \varepsilon.$$

Let

$$\ell_i(\theta_i) = \log \pi(\theta_i \mid \mathbf{x}_i)$$

denote the log-posterior density for population i on the transformed parameter scale. The Metropolis–Hastings algorithm generates a Markov Chain

$$\theta_i^{(0)}, \theta_i^{(1)}, \dots, \theta_i^{(M)}$$

by proposing a candidate value θ_i^* from a proposal density and accepting or rejecting this candidate according to a probability that preserves the posterior distribution as the stationary distribution of the chain.

6.5.1 Random-Walk Metropolis Sampler

The random-walk Metropolis sampler uses a local proposal distribution centered at the current value of the chain. At iteration m , a candidate value is generated as

$$\theta_i^* \sim N_3 \left(\theta_i^{(m-1)}, c^2 \hat{\Sigma}_{\text{pilot},i} \right),$$

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

Because the random-walk proposal is symmetric,

$$q(\theta_i^* \mid \theta_i^{(m-1)}) = q(\theta_i^{(m-1)} \mid \theta_i^*),$$

the proposal terms cancel from the Metropolis–Hastings ratio. Therefore, the acceptance probability is

$$\alpha_{\text{RW}} = \min \left\{ 1, \frac{\pi(\theta_i^* | \mathbf{x}_i)}{\pi(\theta_i^{(m-1)} | \mathbf{x}_i)} \right\}.$$

Equivalently, on the log scale,

$$\log \alpha_{\text{RW}} = \min \left\{ 0, \ell_i(\theta_i^*) - \ell_i(\theta_i^{(m-1)}) \right\}.$$

The pilot covariance matrix is computed from a short preliminary random-walk run. If M_p draws are retained from the pilot run, then

$$\hat{\Sigma}_{\text{pilot},i} = \frac{1}{M_p - 1} \sum_{m=1}^{M_p} \left(\theta_i^{(m)} - \bar{\theta}_i \right) \left(\theta_i^{(m)} - \bar{\theta}_i \right)^\top,$$

where

$$\bar{\theta}_i = \frac{1}{M_p} \sum_{m=1}^{M_p} \theta_i^{(m)}.$$

This covariance matrix estimates the posterior scale and dependence structure among the transformed parameters

$$\log \beta_i, \quad \log \eta_i, \quad \log \delta_i.$$

It is used only as a tuning device for the proposal distribution in the final MCMC run.

6.5.2 Independence Metropolis–Hastings Sampler

The independence Metropolis–Hastings sampler uses a proposal distribution that does not depend on the current state of the chain. A candidate value is generated from a fixed proposal density,

$$\theta_i^* \sim q_i(\theta_i).$$

The acceptance probability is

$$\alpha_I = \min \left\{ 1, \frac{\pi(\boldsymbol{\theta}_i^* | \mathbf{x}_i) q_i(\boldsymbol{\theta}_i^{(m-1)})}{\pi(\boldsymbol{\theta}_i^{(m-1)} | \mathbf{x}_i) q_i(\boldsymbol{\theta}_i^*)} \right\}.$$

Equivalently, on the log scale,

$$\log \alpha_I = \min \left\{ 0, \ell_i(\boldsymbol{\theta}_i^*) - \ell_i(\boldsymbol{\theta}_i^{(m-1)}) + \log q_i(\boldsymbol{\theta}_i^{(m-1)}) - \log q_i(\boldsymbol{\theta}_i^*) \right\}.$$

The independence proposal is centered near an approximate posterior mode. Let

$$\hat{\boldsymbol{\theta}}_{L,i} = \arg \max_{\boldsymbol{\theta}_i} \ell_i(\boldsymbol{\theta}_i)$$

denote the posterior mode on the transformed scale. The local curvature of the negative log-posterior at this mode is

$$H_i(\hat{\boldsymbol{\theta}}_{L,i}) = \left[\frac{\partial^2 [-\ell_i(\boldsymbol{\theta}_i)]}{\partial \boldsymbol{\theta}_{ir} \partial \boldsymbol{\theta}_{is}} \right]_{\boldsymbol{\theta}_i = \hat{\boldsymbol{\theta}}_{L,i}}.$$

The Laplace approximation to the posterior covariance matrix is then

$$\hat{\boldsymbol{\Sigma}}_{L,i} = H_i(\hat{\boldsymbol{\theta}}_{L,i})^{-1}.$$

In the implementation, the Laplace covariance matrix is checked before use. If it is numerically stable, it is used in the independence proposal. If it is not stable, the pilot covariance matrix from the preliminary random-walk run is used instead. Therefore, the proposal covariance is based on

$$\hat{\boldsymbol{\Sigma}}_{q,i} = \begin{cases} \hat{\boldsymbol{\Sigma}}_{L,i}, & \text{if the Laplace covariance is numerically stable,} \\ \hat{\boldsymbol{\Sigma}}_{\text{pilot},i}, & \text{otherwise.} \end{cases}$$

This covariance matrix is inflated to allow broader moves:

$$\boldsymbol{\Sigma}_{q,i} = \kappa \hat{\boldsymbol{\Sigma}}_{q,i},$$

where $\kappa > 1$ is an inflation factor.

To improve robustness, the independence proposal is taken to be a mixture of a multivariate normal distribution and a heavier-tailed multivariate t distribution:

$$q_i(\theta_i) = (1 - \omega)\phi_3(\theta_i; \mu_{q,i}, \Sigma_{q,i}) + \omega t_{v,3}(\theta_i; \mu_{q,i}, \Sigma_{q,i}),$$

where $\phi_3(\cdot; \mu, \Sigma)$ is a three-dimensional normal density, $t_{v,3}(\cdot; \mu, \Sigma)$ is a three-dimensional multivariate t density with v degrees of freedom, and ω is the mixture probability. The heavier-tailed component helps protect against proposals that are too narrow, especially when the posterior distribution is skewed or when the threshold parameter is sensitive to the smallest observed lifetime.

6.5.3 Implementation Details

For each population, starting values are obtained by fitting shifted Weibull models over a grid of possible values for δ_i . These starting values are used to initialize the transformed parameter vector

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

A preliminary random-walk run is then used to estimate the pilot covariance matrix. This pilot covariance matrix is used to tune the random-walk proposal. It is also used as a fallback covariance matrix for the independence sampler when the Laplace covariance matrix is numerically unstable.

In the main analyses, four independent chains are run for each population and for each sampler. Each chain is run for 200,000 iterations. The first 20,000 iterations are discarded as burn-in, and every third draw is retained. The retained draws from the four chains are then pooled. Because the populations are modeled independently and independent priors are assigned across populations, the joint posterior distribution factors

into the product of the population-specific posterior distributions. Therefore, joint posterior draws for subset selection are obtained by combining independently retained posterior draws from the fitted population-specific chains.

The two samplers are compared using standard MCMC diagnostics, including trace plots, autocorrelation plots, acceptance rates, effective sample sizes, and Gelman–Rubin $\hat{R}$ diagnostics [37, 38]. These diagnostics are computed separately for the random-walk sampler and the independence Metropolis–Hastings sampler.

6.6 Bayesian Subset Selection in a Weibull Simulation Study

6.6.1 Data-Generating Configuration

The first application consists of five independent three-parameter Weibull populations. The sample size is $n = 150$ for each population. The observations are generated from

$$X_{ij} = \gamma_i + W_{ij}, \quad W_{ij} \sim \text{Weibull}(\beta_i, \eta_i),$$

where β_i is the shape parameter and η_i is the scale parameter.

The true parameter values are shown in Table 6.1. The threshold parameters increase from 5.0 to 6.2, with adjacent populations separated by 0.3. The scale parameters vary mildly around one, and all shape parameters are greater than one. Thus, each population has an increasing hazard function. This setting is intended to represent similar degrading materials whose risk of failure increases with time.

Since the best population is defined as the population with the largest threshold parameter, the true best population in this study is P5.

For additional reliability interpretation, the true Weibull percentiles are computed from

$$t_p = \gamma + \eta[-\log(1 - p)]^{1/\beta}.$$

Table 6.1: Generating parameters for the five Weibull populations

Population	γ	η	β
P1	5.0	0.90	1.20
P2	5.3	0.95	1.35
P3	5.6	1.00	1.50
P4	5.9	1.05	1.65
P5	6.2	1.10	1.80

The true values of $t_{0.10}$, $t_{0.50}$, and $t_{0.90}$ are shown in Table 6.2. The ordering of these percentiles is also consistent with P5 being the strongest population across the lifetime distribution.

Table 6.2: Weibull percentiles computed from the generating parameters

Population	γ	η	β	$t_{0.10}$	$t_{0.50}$	$t_{0.90}$
P1	5.0	0.90	1.20	5.1380	5.6631	6.8034
P2	5.3	0.95	1.35	5.4794	6.0241	7.0621
P3	5.6	1.00	1.50	5.8231	6.3832	7.3437
P4	5.9	1.05	1.65	6.1685	6.7409	7.6407
P5	6.2	1.10	1.80	6.5151	7.0974	7.9483

6.6.2 Posterior Model for the Generated Failure-Time Data

For this fully simulated study, all observations are exact failure times. For population i , let

$$x_{i1}, \dots, x_{in}$$

denote the observed lifetimes, where $n = 150$. Conditional on the three-parameter Weibull parameters

$$(\gamma_i, \eta_i, \beta_i),$$

the density of an exact failure time is

$$f(x_{ij} | \gamma_i, \eta_i, \beta_i) = \frac{\beta_i}{\eta_i} \left(\frac{x_{ij} - \gamma_i}{\eta_i} \right)^{\beta_i - 1} \exp \left[- \left(\frac{x_{ij} - \gamma_i}{\eta_i} \right)^{\beta_i} \right], \quad x_{ij} > \gamma_i.$$

Therefore, the likelihood for population i is

$$L_i(\gamma_i, \eta_i, \beta_i | \mathbf{x}_i) = \prod_{j=1}^{n_i} f(x_{ij} | \gamma_i, \eta_i, \beta_i).$$

As described in the posterior computation section 6.5, the model is implemented on the transformed scale

$$\theta_i = (\theta_{i1}, \theta_{i2}, \theta_{i3})^\top = (\log \beta_i, \log \eta_i, \log \delta_i)^\top,$$

where

$$\gamma_i = x_{i(1)} - \delta_i - \varepsilon, \quad \delta_i > 0,$$

and $x_{i(1)}$ is the smallest observed lifetime in population i . This reparameterization ensures that the threshold parameter remains below the smallest observed lifetime at every MCMC iteration.

Using the priors specified earlier in Section 6.4, the posterior density on the transformed scale is proportional to the likelihood multiplied by the prior density:

$$\pi(\theta_i | \mathbf{x}_i) \propto L_i(\gamma_i, \eta_i, \beta_i | \mathbf{x}_i) \pi(\beta_i) \pi(\eta_i) \pi(\theta_{i3}) \left| \frac{\partial(\beta_i, \eta_i)}{\partial(\theta_{i1}, \theta_{i2})} \right|.$$

Since

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

the Jacobian contribution for β_i and η_i is

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

Thus, the posterior density used for MCMC sampling can be written as

$$\pi(\theta_i | \mathbf{x}_i) \propto L_i(\gamma_i, \eta_i, \beta_i | \mathbf{x}_i) \pi(\beta_i) \pi(\eta_i) \pi(\theta_{i3}) \exp(\theta_{i1}) \exp(\theta_{i2}).$$

Equivalently, the log-posterior is

$$\ell_i(\theta_i) = \log L_i(\gamma_i, \eta_i, \beta_i | \mathbf{x}_i) + \log \pi(\beta_i) + \log \pi(\eta_i) + \log \pi(\theta_{i3}) + \theta_{i1} + \theta_{i2} + C,$$

where C is a constant that does not depend on the model parameters. The Metropolis–Hastings samplers target this posterior distribution separately for each population.

6.6.3 Posterior Estimation of Weibull Parameters

The Bayesian model was fit separately to each population using both the random-walk Metropolis sampler and the independence Metropolis–Hastings sampler. Table 6.3 gives the posterior means of γ , η , and β under both samplers. The posterior means from the two samplers are nearly identical for all parameters and populations, indicating that the posterior estimates are not sensitive to the choice of sampler.

The posterior threshold estimates are close to the true generating values and preserve the true ordering of the populations. In particular, the posterior mean of γ increases from P1 to P5, with P5 having the largest estimated threshold parameter under both samplers. This agrees with the data-generating configuration, where P5 is the true best population according to the threshold criterion.

The posterior scale estimates are also close to the generating values and remain near one for all five populations. The posterior shape estimates are all greater than one,

Table 6.3: Posterior means of γ , η , and β for the fully simulated study

Parameter	Population	Random-Walk Mean	Independence Mean
γ	P1	4.9999	5.0000
γ	P2	5.3151	5.3151
γ	P3	5.5839	5.5840
γ	P4	5.9324	5.9324
γ	P5	6.1875	6.1874
η	P1	0.9380	0.9383
η	P2	0.9514	0.9521
η	P3	1.0003	1.0001
η	P4	1.0100	1.0097
η	P5	1.0861	1.0866
β	P1	1.1592	1.1588
β	P2	1.2814	1.2813
β	P3	1.5470	1.5465
β	P4	1.5265	1.5263
β	P5	1.6939	1.6935

which supports the intended increasing-hazard setting. Some differences between the posterior means and the true generating values are expected because the posterior estimates are based on one finite simulated sample from each population. Although the data are generated from known parameter values, the realized samples do not reproduce the generating distributions exactly. The Weibull shape parameter β controls the shape of the distribution and the behavior of the hazard function, so its estimate can be affected by the observed spread and tail behavior of the sample [39]. Importantly, these differences do not affect the ordering of the threshold parameters or the identification of P5 as the best population under the threshold-based criterion.

Table 6.4 gives the posterior medians of γ , η , and β . The posterior medians from the random-walk Metropolis sampler and the independence Metropolis–Hastings sampler are nearly identical, providing additional evidence that both samplers produce consistent posterior summaries.

The posterior medians are close to the corresponding posterior means in Table 6.3. For η and β , this suggests that the posterior distributions are approximately centered around similar mean and median values. For the threshold parameter γ , the posterior medians are slightly larger than the posterior means for some populations. This is consistent with mild skewness in the posterior distribution of γ , which is expected because the threshold parameter is constrained to lie below the smallest observed lifetime.

The median estimates also preserve the ordering of the threshold parameters across the five populations. In particular, P5 has the largest posterior median of γ under both samplers, agreeing with the posterior mean results and with the true data-generating configuration.

Table 6.5 compares the posterior estimates with the true generating parameter values. The posterior estimates from the random-walk Metropolis sampler and the independence Metropolis–Hastings sampler are nearly identical, so the observed estimation errors are not attributable to the choice of sampler.

For the threshold parameter γ , the estimation errors are small relative to the separation of 0.3 between adjacent generating threshold values. As a result, the posterior estimates preserve the true ordering of the populations, with P5 having the largest threshold estimate under both samplers. The larger differences occur mainly for some of the scale and shape estimates. These differences are expected in a finite simulated sample because η and β are estimated jointly with the threshold parameter and are influenced by

Table 6.4: Posterior medians of γ , η , and β for the fully simulated study.

Parameter	Population	Random-Walk Median	Independence Median
γ	P1	5.0022	5.0021
γ	P2	5.3187	5.3184
γ	P3	5.5890	5.5885
γ	P4	5.9372	5.9367
γ	P5	6.1933	6.1925
η	P1	0.9361	0.9365
η	P2	0.9494	0.9504
η	P3	0.9984	0.9981
η	P4	1.0083	1.0077
η	P5	1.0840	1.0845
β	P1	1.1577	1.1571
β	P2	1.2781	1.2776
β	P3	1.5423	1.5419
β	P4	1.5224	1.5219
β	P5	1.6892	1.6886

the realized spread and tail behavior of the generated lifetimes. These differences do not affect the threshold-based subset selection conclusion.

Table 6.5: Posterior estimates compared with true parameter values.

	Parameter	Pop.	True	RW Mean	RW Median	Indep Mean	Indep Median	RW Mean Error	RW Median Error	Indep Mean Error	Indep Median Error
101	γ	P1	5.00	4.9999	5.0022	5.0000	5.0021	-0.0001	0.0022	0.0000	0.0021
	γ	P2	5.30	5.3151	5.3187	5.3151	5.3184	0.0151	0.0187	0.0151	0.0184
	γ	P3	5.60	5.5839	5.5890	5.5840	5.5885	-0.0161	-0.0110	-0.0160	-0.0115
	γ	P4	5.90	5.9324	5.9372	5.9324	5.9367	0.0324	0.0372	0.0324	0.0367
	γ	P5	6.20	6.1875	6.1933	6.1874	6.1925	-0.0125	-0.0067	-0.0126	-0.0075
	η	P1	0.90	0.9380	0.9361	0.9383	0.9365	0.0380	0.0361	0.0383	0.0365
	η	P2	0.95	0.9514	0.9494	0.9521	0.9504	0.0014	-0.0006	0.0021	0.0004
	η	P3	1.00	1.0003	0.9984	1.0001	0.9981	0.0003	-0.0016	0.0001	-0.0019
	η	P4	1.05	1.0100	1.0083	1.0097	1.0077	-0.0400	-0.0417	-0.0403	-0.0423
	η	P5	1.10	1.0861	1.0840	1.0866	1.0845	-0.0139	-0.0160	-0.0134	-0.0155
	β	P1	1.20	1.1592	1.1577	1.1588	1.1571	-0.0408	-0.0423	-0.0412	-0.0429
	β	P2	1.35	1.2814	1.2781	1.2813	1.2776	-0.0686	-0.0719	-0.0687	-0.0724
	β	P3	1.50	1.5470	1.5423	1.5465	1.5419	0.0470	0.0423	0.0465	0.0419
	β	P4	1.65	1.5265	1.5224	1.5263	1.5219	-0.1235	-0.1276	-0.1237	-0.1281
	β	P5	1.80	1.6939	1.6892	1.6935	1.6886	-0.1061	-0.1108	-0.1065	-0.1114

The posterior means and 95% credible intervals for the Weibull parameters and percentiles are shown in Tables 6.6–6.7. The posterior summaries from the random-walk Metropolis sampler and the independence Metropolis–Hastings sampler are nearly identical, indicating that the two samplers lead to the same posterior conclusions.

For the threshold parameter γ , the credible intervals are well separated across the populations, especially between P5 and the lower-threshold populations. This separation explains why the posterior probability that P5 has the largest threshold parameter is essentially one. The credible intervals for η and β are wider than those for γ , reflecting greater posterior uncertainty in the scale and shape parameters. However, this uncertainty does not change the threshold-based ordering of the populations.

The percentile summaries in Table 6.7 provide an additional reliability interpretation. The posterior means of $t_{0.10}$, $t_{0.50}$, and $t_{0.90}$ generally increase from P1 to P5, which is consistent with the ordering of the generating threshold parameters. In particular, P5 has the largest posterior mean for each of the three percentiles. Thus, in this simulation study, the population selected by the threshold-based criterion also has the largest posterior reliability percentiles.

Figures 6.1–6.5 show the posterior densities of γ , η , and β for the five generated populations. Each figure corresponds to one population and compares the random-walk Metropolis sampler with the independence Metropolis–Hastings sampler. For all three parameters and all five populations, the two density curves are nearly overlapping. This agreement provides visual evidence that the two samplers are targeting the same posterior distributions and supports the numerical agreement observed in the posterior summary tables.

The posterior densities for the threshold parameter γ are more concentrated than those for η and β . The densities for γ are also slightly asymmetric, with most of the

Table 6.6: Posterior means and 95% credible intervals for Weibull parameters.

Criterion	Population	RW Mean	RW 2.5%	RW 97.5%	Indep Mean	Indep 2.5%	Indep 97.5%
γ	P1	4.9999	4.9749	5.0119	5.0000	4.9756	5.0118
γ	P2	5.3151	5.2759	5.3333	5.3151	5.2770	5.3331
γ	P3	5.5839	5.5296	5.6089	5.5840	5.5308	5.6087
γ	P4	5.9324	5.8787	5.9584	5.9324	5.8800	5.9582
γ	P5	6.1875	6.1250	6.2169	6.1874	6.1258	6.2167
η	P1	0.9380	0.8108	1.0772	0.9383	0.8095	1.0776
η	P2	0.9514	0.8291	1.0848	0.9521	0.8284	1.0852
η	P3	1.0003	0.8883	1.1245	1.0001	0.8870	1.1255
η	P4	1.0100	0.8946	1.1367	1.0097	0.8945	1.1364
η	P5	1.0861	0.9715	1.2118	1.0866	0.9713	1.2134
β	P1	1.1592	1.0215	1.3047	1.1588	1.0174	1.3076
β	P2	1.2814	1.1105	1.4713	1.2813	1.1076	1.4757
β	P3	1.5470	1.3424	1.7801	1.5465	1.3379	1.7827
β	P4	1.5265	1.3267	1.7495	1.5263	1.3229	1.7522
β	P5	1.6939	1.4707	1.9427	1.6935	1.4673	1.9450

Table 6.7: Posterior means and 95% credible intervals for Weibull percentiles.

Criterion	Population	RW Mean	RW 2.5%	RW 97.5%	Indep Mean	Indep 2.5%	Indep 97.5%
$t_{0.10}$	P1	5.1349	5.0965	5.1774	5.1349	5.0960	5.1780
$t_{0.10}$	P2	5.4798	5.4354	5.5284	5.4799	5.4348	5.5294
$t_{0.10}$	P3	5.8176	5.7637	5.8741	5.8175	5.7630	5.8751
$t_{0.10}$	P4	6.1638	6.1109	6.2197	6.1638	6.1095	6.2205
$t_{0.10}$	P5	6.4752	6.4148	6.5377	6.4752	6.4135	6.5391
$t_{0.50}$	P1	5.6832	5.5814	5.7926	5.6834	5.5803	5.7926
$t_{0.50}$	P2	6.0293	5.9302	6.1341	6.0298	5.9297	6.1350
$t_{0.50}$	P3	6.3727	6.2808	6.4680	6.3725	6.2792	6.4697
$t_{0.50}$	P4	6.7263	6.6325	6.8239	6.7261	6.6313	6.8248
$t_{0.50}$	P5	7.0618	6.9676	7.1582	7.0621	6.9666	7.1599
$t_{0.90}$	P1	6.9312	6.6840	7.2308	6.9327	6.6832	7.2322
$t_{0.90}$	P2	7.1450	6.9274	7.4100	7.1467	6.9276	7.4137
$t_{0.90}$	P3	7.3033	7.1331	7.5064	7.3034	7.1320	7.5078
$t_{0.90}$	P4	7.6810	7.5055	7.8890	7.6807	7.5059	7.8904
$t_{0.90}$	P5	7.9684	7.8078	8.1583	7.9696	7.8081	8.1605

posterior mass close to the upper feasible region and a longer tail to the left. This behavior is expected because the threshold parameter is constrained to lie below the smallest observed lifetime. The concentration of the γ densities, together with their separation across populations, supports the stability of the threshold-based ordering.

The posterior densities for η and β are wider and more bell-shaped, indicating greater posterior uncertainty in the scale and shape parameters. This is consistent with the credible interval results, where η and β have wider posterior intervals than γ . Although there is more uncertainty in these parameters, the posterior distributions remain centered near plausible values from the generating configuration, and the estimated shape parameters remain greater than one. Thus, the fitted populations retain the intended increasing-hazard behavior.

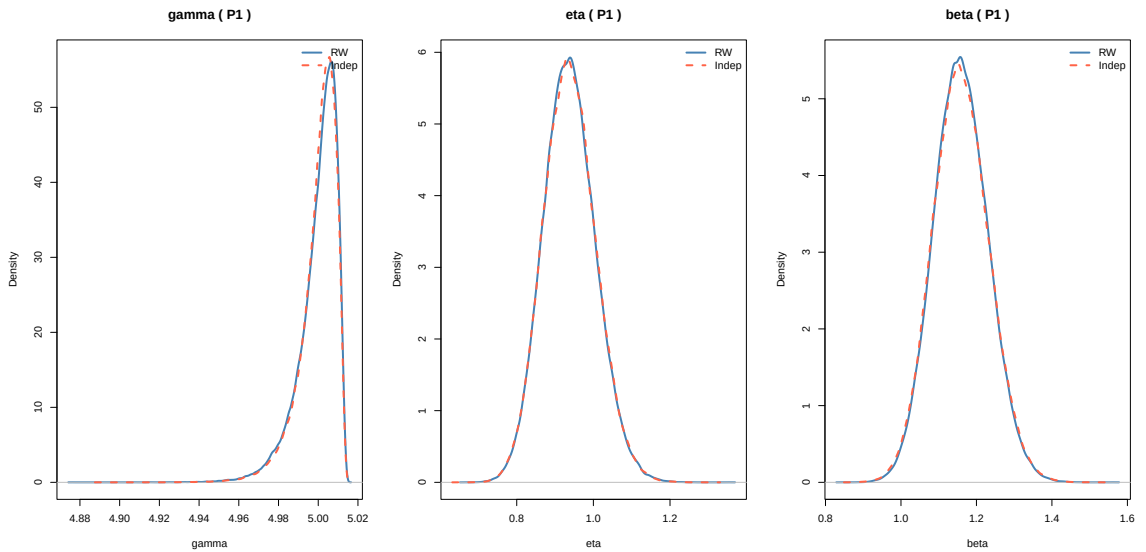


Figure 6.1: Posterior densities of γ , η , and β for Population P1.

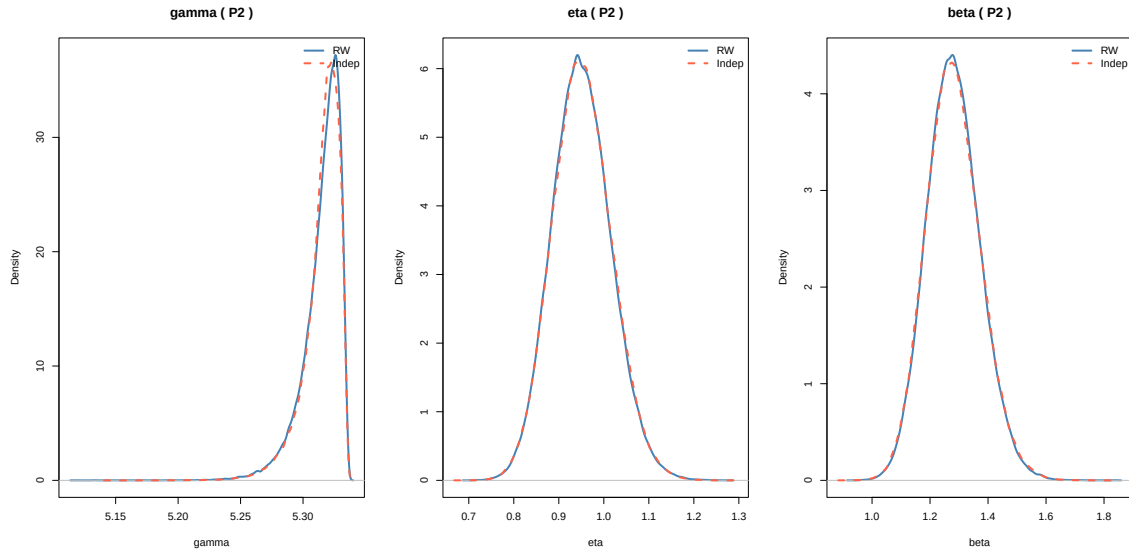


Figure 6.2: Posterior densities of γ , η , and β for Population P2.

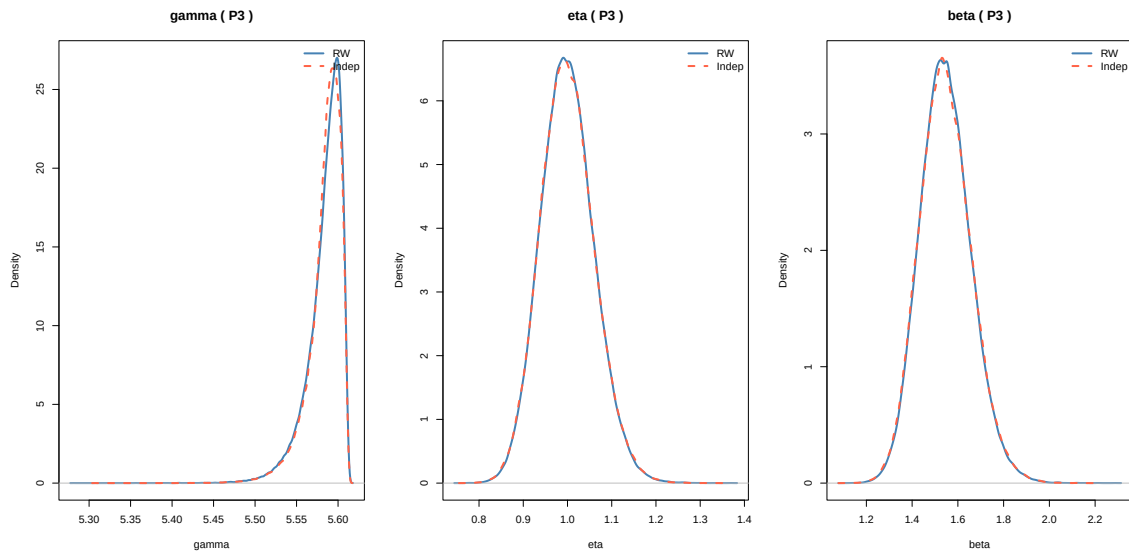


Figure 6.3: Posterior densities of γ , η , and β for Population P3.

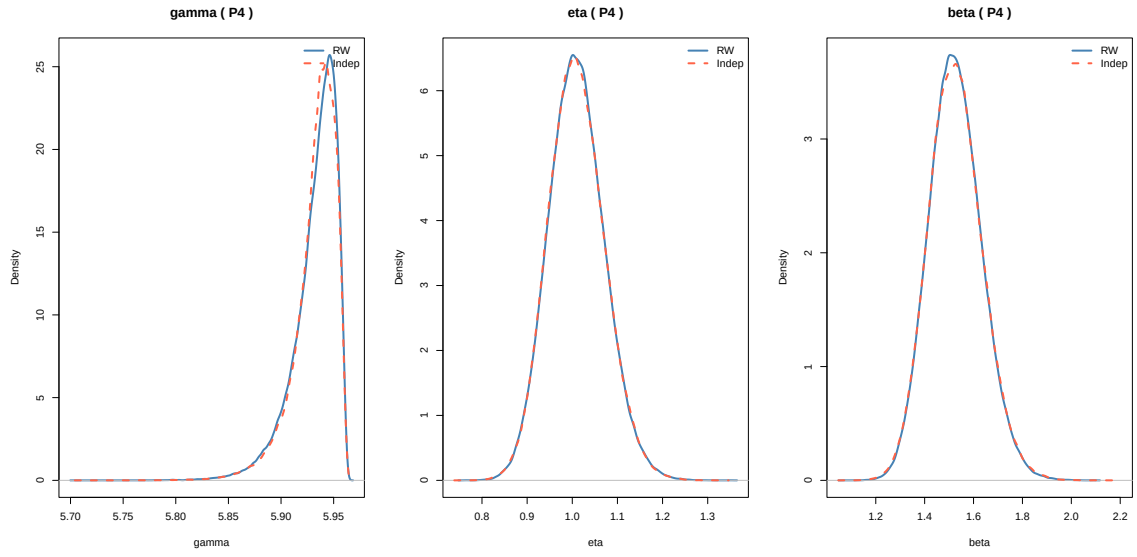


Figure 6.4: Posterior densities of γ , η , and β for Population P4.

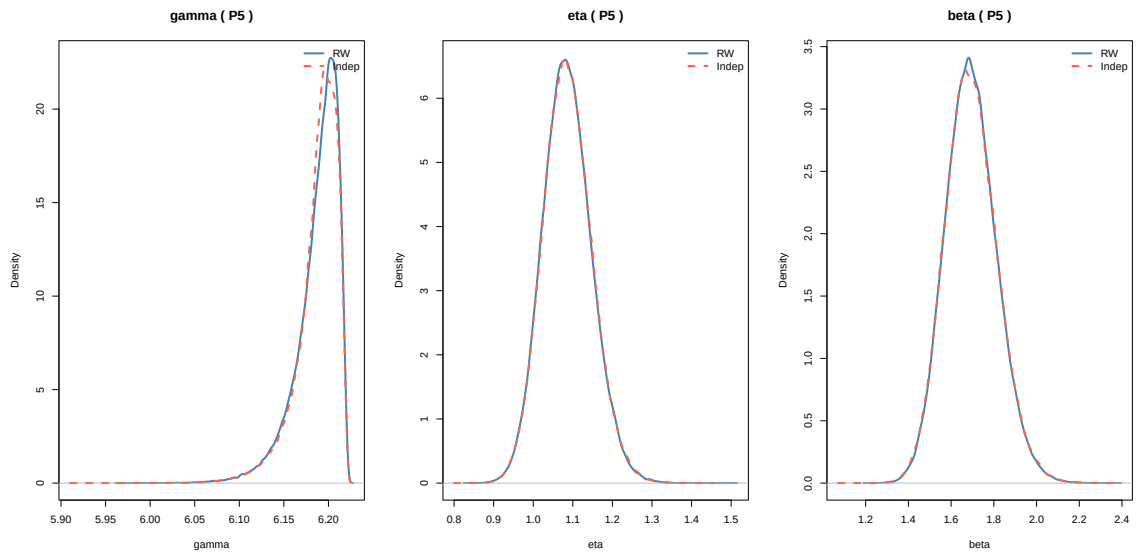


Figure 6.5: Posterior densities of γ , η , and β for Population P5.

6.6.4 Posterior Selection Probabilities

Table 6.8 gives the posterior probabilities of having the largest value of γ , η , and β . The formal Bayesian subset selection rule is based on the threshold parameter γ . The posterior probabilities for η and β are reported only as supplementary summaries, since they help describe how the fitted populations compare in terms of scale and shape.

For the threshold parameter, P5 has posterior probability 1.0000 of having the largest γ under both samplers. Therefore, for any of the posterior probability levels considered in the earlier chapters, such as

$$P^* = 0.75, 0.90, 0.95, 0.975, \text{ or } 0.99,$$

the Bayesian selected subset is

$$\mathcal{S}_B = \{\text{P5}\}.$$

The estimated posterior probability of correct selection for this subset is 1.0000.

The posterior probabilities for η and β are less concentrated than those for γ . For η , P5 has posterior probability approximately 0.68 of having the largest scale parameter. For β , P5 has posterior probability approximately 0.73 of having the largest shape parameter. The remaining posterior probability is mostly assigned to P3 and P4, indicating some posterior uncertainty in the ordering of the scale and shape parameters. This is consistent with the posterior density and credible interval results, where η and β show more overlap across populations than γ .

These results emphasize that the threshold parameter is much more sharply ordered than the scale and shape parameters in this simulated study. Since the best population is formally defined by the largest threshold parameter, the uncertainty in η and

β does not alter the subset selection conclusion. Both samplers identify P5 as the unique selected population.

Table 6.8: Posterior probabilities of having the largest parameter value.

Parameter	Population	Random-Walk Probability	Independence Probability
γ	P1	0.0000	0.0000
γ	P2	0.0000	0.0000
γ	P3	0.0000	0.0000
γ	P4	0.0000	0.0000
γ	P5	1.0000	1.0000
η	P1	0.0284	0.0281
η	P2	0.0365	0.0368
η	P3	0.1097	0.1099
η	P4	0.1436	0.1439
η	P5	0.6818	0.6813
β	P1	0.0000	0.0000
β	P2	0.0007	0.0007
β	P3	0.1534	0.1564
β	P4	0.1127	0.1159
β	P5	0.7332	0.7270

6.6.5 Posterior Weibull Percentiles

Table 6.9 gives posterior means of the Weibull percentiles $t_{0.10}$, $t_{0.50}$, and $t_{0.90}$. These percentiles provide an additional reliability interpretation because they describe early, median, and upper-tail failure times. The 10th percentile represents an early-failure

point, the 50th percentile represents the median lifetime, and the 90th percentile represents a later point in the lifetime distribution.

The posterior mean percentiles increase from P1 to P5 for all three percentile levels under both samplers. Thus, the population selected by the threshold parameter also has the largest posterior mean for the early, median, and upper-tail Weibull percentiles. This indicates that the threshold-based selection conclusion is consistent with the broader reliability behavior of the fitted Weibull distributions.

The agreement between the random-walk and independence sampler estimates is again very close for all percentiles. This provides additional evidence that the posterior summaries are stable with respect to the choice of sampler. The separation between populations is visible not only for the early-failure percentile $t_{0.10}$, but also for the median and upper-tail percentiles. Therefore, P5 is not only estimated to have the largest lower endpoint, but also the largest posterior mean lifetime percentiles across the distribution.

Table 6.10 gives the posterior probabilities of having the largest Weibull percentile. These probabilities provide a percentile-based comparison of the five populations and supplement the formal threshold-based selection rule.

For the 10th and 50th percentiles, P5 has posterior probability 1.0000 of having the largest percentile under both samplers. Thus, P5 is clearly identified as having the largest early-failure and median lifetime percentiles. For the 90th percentile, P5 still has posterior probability approximately 0.983 of being largest, while P4 receives posterior probability approximately 0.017. This indicates that the evidence for P5 remains very strong even when the comparison is based on the upper tail of the lifetime distribution. The small posterior probability assigned to P4 for $t_{0.90}$ reflects slightly greater uncertainty in the upper tail. This is consistent with the wider posterior intervals for the scale and shape parameters, since upper-tail percentiles depend on all three Weibull parameters.

Table 6.9: Posterior means of Weibull percentiles.

Percentile	Population	Random-Walk Mean	Independence Mean
10%	P1	5.1349	5.1349
10%	P2	5.4798	5.4799
10%	P3	5.8176	5.8175
10%	P4	6.1638	6.1638
10%	P5	6.4752	6.4752
50%	P1	5.6832	5.6834
50%	P2	6.0293	6.0298
50%	P3	6.3727	6.3725
50%	P4	6.7263	6.7261
50%	P5	7.0618	7.0621
90%	P1	6.9312	6.9327
90%	P2	7.1450	7.1467
90%	P3	7.3033	7.3034
90%	P4	7.6810	7.6807
90%	P5	7.9684	7.9696

Nevertheless, the percentile-based results agree with the threshold-based selection conclusion: P5 is the dominant population under both samplers and across all three reliability percentiles.

The threshold-based and percentile-based selection rules do not necessarily have to identify the same population at each posterior draw. To examine this relationship, the joint agreement probability was computed for each percentile. For a retained posterior draw m , agreement occurs when the population with the largest threshold parameter is also the population with the largest Weibull percentile:

$$\arg \max_{1 \leq i \leq 5} \gamma_i^{(m)} = \arg \max_{1 \leq i \leq 5} t_{p,i}^{(m)}.$$

Table 6.10: Posterior probabilities of having the largest Weibull percentile.

Percentile	Population	Random-Walk Probability	Independence Probability
10%	P1	0.0000	0.0000
10%	P2	0.0000	0.0000
10%	P3	0.0000	0.0000
10%	P4	0.0000	0.0000
10%	P5	1.0000	1.0000
50%	P1	0.0000	0.0000
50%	P2	0.0000	0.0000
50%	P3	0.0000	0.0000
50%	P4	0.0000	0.0000
50%	P5	1.0000	1.0000
90%	P1	0.0000	0.0000
90%	P2	0.0000	0.0000
90%	P3	0.0000	0.0000
90%	P4	0.0166	0.0165
90%	P5	0.9834	0.9835

The joint agreement probability is estimated by the proportion of retained posterior draws for which this equality holds. Table 6.11 gives the resulting agreement probabilities for $t_{0.10}$, $t_{0.50}$, and $t_{0.90}$.

Table 6.11: Joint agreement between threshold-based and percentile-based selection.

Percentile	RW Agreement Probability	Independence Agreement Probability
10%	1.0000	1.0000
50%	1.0000	1.0000
90%	0.9834	0.9835

The agreement probability is 1.0000 for the 10th and 50th percentiles under both samplers. This means that, for every retained posterior draw, the population with the largest threshold parameter is also the population with the largest early-failure and median lifetime percentiles. For the 90th percentile, the agreement probability is approximately 0.983. The slight decrease reflects the small posterior probability that P4 has the largest upper-tail percentile, as shown in Table 6.10.

Overall, these results show that threshold-based selection is highly consistent with percentile-based reliability comparisons in this fully generated study. Although the formal Bayesian selected subset is based on the threshold parameter γ , the same population, P5, is also strongly supported when the populations are compared using Weibull lifetime percentiles.

6.6.6 MCMC Diagnostics

Representative trace plots for the fully simulated study are shown in Figures 6.6–6.8. Population P1 represents the lower end of the threshold ordering, while P5 is the true best population and the population selected by the Bayesian subset-selection rule. The full set of trace plots for all five populations under both samplers is provided in Appendix C, Section C.3.

The trace plots are shown for the transformed parameters $\log(\beta)$, $\log(\eta)$, and $\log(\delta)$, since these are the parameters sampled directly by the MCMC algorithms. The threshold parameter γ is then recovered from δ through

$$\gamma = x_{(1)} - \delta - \varepsilon.$$

The displayed trace plots show stable behavior across the four chains, with no visible long-term drift. The chains fluctuate around common posterior regions for both

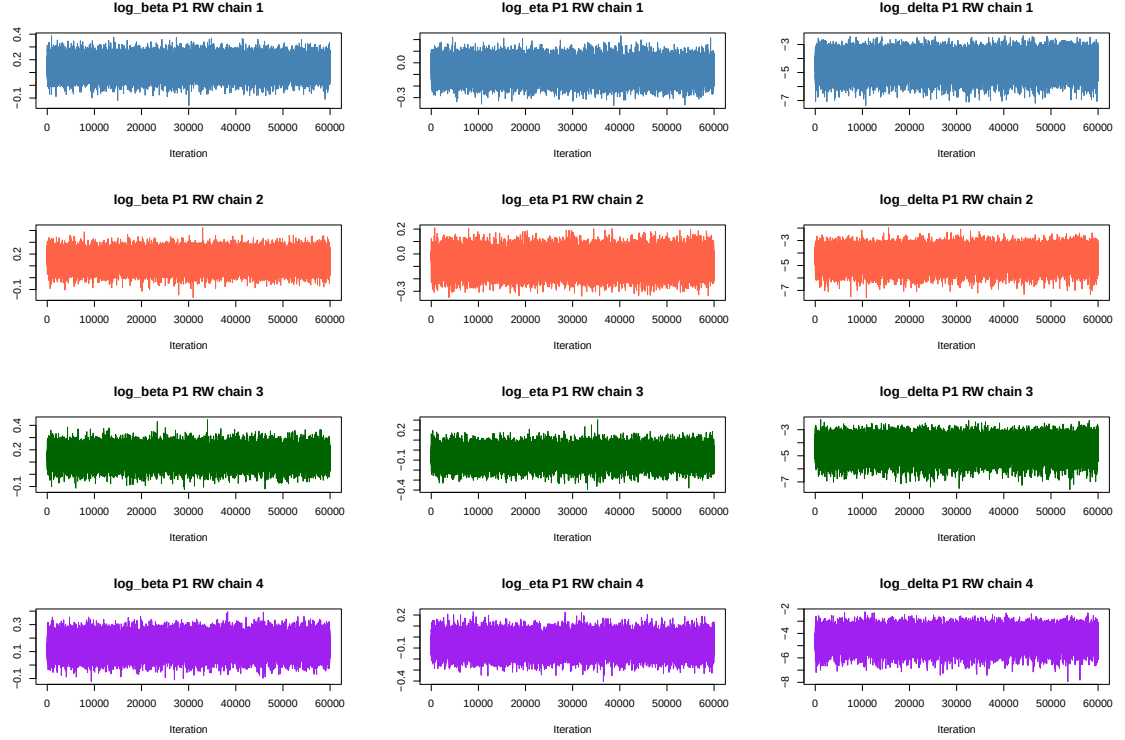


Figure 6.6: Random-walk trace plots for Population P1

samplers, supporting the convergence assessment from the Gelman–Rubin diagnostics and effective sample sizes reported below.

The trace plots for $\log(\delta)$ show more variability than those for $\log(\beta)$ and $\log(\eta)$, because δ determines the threshold parameter through the lower-endpoint constraint. Thus, small changes in δ directly affect the recovered value of γ . Despite this additional variability, the four chains overlap across the posterior region and remain stable for both the lower-threshold population P1 and the selected population P5.

The visual diagnostics also support the numerical agreement between the random-walk Metropolis sampler and the independent Metropolis–Hastings sampler. Both samplers produce chains that mix around similar posterior regions for the transformed

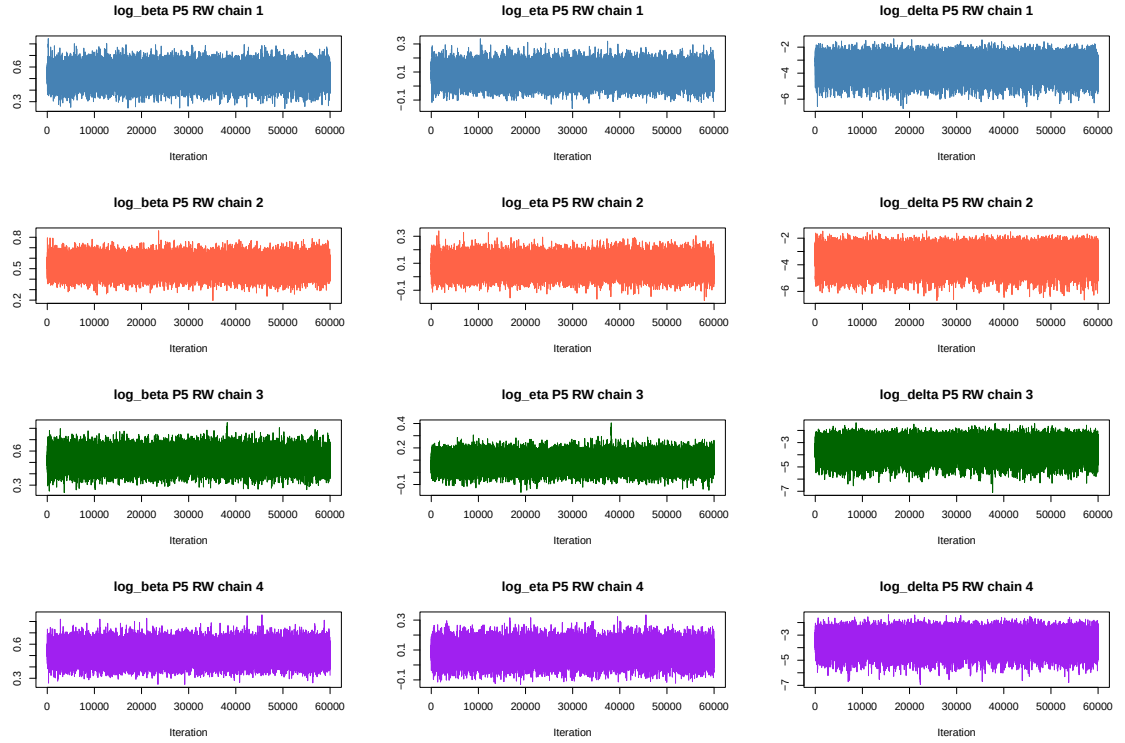


Figure 6.7: Random-walk trace plots for Population P5

shape, scale, and threshold-gap parameters. Therefore, the representative trace plots do not indicate evidence of non-convergence or sampler-dependent posterior behavior.

Figure 6.9 shows representative autocorrelation functions for γ , η , and β under both samplers for Population P5. Population P5 is used as the representative case because it is the true best population and the population selected by the Bayesian subset-selection rule. The full set of autocorrelation plots for all five populations is provided in Appendix C, Section C.4.

The autocorrelation functions decrease with increasing lag under both samplers, indicating that dependence between retained draws weakens as the lag increases. The independent Metropolis–Hastings sampler generally shows smaller autocorrelation at

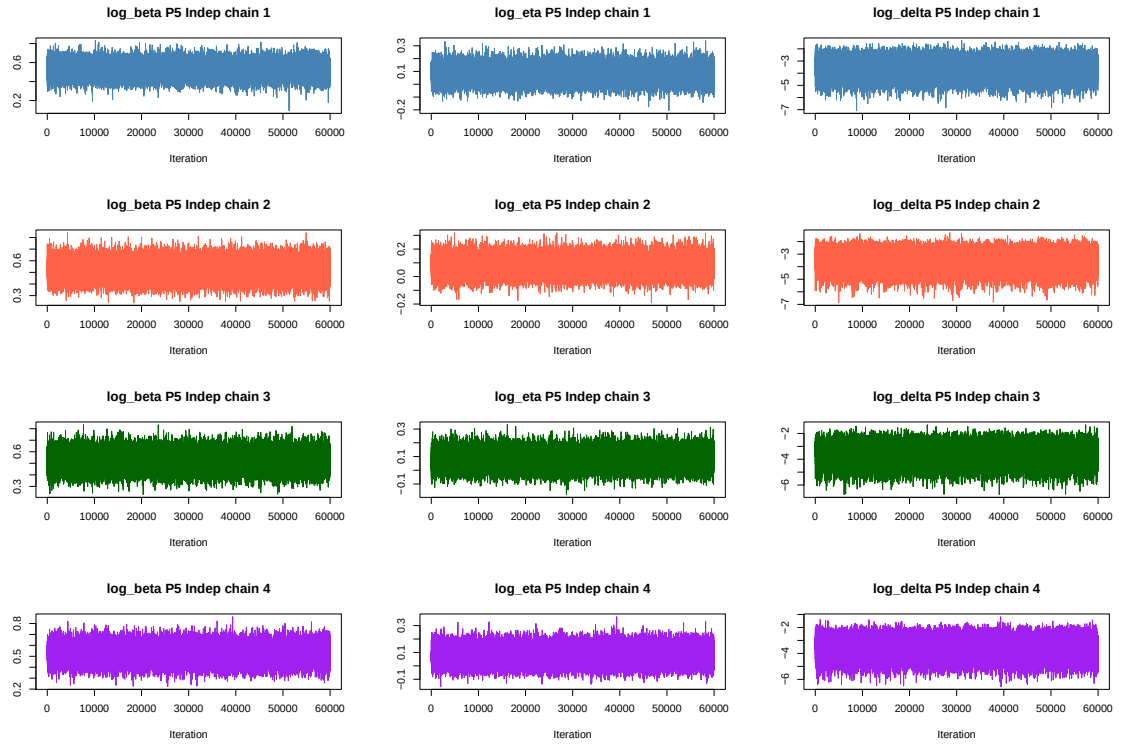


Figure 6.8: Independent-sampler trace plots for Population P5

short lags than the random-walk Metropolis sampler. This is consistent with the purpose of the independent proposal, which uses a proposal distribution designed to approximate the posterior distribution more globally.

The reduction in autocorrelation is relevant for effective posterior sampling, since lower serial dependence corresponds to larger effective sample sizes. Although some autocorrelation is still visible at small lags, especially for parameters related to the threshold component, the overall autocorrelation pattern supports adequate mixing. Together with the trace plots, Gelman–Rubin diagnostics, and effective sample sizes, the Autocorrelation Function (ACF) results support the stability of the MCMC output for the fully simulated study.

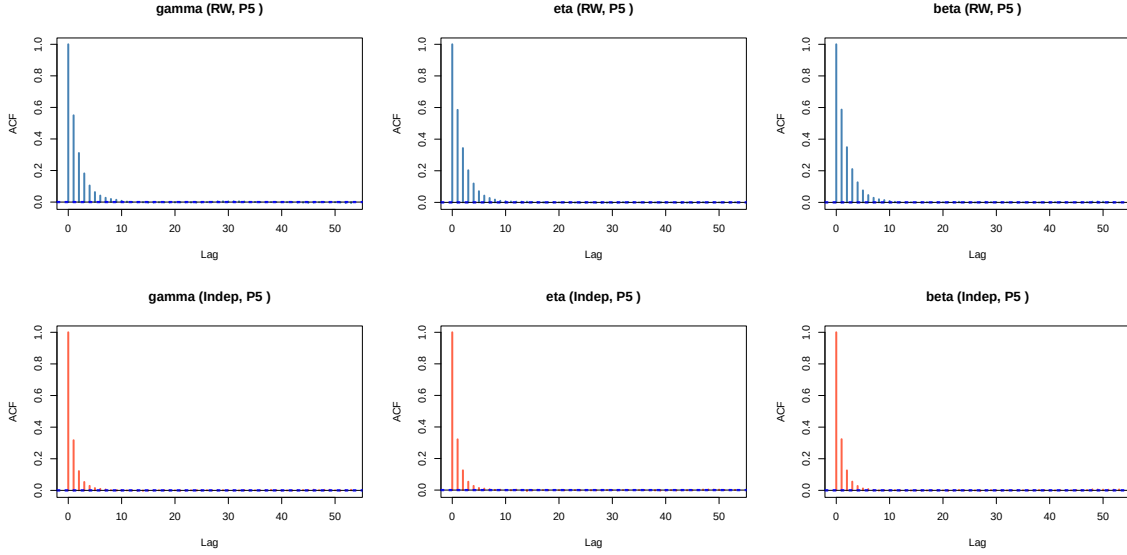


Figure 6.9: Autocorrelation functions for γ , η , and β for Population P5

Table 6.12 summarizes the acceptance rates across the four chains for each population. In a Metropolis–Hastings algorithm, a proposed value is either accepted or rejected at each iteration. Let $A_m = 1$ if the proposal at iteration m is accepted and let $A_m = 0$ otherwise. If M proposal iterations are used to compute the diagnostic, then the empirical acceptance rate is

$$\hat{r}_{\text{acc}} = \frac{1}{M} \sum_{m=1}^M A_m.$$

Thus, the acceptance rate is the proportion of proposed moves that are accepted by the sampler.

The random-walk acceptance rates are approximately 0.37 for all five populations, while the independence sampler acceptance rates are approximately 0.25. The minimum and maximum values across the four chains are close to the corresponding chain averages, indicating that the tuning behavior is stable across chains.

Table 6.12: Acceptance rates across chains.

Population	Random-Walk Sampler			Independence Sampler		
	Mean	Min	Max	Mean	Min	Max
P1	0.370	0.362	0.378	0.261	0.260	0.261
P2	0.372	0.367	0.376	0.251	0.249	0.252
P3	0.366	0.359	0.370	0.247	0.246	0.249
P4	0.370	0.361	0.377	0.250	0.249	0.250
P5	0.369	0.362	0.378	0.247	0.246	0.248

Acceptance rates should not be interpreted using a single universal cutoff, since desirable values depend on the target distribution, the dimension of the parameter space, and the proposal mechanism [40, 41]. In this study, the acceptance rates are reasonable for the three-dimensional posterior simulation used for each population. Although the independence sampler has lower acceptance rates than the random-walk sampler, it also produces larger effective sample sizes and lower autocorrelation. Therefore, the lower acceptance rate does not indicate poorer performance; rather, the overall diagnostics suggest that both samplers perform adequately, with the independence sampler being more efficient in terms of effective posterior draws.

Table 6.13 gives the Gelman–Rubin convergence diagnostics for the transformed Weibull parameters. The Gelman–Rubin diagnostic, also called the potential scale reduction factor, compares variation between multiple chains with variation within each chain [37]. Suppose that J parallel chains are run, each with M retained draws for a scalar parameter ψ . Let $\bar{\psi}_{\cdot j}$ denote the sample mean of chain j , let $\bar{\psi}_{\cdot\cdot}$ denote the overall mean across chains, and let s_j^2 denote the sample variance within chain j . The within-chain

variance is

$$W = \frac{1}{J} \sum_{j=1}^J s_j^2,$$

and the between-chain variance is

$$B = \frac{M}{J-1} \sum_{j=1}^J (\bar{\psi}_{\cdot j} - \bar{\psi}_{\cdot\cdot})^2.$$

An estimate of the marginal posterior variance is

$$\hat{V} = \frac{M-1}{M} W + \frac{1}{M} B.$$

The potential scale reduction factor is then

$$\hat{R} = \sqrt{\frac{\hat{V}}{W}}.$$

Values of $\hat{R}$ close to one indicate that the between-chain and within-chain variability are similar, suggesting that the chains have mixed into the same posterior region.

The results in Table 6.13 show that all $\hat{R}$ values are essentially equal to one for both samplers and for all transformed parameters. This supports convergence of the four chains for both the random-walk Metropolis sampler and the independence Metropolis–Hastings sampler.

Table 6.14 gives effective sample sizes for γ , η , and β . The effective sample size measures the amount of independent posterior information contained in an autocorrelated MCMC sample [42]. If M retained posterior draws are available for a scalar parameter and ρ_ℓ denotes the autocorrelation at lag ℓ , then the effective sample size may be written as

$$\text{ESS} = \frac{M}{1 + 2 \sum_{\ell=1}^{\infty} \rho_\ell}.$$

When the retained draws are independent, the autocorrelations are approximately zero and $\text{ESS} \approx M$. Positive autocorrelation reduces the effective sample size below the nominal

Table 6.13: Gelman–Rubin $\hat{R}$ diagnostics for the transformed Weibull parameters.

Population	Parameter	Random-Walk $\hat{R}$	Independence $\hat{R}$
P1	$\log(\beta)$	1.0001	1.0001
	$\log(\eta)$	1.0000	1.0000
	$\log(\delta)$	1.0001	1.0000
P2	$\log(\beta)$	1.0000	1.0000
	$\log(\eta)$	1.0000	1.0000
	$\log(\delta)$	1.0001	1.0000
P3	$\log(\beta)$	1.0000	1.0000
	$\log(\eta)$	1.0000	1.0000
	$\log(\delta)$	1.0000	1.0000
P4	$\log(\beta)$	1.0001	1.0000
	$\log(\eta)$	1.0000	1.0000
	$\log(\delta)$	1.0001	1.0000
P5	$\log(\beta)$	1.0000	1.0000
	$\log(\eta)$	1.0001	1.0000
	$\log(\delta)$	1.0001	1.0000

number of retained draws. In practice, the infinite sum is estimated from the simulated Markov Chain, usually by truncating the autocorrelation sequence at an appropriate lag. Larger ESS values indicate lower serial dependence and more efficient posterior sampling.

The effective sample sizes are large for both samplers, indicating that the posterior means, medians, credible intervals, and selection probabilities are based on stable simulation output. The independence Metropolis–Hastings sampler has larger effective sample sizes than the random-walk Metropolis sampler for every population and parameter. This is consistent with the autocorrelation plots, where the independence sampler generally shows lower serial dependence. Nevertheless, the random-walk sampler also produces sufficiently large effective sample sizes for stable posterior inference.

Table 6.14: Effective sample sizes for Weibull parameters.

Population	Random-Walk Sampler			Independence Sampler		
	γ	η	β	γ	η	β
P1	71,098.1	62,475.0	64,260.4	120,002.8	116,065.9	115,208.3
P2	67,867.1	63,227.2	63,275.2	118,384.5	113,620.2	113,800.3
P3	62,770.7	59,729.4	60,793.9	115,065.8	111,710.3	113,103.3
P4	67,268.2	61,653.2	62,210.2	116,446.8	112,998.0	114,141.8
P5	65,866.5	62,806.8	61,060.1	114,816.9	114,966.8	113,732.1

Table 6.15 gives effective sample sizes for the Weibull percentiles. These values are also large, indicating that the percentile summaries and percentile-based selection probabilities are based on stable posterior simulation output.

The percentile effective sample sizes follow the same pattern as the parameter effective sample sizes. The independence sampler produces larger ESS values than the random-walk sampler, reflecting more efficient sampling. However, the random-walk ESS values are still large across all populations and percentile levels. Therefore, the posterior percentile means and the posterior probabilities of having the largest percentile are not driven by Monte Carlo instability.

6.6.7 Prior Sensitivity Analysis

Prior sensitivity analysis was used to examine whether the posterior inference and subset selection conclusion were sensitive to the prior specification for the threshold-distance parameter. Such sensitivity checks are commonly recommended in applied Bayesian analyses, especially when posterior inference may be affected by weakly informative or tuning-related prior choices [33, 30]. The prior sensitivity analysis examines the effect of changing the prior standard deviation of $\theta_3 = \log(\delta)$. The values

Table 6.15: Effective sample sizes for Weibull percentiles.

Percentile	Population	ESS (RW)	ESS (Independence)
10%	P1	65,631.4	117,370.6
10%	P2	67,996.5	113,963.6
10%	P3	67,782.4	114,774.3
10%	P4	67,534.2	114,646.8
10%	P5	64,801.0	113,454.3
50%	P1	62,727.7	117,337.4
50%	P2	64,929.0	112,111.1
50%	P3	63,333.7	114,551.3
50%	P4	63,967.9	114,437.4
50%	P5	64,651.6	113,931.7
90%	P1	62,974.2	118,215.8
90%	P2	62,050.1	113,822.6
90%	P3	62,059.9	112,880.8
90%	P4	61,129.0	114,283.1
90%	P5	63,382.4	111,719.5

considered are 0.5, 1.0, and 1.5. Table 6.16 gives posterior summaries for γ and δ under these three choices. Since the threshold parameter is written as

$$\gamma = x_{(1)} - \delta - \varepsilon,$$

changes in the posterior distribution of δ directly affect the posterior distribution of γ . In particular, smaller posterior values of δ correspond to larger posterior values of γ , placing the estimated threshold closer to the smallest observed failure time.

The posterior summaries show some sensitivity to the prior standard deviation of $\log(\delta)$. As σ_{θ_3} increases from 0.5 to 1.5, the posterior mean of δ generally decreases, and the posterior mean of γ correspondingly increases. This behavior is expected because γ

and δ are linked through the threshold reparameterization. A smaller value of δ moves the threshold closer to the smallest observed lifetime.

Although the posterior means shift slightly across the three prior specifications, the ordering of the threshold parameters remains unchanged. In all cases, P5 has the largest posterior mean of γ , followed by P4, P3, P2, and P1. Therefore, the prior sensitivity analysis shows that the numerical posterior estimates of γ are somewhat affected by the prior spread assigned to $\log(\delta)$, but the threshold-based subset selection conclusion is robust. The selected population remains P5 under all three prior specifications.

Figure 6.10 displays the same prior sensitivity analysis graphically. Each panel corresponds to one population. For each prior standard deviation setting, the solid line gives the posterior mean of γ , calculated from the retained posterior draws as

$$\hat{E}(\gamma_i | \mathbf{x}) = \frac{1}{M} \sum_{m=1}^M \gamma_i^{(m)}.$$

The dashed lines give the corresponding equal-tail 95% credible interval, obtained from the 2.5th and 97.5th percentiles of the posterior draws:

$$\left[Q_{0.025}(\gamma_i^{(1)}, \dots, \gamma_i^{(M)}), Q_{0.975}(\gamma_i^{(1)}, \dots, \gamma_i^{(M)}) \right].$$

The posterior mean lies inside the credible interval for each population and prior setting, as expected for these posterior distributions. The width of the interval reflects posterior uncertainty in the threshold estimate under each prior specification. The posterior means shift slightly as the prior standard deviation of $\log(\delta)$ changes, which is consistent with the numerical summaries in Table 6.16. Since

$$\gamma = x_{(1)} - \delta - \varepsilon,$$

changes in the posterior distribution of δ directly affect the posterior distribution of γ .

Table 6.16: Prior sensitivity analysis for the prior standard deviation of $\log(\delta)$.

σ_{θ_3}	Population	γ			δ		
		Mean	2.5%	97.5%	Mean	2.5%	97.5%
0.5	P1	4.9870	4.9605	5.0036	0.0280	0.0113	0.0545
0.5	P2	5.3020	5.2660	5.3234	0.0356	0.0142	0.0716
0.5	P3	5.5736	5.5317	5.5984	0.0410	0.0163	0.0829
0.5	P4	5.9227	5.8809	5.9481	0.0420	0.0166	0.0838
0.5	P5	6.1788	6.1306	6.2062	0.0448	0.0174	0.0930
1.0	P1	4.9999	4.9754	5.0119	0.0150	0.0030	0.0396
1.0	P2	5.3149	5.2748	5.3334	0.0227	0.0041	0.0628
1.0	P3	5.5839	5.5290	5.6087	0.0308	0.0059	0.0856
1.0	P4	5.9320	5.8795	5.9586	0.0327	0.0061	0.0852
1.0	P5	6.1881	6.1253	6.2168	0.0355	0.0068	0.0983
1.5	P1	5.0051	4.9830	5.0139	0.0099	0.0010	0.0320
1.5	P2	5.3213	5.2823	5.3361	0.0162	0.0014	0.0553
1.5	P3	5.5902	5.5350	5.6123	0.0245	0.0023	0.0796
1.5	P4	5.9382	5.8810	5.9619	0.0265	0.0028	0.0837
1.5	P5	6.1920	6.1213	6.2203	0.0317	0.0034	0.1023

The graphical results show that increasing the prior standard deviation of $\log(\delta)$ produces small changes in the posterior mean and credible interval for γ , but the relative ordering of the five threshold parameters remains stable. In particular, P5 remains the population with the largest threshold estimate under all prior sensitivity settings considered. Thus, the subset selection conclusion is robust to the prior specifications examined here.

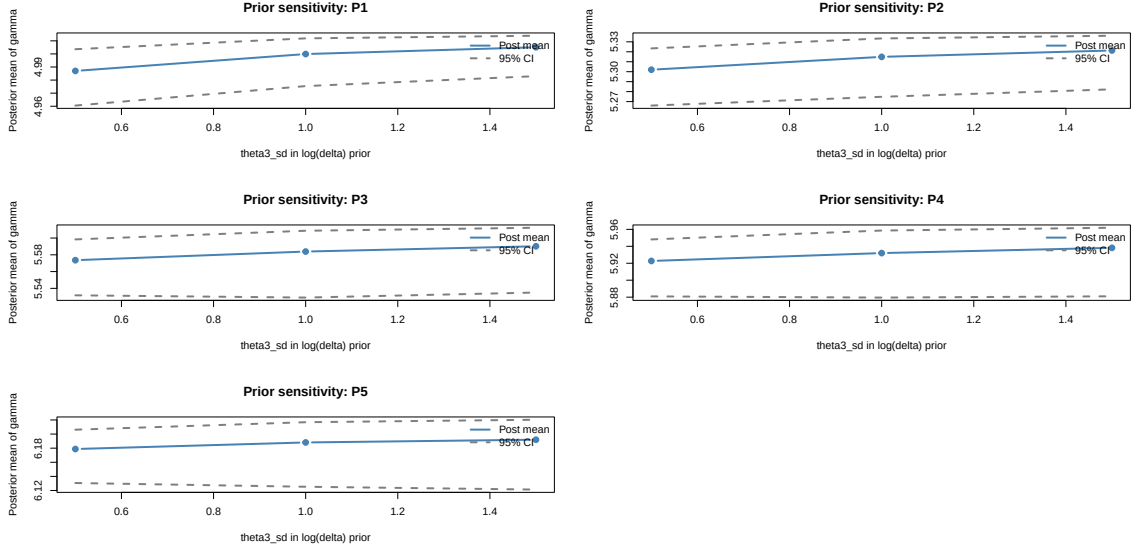


Figure 6.10: Prior sensitivity analysis for the fully simulated Weibull study. Each panel shows the posterior mean and 95% credible interval for γ under different prior standard deviations for $\log(\delta)$.

6.6.8 Posterior Predictive Checks

Posterior predictive checks were used to assess the adequacy of the fitted Bayesian Weibull model, following the posterior predictive model-checking framework of Gelman, Meng, and Stern [43]. Let $T(\mathbf{x}_i)$ denote a summary statistic computed from the observed data for population i , such as the minimum, mean, standard deviation, or a sample percentile. For each retained posterior draw, a replicated dataset $\mathbf{x}_i^{\text{rep},(m)}$ is generated from the fitted Weibull model, and the same statistic is computed as $T(\mathbf{x}_i^{\text{rep},(m)})$. The posterior predictive distribution of the statistic is then approximated by the collection

$$T(\mathbf{x}_i^{\text{rep},(1)}), \dots, T(\mathbf{x}_i^{\text{rep},(M)}).$$

The observed value in Table 6.17 is the statistic computed from the original generated dataset. The simulated mean and simulated standard deviation are the mean and standard deviation of the corresponding statistic computed across the posterior predictive replicated datasets. In this implementation, the Bayesian posterior predictive p -value is computed as

$$p_B = \frac{1}{M} \sum_{m=1}^M I \left(T(\mathbf{x}_i^{\text{rep},(m)}) \geq T(\mathbf{x}_i) \right),$$

Thus, p_B estimates the posterior predictive probability that a replicated statistic is at least as large as the observed statistic. Values of p_B near 0 or 1 would indicate that the observed statistic is unusual under the fitted model, while values away from the extremes suggest that the fitted model can reproduce that feature of the data.

The statistics considered were the minimum, mean, standard deviation, 10th percentile, median, and 90th percentile. Table 6.17 reports the observed statistic, posterior predictive mean, posterior predictive standard deviation, and Bayesian posterior predictive p -value. The selected statistics were chosen to evaluate different features of the fitted Weibull model. The sample minimum assesses the lower-end behavior of the data, which is important because the three-parameter Weibull model includes a threshold parameter. The mean, median, and standard deviation summarize the center and spread of the observed lifetimes. The 10th and 90th percentiles assess early-failure and upper-tail behavior, respectively. Thus, the posterior predictive checks examine whether the fitted model can reproduce the lower endpoint, central tendency, variability, and tail behavior of the generated datasets.

Table 6.17: Posterior predictive checks for the fully simulated study.

Population	Statistic	Observed	Simulated Mean	Simulated SD	Bayes p -value
P1	min	5.0150	5.0125	0.0143	0.385
P1	mean	5.8776	5.8984	0.0869	0.585
P1	sd	0.8368	0.7745	0.1016	0.244
P1	q10	5.1828	5.1437	0.0380	0.146
P1	median	5.5877	5.6918	0.0867	0.893
P1	q90	6.7065	6.9254	0.2193	0.840
P2	min	5.3376	5.3323	0.0196	0.363
P2	mean	6.1907	6.1990	0.0800	0.512
P2	sd	0.6745	0.6984	0.0851	0.570
P2	q10	5.4632	5.4860	0.0412	0.694
P2	median	6.0051	6.0320	0.0830	0.610
P2	q90	7.1824	7.1276	0.1906	0.381
P3	min	5.6147	5.6210	0.0285	0.565
P3	mean	6.4816	6.4835	0.0666	0.498
P3	sd	0.5883	0.5911	0.0565	0.485
P3	q10	5.8237	5.8273	0.0482	0.514
P3	median	6.3791	6.3742	0.0774	0.465
P3	q90	7.2825	7.2807	0.1458	0.474
P4	min	5.9647	5.9655	0.0286	0.504
P4	mean	6.8405	6.8413	0.0684	0.502

Continued on next page

Table 6.17: Posterior predictive checks for the fully simulated study (continued).

Population	Statistic	Observed	Simulated Mean	Simulated SD	Bayes p -value
P4	sd	0.6101	0.6076	0.0605	0.462
P4	q10	6.2095	6.1705	0.0479	0.195
P4	median	6.7162	6.7261	0.0764	0.553
P4	q90	7.6192	7.6655	0.1516	0.609
P5	min	6.2236	6.2399	0.0364	0.658
P5	mean	7.1598	7.1580	0.0670	0.467
P5	sd	0.5892	0.5868	0.0559	0.466
P5	q10	6.4958	6.4857	0.0547	0.422
P5	median	7.0703	7.0639	0.0770	0.470
P5	q90	8.0765	7.9509	0.1408	0.175

The posterior predictive summaries show that the observed statistics are generally close to the corresponding simulated means. The Bayesian posterior predictive p -values are not close to 0 or 1 for any of the statistics considered. This indicates that the fitted Weibull models are able to reproduce the main features of the generated datasets, including the lower endpoint, center, spread, and upper-tail behavior.

Some posterior predictive p -values are farther from 0.5 than others, such as the 10th percentile for P1 and the 90th percentile for P5. However, these values are not extreme and do not provide strong evidence of lack of fit. Overall, the posterior predictive checks support the adequacy of the fitted three-parameter Weibull models for the fully simulated study.

Figure 6.11 displays representative posterior predictive distributions for selected statistics for Population P5. Population P5 is shown in the main chapter because it is the true best population and the population selected by the Bayesian subset-selection rule. The full set of posterior predictive check plots for all five populations is provided in Appendix C, Section C.5.

The figure provides a graphical complement to the numerical posterior predictive summaries in Table 6.17. Within each panel, the distribution is computed from posterior predictive replicated datasets, and the observed value is marked by a vertical line. The observed values for P5 generally fall within the central region of the simulated posterior predictive distributions, indicating that the fitted Weibull model reproduces the main empirical features of the selected population.

The numerical summaries in Table 6.17 show the same pattern across all five populations. The smallest posterior predictive p -values occur for P1 $q_{0.10}$, P4 $q_{0.10}$, and P5 $q_{0.90}$, but these values are not extreme and do not show a systematic pattern of lack of fit across populations or summary statistics. Overall, the posterior predictive checks do not suggest strong lack of fit for the generated datasets. These results support the use of the three-parameter Weibull model as an adequate working model for the fully simulated study, while recognizing that posterior predictive checks are diagnostic tools rather than formal proof that the model is correct.

6.6.9 Summary of the Fully Simulated Study

The fully simulated study shows that the Bayesian subset selection procedure correctly identifies the population with the largest threshold parameter in a controlled setting. The posterior means and medians are close to the true generating values, and the posterior summaries for γ preserve the true ordering of the populations. Both the

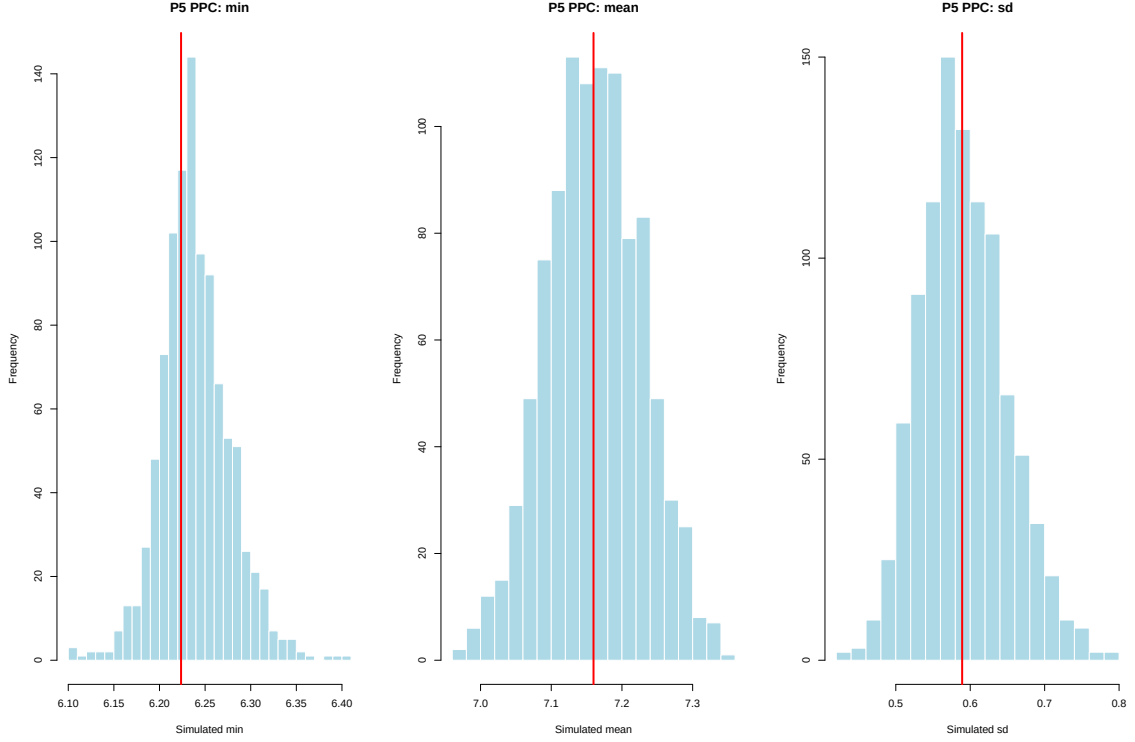


Figure 6.11: Representative posterior predictive checks for Population P5

random-walk Metropolis sampler and the independence Metropolis–Hastings sampler produce nearly identical posterior summaries and selection probabilities.

The posterior probability that P5 has the largest threshold parameter is 1.0000 under both samplers. Therefore, for $P^* = 0.95$, the Bayesian selected subset is

$$\mathcal{S}_B = \{\text{P5}\}.$$

The percentile-based analysis gives the same practical conclusion. P5 has posterior probability 1.0000 of having the largest 10th and 50th percentiles and posterior probability approximately 0.983 of having the largest 90th percentile. The MCMC diagnostics, prior sensitivity analysis, and posterior predictive checks further support the stability of the

posterior inference and the adequacy of the fitted Weibull model for this fully generated study. To replicate the results in Section 6.6, use the R-code in Appendix B.17.

6.7 Alloy T7987 Fatigue Life Data

6.7.1 Data Description

The second application begins with the Alloy T7987 fatigue-life data. The data consist of 67 exact failure times and 5 right-censored observations at 300 thousand cycles. Time is measured in thousands of cycles. The censored observations correspond to specimens that were tested up to 300 thousand cycles without failure.

An alloy is a metallic material formed by combining a base metal with one or more additional elements in order to modify mechanical properties such as strength, hardness, corrosion resistance, and fatigue performance. The Alloy T7987 data represent fatigue lives of specimens from a metallic alloy, where the label T7987 is best interpreted as a material or test designation rather than a fully described chemical alloy grade. In the three-parameter Weibull model used here, the threshold parameter γ represents a lower fatigue-life bound under the test conditions, the scale parameter η describes the characteristic spread of cycles to failure beyond that threshold, and the shape parameter β controls the pattern of failure risk as cyclic loading accumulates. When $\beta > 1$, the implied failure rate increases with accumulated cycles, which is consistent with a fatigue-damage process in which specimens become more likely to fail as repeated loading continues.

The purpose of this section is twofold. First, the original alloy fatigue data are used to examine whether the three-parameter Weibull model provides a reasonable fit to the observed fatigue-life distribution. Second, the fitted model provides the baseline parameter values used later to construct the semi-simulated five-population censored study in Section 6.8.

Both maximum likelihood and Bayesian estimation are considered. The maximum likelihood fit provides a classical point estimate and fitted distributional summaries. The Bayesian fit provides posterior uncertainty for the threshold, scale, and shape parameters, as well as for selected Weibull quantiles.

6.7.2 Maximum Likelihood Fit

The maximum likelihood analysis uses the right-censored three-parameter Weibull likelihood described in Subsection 6.3.2. For an exact failure, the likelihood contribution is the Weibull density. For a right-censored observation, the likelihood contribution is the Weibull survival function.

The maximum likelihood estimates and approximate standard errors are shown in Table 6.18. The estimated threshold parameter is approximately 92.99 thousand cycles. The estimated scale parameter is approximately 93.26, and the estimated shape parameter is approximately 1.32. Since the fitted shape parameter is greater than one, the fitted Weibull model implies an increasing hazard rate.

Table 6.18: Maximum likelihood estimates for the three-parameter Weibull fit to the alloy fatigue data.

Parameter	Estimate	Standard Error
γ	92.9928	1.9516
η	93.2641	9.4082
β	1.3202	0.1565
δ	1.0072	1.9516

The fitted Weibull quantiles from the maximum likelihood estimates are shown in Table 6.19. The fitted median fatigue life is approximately 163.65 thousand cycles. The fitted 90th percentile is approximately 268.42 thousand cycles, indicating a long right tail in the fitted lifetime distribution.

Table 6.19: Fitted Weibull quantiles from the maximum likelihood fit.

p	Fitted quantile, in thousands of cycles
0.10	109.9519
0.25	129.2878
0.50	163.6476
0.75	212.4379
0.90	268.4168

6.7.3 Graphical Assessment of the Maximum Likelihood Fit

Figure 6.12 shows the histogram of the observed failure times with the fitted three-parameter Weibull density overlaid. The fitted density was obtained by substituting the maximum likelihood estimates into this expression:

$$\hat{f}(t) = \frac{\hat{\beta}}{\hat{\eta}} \left(\frac{t - \hat{\gamma}}{\hat{\eta}} \right)^{\hat{\beta}-1} \exp \left[- \left(\frac{t - \hat{\gamma}}{\hat{\eta}} \right)^{\hat{\beta}} \right], \quad t > \hat{\gamma},$$

and it captures the main right-skewed shape of the observed failure-time distribution. The model places most of its mass in the central region of the observed data, with a decreasing right tail extending toward 300 thousand cycles. The histogram shows some discreteness and sampling variation, which is expected because the plot is based on the observed

failure times only. Although the histogram uses only the observed failures, the fitted density is based on the maximum likelihood estimates obtained from the censored likelihood for the full Alloy data.

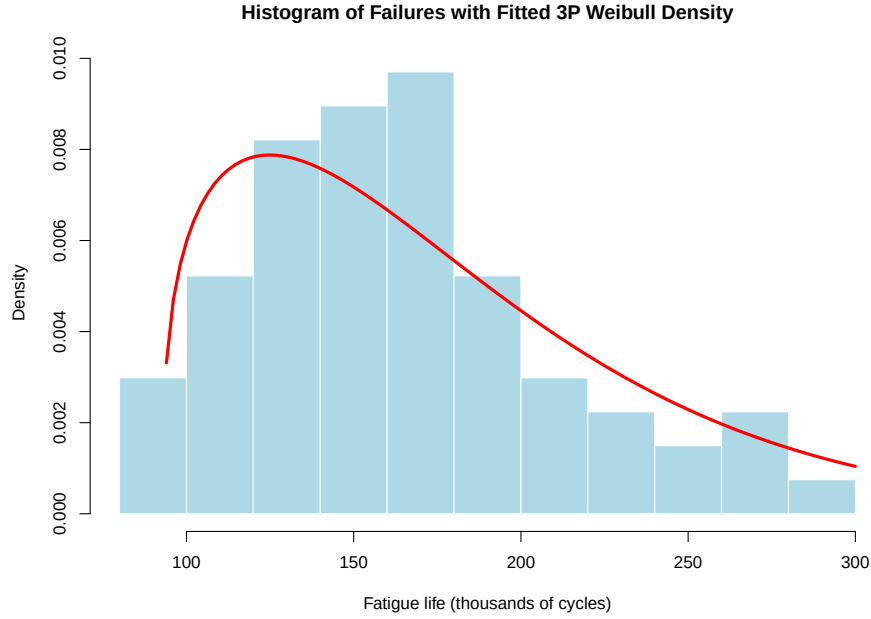


Figure 6.12: Histogram of observed Alloy T7987 failure times with the fitted three-parameter Weibull density.

Figure 6.13 compares the empirical CDF of the observed failures with the fitted three-parameter Weibull CDF. The fitted distribution function was obtained by substituting the maximum likelihood estimates into this expression:

$$\hat{F}(t) = 1 - \exp \left[- \left(\frac{t - \hat{\gamma}}{\hat{\eta}} \right)^{\hat{\beta}} \right], \quad t > \hat{\gamma}.$$

It follows the empirical step function reasonably well through the main body of the data. In the upper tail, the fitted CDF is smoother and remains below one at 300 thousand cycles, reflecting the positive fitted survival probability beyond that time. This behavior is appropriate because the censored specimens survived to 300 thousand cycles and therefore provide information about the upper tail of the failure-time distribution. The empirical CDF shown here is based only on the observed failures, whereas the fitted CDF is based on the maximum likelihood estimates obtained from the censored likelihood for the full Alloy data.

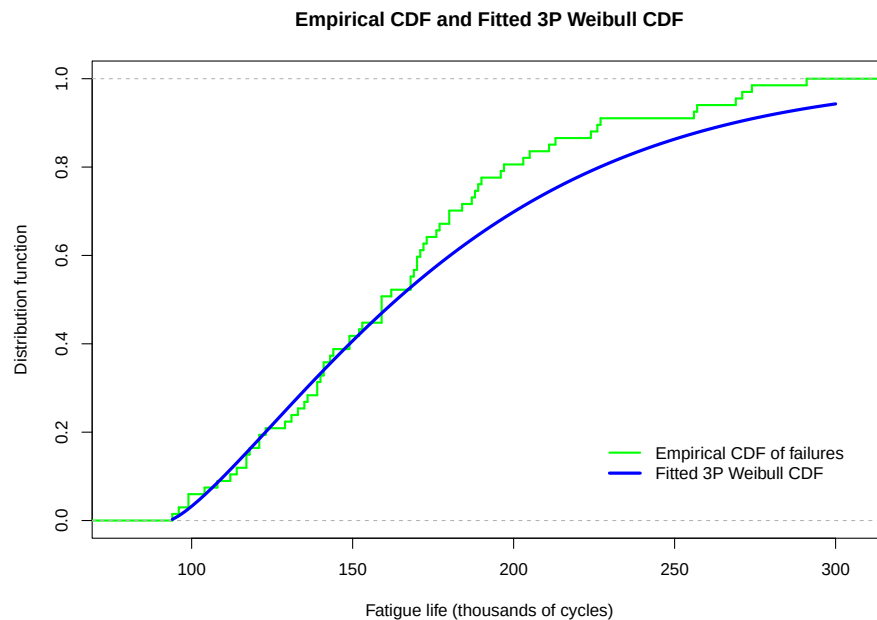


Figure 6.13: Empirical distribution function of the Alloy T7987 failure times with the fitted three-parameter Weibull distribution function.

For the fitted three-parameter Weibull model, the survival function is

$$\hat{S}(t) = \exp \left[- \left(\frac{t - \hat{\gamma}}{\hat{\eta}} \right)^{\hat{\beta}} \right], \quad t > \hat{\gamma}.$$

For comparison, the nonparametric survival estimate was obtained using the Kaplan–Meier estimator,

$$\hat{S}_{KM}(t) = \prod_{t_j \leq t} \left(1 - \frac{d_j}{n_j} \right),$$

where t_j denotes an observed failure time, d_j is the number of failures at t_j , and n_j is the number of specimens at risk immediately before t_j . Right-censored observations do not cause a downward jump in the Kaplan–Meier curve, but they contribute information by reducing the risk set after the censoring time.

Figure 6.14 shows the Kaplan–Meier survival estimate and the fitted three-parameter Weibull survival curve. The fitted survival curve follows the Kaplan–Meier estimate closely over much of the observed range. The survival probability declines gradually for $t > \hat{\gamma}$ and remains positive in the upper tail. This is consistent with the censored specimens at 300 thousand cycles, which indicate survival up to the censoring time. The agreement between the fitted curve and the Kaplan–Meier estimate supports the use of the three-parameter Weibull model as a reasonable working model for the Alloy fatigue-life data.

For the three-parameter Weibull model, the hazard function is

$$h(t) = \frac{f(t)}{S(t)} = \frac{\beta}{\eta} \left(\frac{t - \gamma}{\eta} \right)^{\beta-1}, \quad t > \gamma.$$

Therefore, the fitted hazard function is

$$\hat{h}(t) = \frac{\hat{\beta}}{\hat{\eta}} \left(\frac{t - \hat{\gamma}}{\hat{\eta}} \right)^{\hat{\beta}-1}, \quad t > \hat{\gamma}.$$

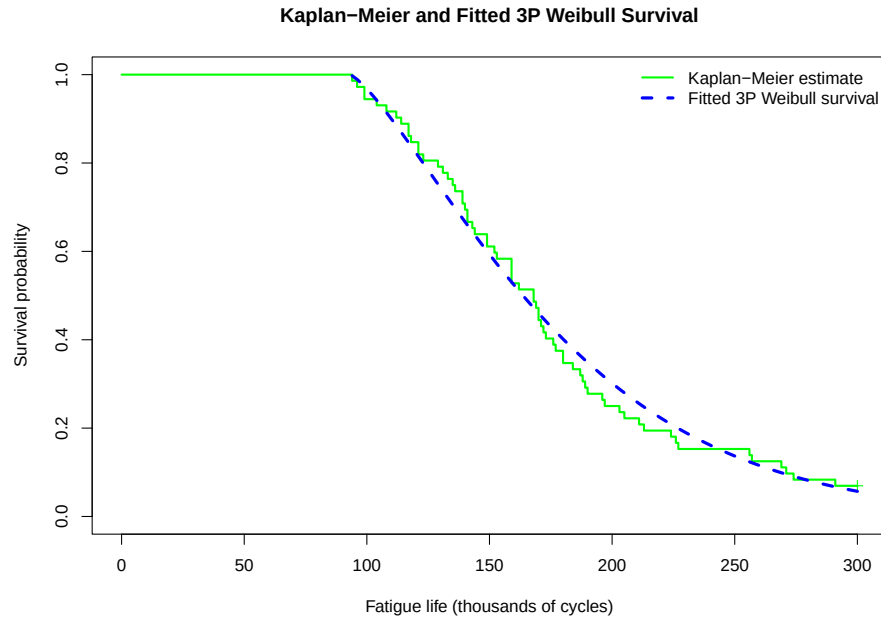


Figure 6.14: Kaplan–Meier survival estimate for the Alloy T7987 data with the fitted three-parameter Weibull survival curve.

Figure 6.15 shows the fitted hazard function for the Alloy data. Since the fitted shape parameter satisfies $\hat{\beta} > 1$, the estimated hazard rate increases with fatigue life. This is consistent with the interpretation of the data as a fatigue or degradation process: as specimens experience more cycles, the instantaneous risk of failure increases. The hazard curve should be interpreted as a model-based summary implied by the fitted three-parameter Weibull distribution rather than as a separate nonparametric estimate of the hazard.

For the Alloy data, a Weibull probability plot was used as a graphical assessment of the fitted three-parameter Weibull model. Under the three-parameter Weibull

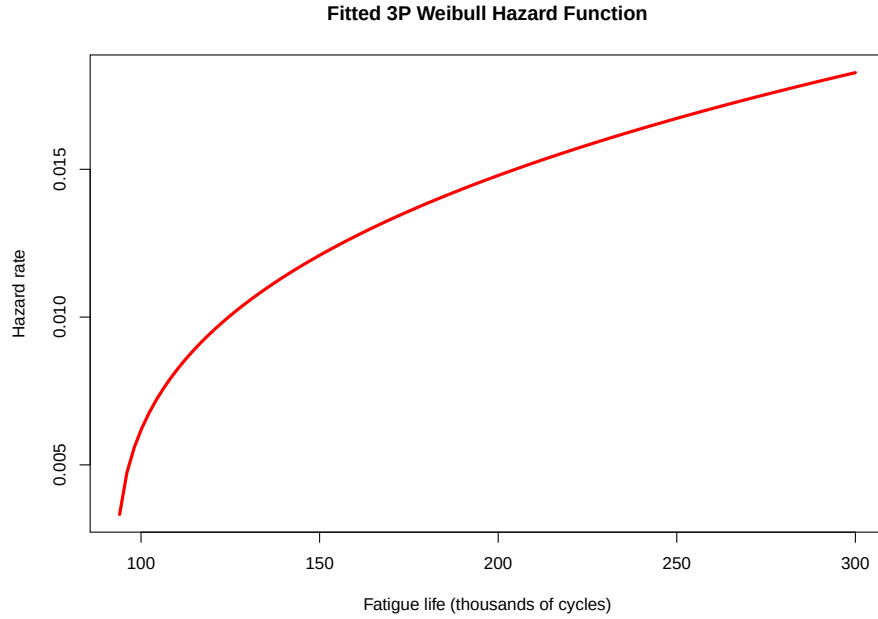


Figure 6.15: Fitted three-parameter Weibull hazard function for the Alloy T7987 fatigue-life data.

distribution, the survival function is

$$S(t) = \exp \left[- \left(\frac{t - \gamma}{\eta} \right)^\beta \right], \quad t > \gamma,$$

Taking the negative logarithm of the survival function gives

$$-\log S(t) = \left(\frac{t - \gamma}{\eta} \right)^\beta.$$

Taking logarithms again yields

$$\log[-\log S(t)] = \beta \log \left(\frac{t - \gamma}{\eta} \right) = \beta [\log(t - \gamma) - \log(\eta)].$$

Therefore, if

$$x = \log(t - \gamma) \quad \text{and} \quad y = \log[-\log S(t)],$$

then the Weibull model implies the linear relationship

$$y = \beta x - \beta \log(\eta).$$

Thus, on a Weibull probability plot, data that are well described by a three-parameter Weibull distribution should approximately follow a straight line. For the censored Alloy data, the survival probabilities were estimated using the Kaplan-Meier estimator, $\hat{S}_{KM}(t)$. The plotted empirical points were therefore computed as

$$x_j = \log(t_j - \hat{\gamma}), \quad y_j = \log[-\log(\hat{S}_{KM}(t_j))],$$

where t_j denotes an observed failure time. The fitted line shown in the plot was obtained by substituting the maximum likelihood estimates $\hat{\beta}$, $\hat{\eta}$, and $\hat{\gamma}$ into the linear Weibull probability plot equation:

$$\hat{y} = \hat{\beta}x - \hat{\beta} \log(\hat{\eta}).$$

The vertical intervals in the probability plot were obtained by transforming the Kaplan–Meier confidence limits at each observed failure time. Let

$$\hat{S}_{KM}(t_j)$$

denote the Kaplan–Meier estimate at failure time t_j , and let

$$[L_j, U_j]$$

denote the corresponding approximate 95% confidence interval for the survival probability. These limits were obtained from the Kaplan–Meier standard error based on Greenwood’s formula and the usual large-sample confidence interval approximation for

the survival estimate. Since the Weibull probability plot uses the transformation

$$g(S) = \log[-\log(S)],$$

and this transformation is decreasing in S for $0 < S < 1$, the transformed confidence interval is

$$[\log\{-\log(U_j)\}, \log\{-\log(L_j)\}].$$

These transformed limits give the vertical intervals shown at

$$x_j = \log(t_j - \hat{\gamma}).$$

The dashed curves give an approximate pointwise confidence band for the fitted Weibull probability line. This band was constructed using the large-sample normal approximation to the maximum likelihood estimator. In the numerical optimization, the parameters were estimated on the transformed scale

$$\theta = (\theta_1, \theta_2, \theta_3)^\top = (\log(\beta), \log(\eta), \log(\delta))^\top,$$

where

$$\delta = \min(t_i) - \gamma - \varepsilon,$$

and $\varepsilon = 10^{-6}$. This transformation ensures that the constrained Weibull parameters are represented on an unconstrained scale during optimization.

Let $\hat{\theta}$ denote the Maximum Likelihood Estimation (MLE) on this transformed scale. The covariance matrix of $\hat{\theta}$ was approximated by

$$\hat{\Sigma} = H(\hat{\theta})^{-1},$$

where

$$H(\hat{\theta}) = \left[\frac{\partial^2 [-\ell(\theta)]}{\partial \theta_r \partial \theta_s} \right]_{\theta=\hat{\theta}}$$

is the observed Hessian matrix of the negative log-likelihood evaluated at the MLE. Using this approximation, plausible values of the transformed parameter vector were generated from

$$\theta^{(b)} \sim N(\hat{\theta}, \hat{\Sigma}), \quad b = 1, \dots, B.$$

Each simulated vector was transformed back to the original parameter scale to obtain

$$\beta^{(b)} = \exp(\theta_1^{(b)}) \quad \text{and} \quad \eta^{(b)} = \exp(\theta_2^{(b)}).$$

For each simulated parameter vector, the corresponding fitted line on the Weibull probability plot scale was computed as

$$y^{(b)}(x) = \beta^{(b)} \left[x - \log(\eta^{(b)}) \right].$$

The lower and upper dashed curves were then obtained as the pointwise 2.5th and 97.5th percentiles of these simulated fitted lines over the grid of x -values.

Figure 6.16 gives the Weibull probability plot with the fitted three-parameter Weibull line. The transformed Kaplan–Meier failure-time points are approximately linear on the Weibull scale, with most points lying close to the fitted line or within the approximate fitted-line confidence band. The vertical intervals show the uncertainty in the Kaplan–Meier estimates, while the dashed curves represent the approximate pointwise confidence band for the fitted Weibull probability line. Overall, the close agreement between the transformed empirical points and the fitted line provides visual support for using the three-parameter Weibull distribution as a reasonable working model for the Alloy fatigue-life data.

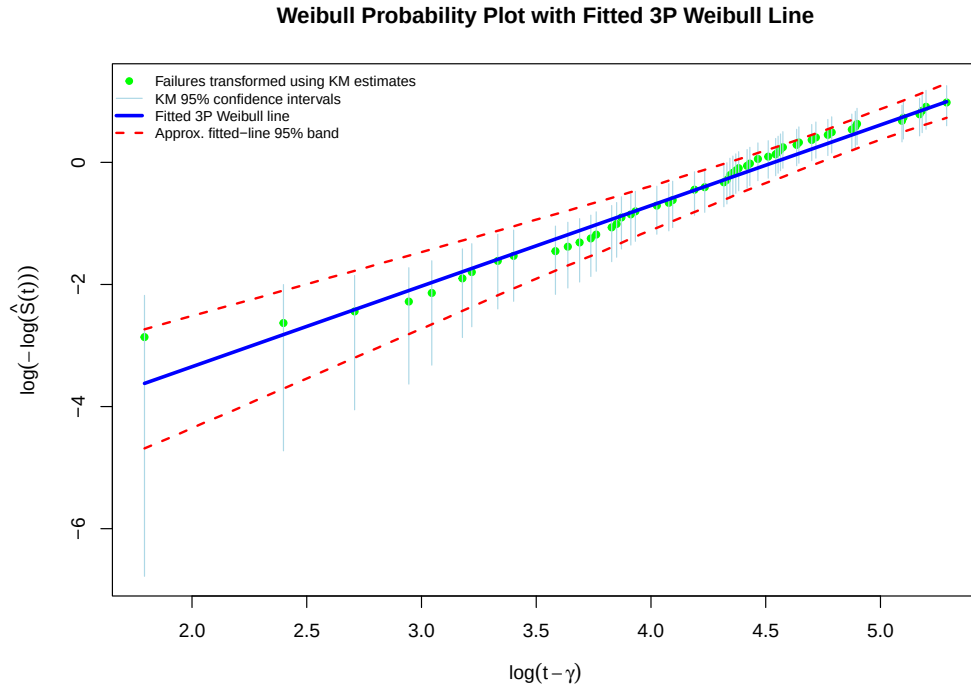


Figure 6.16: Weibull probability plot for the Alloy T7987 data with the fitted three-parameter Weibull line.

Taken together, the density, CDF, survival, hazard, and probability plots indicate that the fitted three-parameter Weibull model gives a reasonable summary of the original fatigue-life distribution. The estimated shape parameter supports an increasing hazard, and the fitted survival curve is consistent with the Kaplan-Meier estimate.

6.7.4 Bayesian Fit to the Original Fatigue-Life Data

A Bayesian three-parameter Weibull model was also fit to the original Alloy T7987 fatigue-life data using the censored likelihood described in Subsection 6.3.2. The likelihood combines Weibull density contributions for exact failure times with

survival-function contributions for the right-censored observations. Posterior simulation was carried out using a custom random-walk Metropolis–Hastings sampler.

Table 6.20 gives the posterior summaries for the Weibull parameters and selected fitted quantiles. The posterior mean of γ is 92.9930, which is nearly identical to the maximum likelihood estimate. The posterior mean of η is 92.7443, and the posterior mean of β is 1.2737. The posterior distribution for β is mostly above one, again supporting an increasing hazard interpretation.

Table 6.20: Bayesian posterior summary for the three-parameter Weibull fit to the alloy fatigue data.

Parameter	Posterior Mean	Posterior SD	2.5%	Median	97.5%	$\hat{R}$	n_{eff}
γ	92.9930	1.9305	87.4495	93.7752	94.0000	1.000	58,531
η	92.7443	9.7957	74.7882	92.2668	113.3203	1.000	58,096
β	1.2737	0.1505	1.0014	1.2659	1.5916	1.000	46,668
δ	1.0070	1.9305	0.0000	0.2248	6.5505	1.000	58,531
$q_{0.10}$	108.9073	3.5747	102.4384	108.7326	116.4007	1.000	58,778
$q_{0.25}$	127.7590	5.5656	117.2165	127.6204	139.0029	1.000	56,201
$q_{0.50}$	162.3866	7.8143	147.5517	162.1672	178.2104	1.000	59,404
$q_{0.75}$	213.2158	11.9393	192.2408	212.3491	239.1924	1.000	57,420
$q_{0.90}$	273.2473	20.4081	240.2941	270.8891	320.2998	1.000	50,338

The posterior uncertainty increases for higher quantiles. For example, the posterior standard deviation is 3.5747 for $q_{0.10}$, 7.8143 for $q_{0.50}$, and 20.4081 for $q_{0.90}$. This is expected because upper-tail behavior is less precisely estimated, especially in the presence of right-censored observations.

Table 6.21 compares the maximum likelihood estimates with the Bayesian posterior summaries for the Weibull parameters. The MLEs are close to the posterior

means and medians. The agreement between the two methods suggests that the data are informative enough to produce stable parameter estimates under both approaches.

Table 6.21: Maximum likelihood and Bayesian posterior comparison for the alloy fatigue data.

Parameter	MLE	Posterior Mean	Median	2.5%	97.5%
β	1.3202	1.2737	1.2659	1.0014	1.5916
δ	1.0072	1.0070	0.2248	0.0000	6.5505
η	93.2641	92.7443	92.2668	74.7882	113.3203
γ	92.9928	92.9930	93.7752	87.4495	94.0000

The posterior median of γ is closer to the smallest observed failure time than the posterior mean. This reflects the skewness of the posterior distribution for the threshold parameter. Because γ must lie below the minimum observed failure time, the posterior distribution can be concentrated near the boundary while still having a left tail.

6.7.5 Posterior Density Plots

Figure 6.17 shows posterior density estimates for γ , η , β , and δ from the Bayesian three-parameter Weibull fit to the Alloy fatigue-life data.

The posterior density of γ in Figure 6.17a is concentrated near the upper boundary imposed by the minimum observed failure time. The density is left-skewed, with most posterior mass near 94 thousand cycles and a tail extending to smaller values. This behavior is typical for a threshold parameter in a three-parameter Weibull model because the threshold must remain below the smallest observed lifetime.

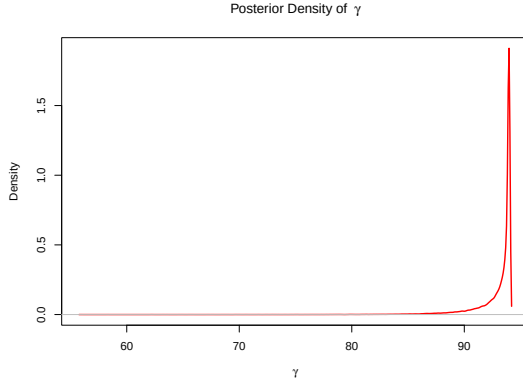
Figure 6.17b shows the posterior density of η . The density is unimodal and centered near 92 thousand cycles, with most posterior mass between approximately 75 and 113 thousand cycles. This interval reflects moderate uncertainty about the scale of the lifetime distribution.

Figure 6.17c shows the posterior density of β . The density is centered around 1.27 and is mostly above one. This confirms the increasing-hazard interpretation obtained from the maximum likelihood fit. The posterior uncertainty for β is moderate, but the lower end of the 95% credible interval is close to one, so the data support an increasing hazard while still allowing values near the constant-hazard boundary.

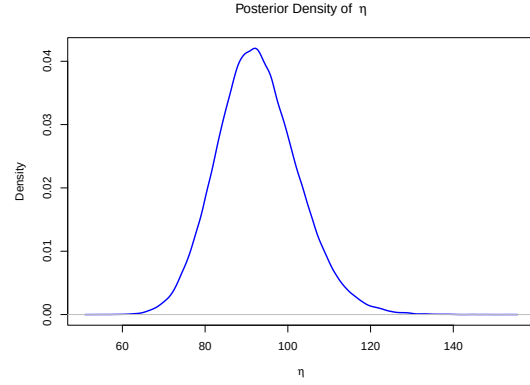
Figure 6.17d shows the posterior density of δ , where

$$\delta = t_{(1)} - \gamma - \varepsilon.$$

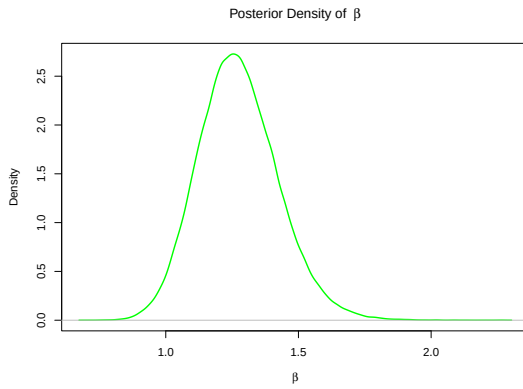
The posterior distribution of δ is strongly right-skewed, with much of the mass near zero. This corresponds to the posterior distribution of γ being concentrated close to the smallest observed failure time. The right tail of δ corresponds to posterior draws where the threshold parameter is farther below the observed minimum.



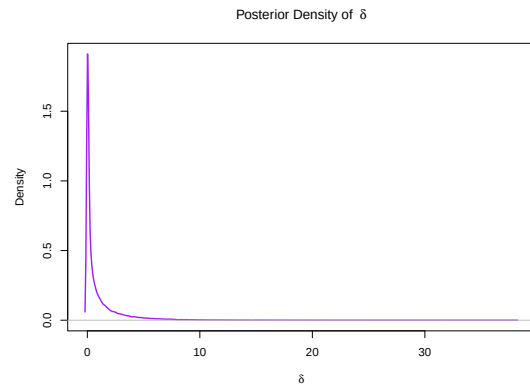
(a) Posterior density of γ .



(b) Posterior density of η .



(c) Posterior density of β .



(d) Posterior density of δ .

Figure 6.17: Posterior density plots for γ , η , β , and δ from the Bayesian three-parameter Weibull fit to the Alloy T7987 fatigue-life data.

6.7.6 Traceplots and MCMC Behavior

Figure 6.18 shows traceplots for the Bayesian fit. The traceplots are displayed for γ , η , β , and δ . The chains fluctuate around stable posterior regions and do not show systematic long-term drift. This supports the convergence assessment reported in Table 6.20, where the $\hat{R}$ values are equal to one for the monitored quantities.

The traceplot for γ in Figure 6.18a shows frequent draws near the upper boundary, with occasional downward movements corresponding to smaller threshold values. This pattern is consistent with the left-skewed posterior density of γ .

The traceplot for η in Figure 6.18b shows stable mixing around the posterior center. The sampled values move throughout the main posterior range without visible trends, suggesting that the sampler is exploring the posterior scale parameter adequately.

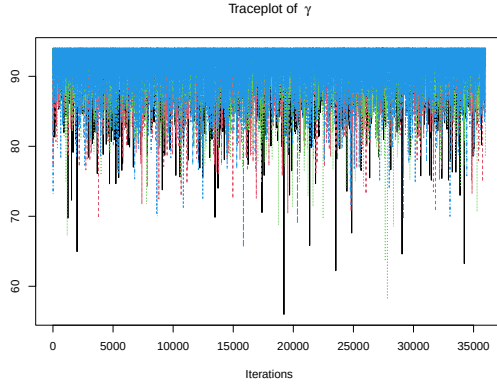
The traceplot for β in Figure 6.18c also shows stable behavior. The sampled shape parameters remain mostly above one, which is consistent with the posterior density and with the fitted increasing hazard function.

The traceplot for δ in Figure 6.18d has a right-skewed pattern with many draws near zero and occasional larger values. This mirrors the posterior density of δ and reflects the sensitivity of the threshold-distance parameter to the smallest observed failure time.

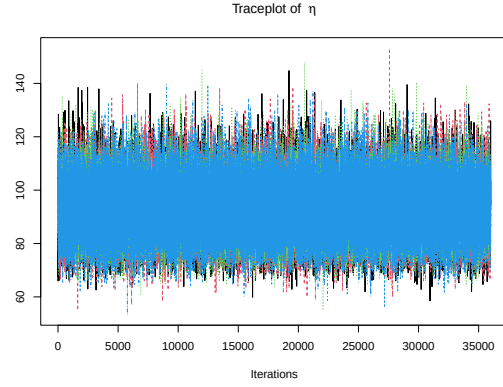
6.7.7 Summary of the Original Fatigue-Life Analysis

The maximum likelihood and Bayesian analyses give consistent conclusions for the original fatigue-life data. Both methods estimate the threshold parameter near 93 thousand cycles and the shape parameter above one, indicating an increasing hazard rate. The fitted density, CDF, survival curve, hazard function, and Weibull probability plot support the use of a three-parameter Weibull model as a reasonable working model for these data.

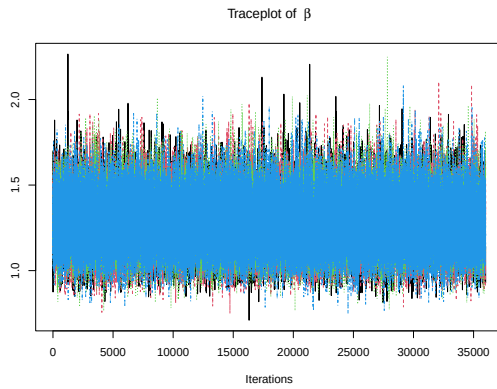
The Bayesian posterior summaries provide uncertainty quantification for the Weibull parameters and fitted quantiles. The posterior distribution of γ is concentrated near the minimum observed failure time, and the posterior distribution of δ is right-skewed. This behavior is expected for the threshold component of the



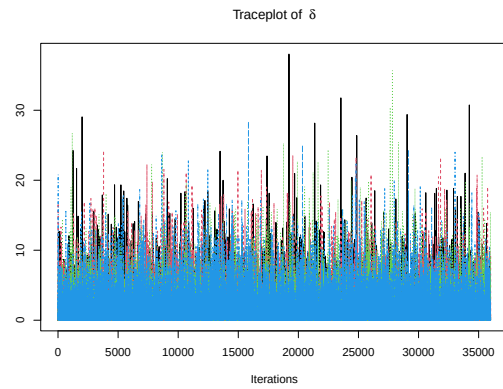
(a) Traceplot of γ .



(b) Traceplot of η .



(c) Traceplot of β .



(d) Traceplot of δ .

Figure 6.18: Traceplots for γ , η , β , and δ from the Bayesian three-parameter Weibull fit to the Alloy T7987 fatigue-life data.

three-parameter Weibull model. The traceplots and convergence diagnostics indicate stable posterior simulation for the monitored quantities.

The posterior summaries also show increasing uncertainty for higher quantiles, especially for $q_{0.90}$, because upper-tail inference is affected by the right-censored observations at 300 thousand cycles. The Bayesian posterior estimates from this section are used as the baseline for constructing the semi-simulated censored study in the next

section. This choice keeps the data-generating setup aligned with the Bayesian subset-selection framework used later in the chapter. Since the maximum likelihood estimates and Bayesian posterior means are very close for the original Alloy data, this choice does not materially change the baseline population. However, the Bayesian estimates provide a natural reference point because they reflect the posterior structure used in the subsequent analysis. The MLE and Bayesian fit results of this section were obtained by the R-code in Appendix B.18–B.19.

6.8 Semi-Simulated Censored Study Based on the Alloy Fatigue Data

6.8.1 Construction of the Semi-Simulated Populations

The final application uses a semi-simulated five-population censored study based on the Alloy T7987 fatigue-life data. Population P1 is based on the Bayesian posterior estimates from the original fatigue-life data. Populations P2–P5 are generated to be similar to the baseline population but with progressively larger threshold parameters. This creates a censored reliability-data example in which the populations have comparable distributional behavior, while the best population is still defined by the largest threshold parameter.

The parameter values used to construct the five populations are shown in Table 6.22. The threshold parameters increase from 92.993 for P1 to 100.993 for P5. The scale and shape parameters are not forced to increase monotonically; instead, they are kept close to the fitted baseline values so that the generated populations remain similar to the original fatigue-life behavior. Thus, the primary ordering of the populations is determined by the threshold parameter γ , rather than by systematic changes in the scale or shape parameters.

Table 6.22: Parameters used for the semi-simulated alloy fatigue study.

Population	γ	η	β	Source
P1	92.993	92.7443	1.2737	Alloy data fitted baseline
P2	94.993	90.8894	1.2482	Generated similar population
P3	96.993	92.7443	1.2737	Generated similar population
P4	98.993	94.5992	1.2992	Generated similar population
P5	100.993	96.4541	1.3246	Generated similar population

The corresponding Weibull percentiles computed from these parameter values are given in Table 6.23. These values provide the reference percentile behavior for the semi-simulated populations. The true threshold parameter is largest for P5, so the true best population under the threshold-based selection rule is P5.

Table 6.23: Weibull percentiles computed from the parameters used for the semi-simulated alloy fatigue study.

Population	γ	η	β	$t_{0.10}$	$t_{0.50}$	$t_{0.90}$
P1	92.993	92.7443	1.2737	108.8411	162.5462	271.5052
P2	94.993	90.8894	1.2482	109.9741	162.7560	272.2885
P3	96.993	92.7443	1.2737	112.8411	166.5462	275.5052
P4	98.993	94.5992	1.2992	115.7279	170.3387	278.7526
P5	100.993	96.4541	1.3246	118.6340	174.1333	282.0284

Each population contains 72 observations. A censoring time of 300 thousand cycles is used, producing both exact failures and right-censored observations. The censoring summary is shown in Table 6.24. The censoring percentage ranges from 6.94%

to 12.50%. Thus, this study is more realistic and more challenging than the fully simulated exact failure-time study because not all lifetimes are observed exactly.

Table 6.24: Censoring summary for the semi-simulated alloy fatigue study.

Population	n	Failures	Censored	Percent Censored	Min Time	Max Time
P1	72	67	5	6.94	94.0000	300
P2	72	63	9	12.50	100.2628	300
P3	72	67	5	6.94	101.7424	300
P4	72	65	7	9.72	101.2739	300
P5	72	67	5	6.94	102.5337	300

6.8.2 Posterior Summaries for Weibull Parameters

The censored three-parameter Weibull model was fit separately to each population using both the random-walk and independent Metropolis–Hastings samplers. The likelihood for each population follows the right-censored Weibull formulation described in Subsection 6.3.2, with density contributions for exact failures and survival-function contributions for right-censored observations. The use of two Metropolis–Hastings samplers provides a numerical check on the stability of the posterior summaries, since both samplers target the same posterior distribution.

The censored three-parameter Weibull model was fit separately to each population using both the random-walk and independent Metropolis–Hastings samplers. The posterior summaries for γ , η , β , and δ are given in Table 6.25.

Table 6.25: Posterior summaries for Weibull parameters in the semi-simulated alloy fatigue study.

Parameter	Pop.	RW Mean	RW Median	RW 2.5%	RW 97.5%	Indep Mean	Indep Median	Indep 2.5%	Indep 97.5%
γ	P1	92.6351	93.2870	87.4915	93.9723	92.6488	93.2354	87.6498	93.9702
γ	P2	99.2982	99.7991	95.3508	100.2460	99.3179	99.7739	95.5132	100.2444
γ	P3	100.3579	101.0531	94.9090	101.7160	100.3841	101.0121	95.0723	101.7153
γ	P4	100.3876	100.8492	96.7317	101.2585	100.3947	100.8273	96.8464	101.2576
γ	P5	101.1260	101.8111	95.7891	102.5059	101.1308	101.7480	95.8964	102.5045
η	P1	94.4843	94.0015	77.1021	114.5755	94.4704	93.9807	77.0393	114.5213
η	P2	102.5622	101.9160	81.2505	127.6475	102.4909	101.8529	81.2284	127.6857
η	P3	98.7007	98.1845	80.0642	120.2761	98.6325	98.1096	79.9111	120.2748
η	P4	102.8857	102.2013	81.3138	128.4027	102.8236	102.0964	81.0843	128.6615
η	P5	97.1267	96.7124	78.9459	117.9522	97.2129	96.7630	79.0520	118.1464
β	P1	1.3145	1.3077	1.0572	1.6130	1.3132	1.3067	1.0508	1.6148
β	P2	1.1512	1.1450	0.9149	1.4246	1.1499	1.1438	0.9104	1.4265
β	P3	1.2832	1.2753	1.0274	1.5845	1.2828	1.2742	1.0215	1.5934
β	P4	1.1231	1.1171	0.8970	1.3835	1.1229	1.1170	0.8930	1.3871
β	P5	1.3025	1.2951	1.0450	1.6035	1.3028	1.2956	1.0410	1.6069

Continued on next page

Table 6.25: Posterior summaries for Weibull parameters in the semi-simulated alloy fatigue study continued.

Parameter	Pop.	RW Mean	RW Median	RW 2.5%	RW 97.5%	Indep Mean	Indep Median	Indep 2.5%	Indep 97.5%
δ	P1	1.3649	0.7130	0.0277	6.5085	1.3512	0.7646	0.0298	6.3502
δ	P2	0.9646	0.4637	0.0168	4.9120	0.9449	0.4889	0.0184	4.7496
δ	P3	1.3845	0.6893	0.0264	6.8334	1.3582	0.7303	0.0271	6.6700
δ	P4	0.8863	0.4247	0.0153	4.5422	0.8791	0.4466	0.0162	4.4274
δ	P5	1.4077	0.7225	0.0277	6.7445	1.4029	0.7856	0.0291	6.6373

The posterior means of γ increase from P1 to P5 overall, but the credible intervals overlap considerably among P3, P4, and P5. This differs from the fully simulated exact-data study, where P5 was identified with posterior probability one. In the censored semi-simulated study, the data support P5 as the most likely best population, but uncertainty remains because the populations are intentionally similar and the observations include right-censoring.

The posterior summaries for η and β also show substantial overlap across populations. This overlap is expected because the semi-simulated populations were constructed to be similar in scale and shape, while the main ordering was induced through the threshold parameter. The posterior means of β are generally greater than one, although the lower credible limits for P2 and P4 fall below one. Thus, the posterior evidence for increasing hazards is strongest for P1, P3, and P5, and weaker for P2 and P4.

Figure 6.19–6.23 show posterior densities for γ , η , and β for all five populations. Each figure corresponds to one population. The random-walk and independence sampler density curves are nearly overlapping, which supports the agreement between the two MCMC methods. The posterior densities for γ shift toward larger values across the populations, reflecting the increasing threshold parameters used to construct the semi-simulated example. For some populations, the random-walk density for γ has a slightly higher peak than the corresponding independent-sampler density, suggesting a slightly more concentrated kernel density estimate. However, the locations and shapes of the two density estimates are very similar, so this difference does not indicate a substantive disagreement between the samplers.

The posterior densities for γ are more irregular and more boundary-influenced than those for η and β , reflecting the threshold constraint in the three-parameter Weibull model. In contrast, the densities for η and β are smoother and show substantial overlap

across populations, which is consistent with the construction of the semi-simulated populations as having similar scale and shape behavior.

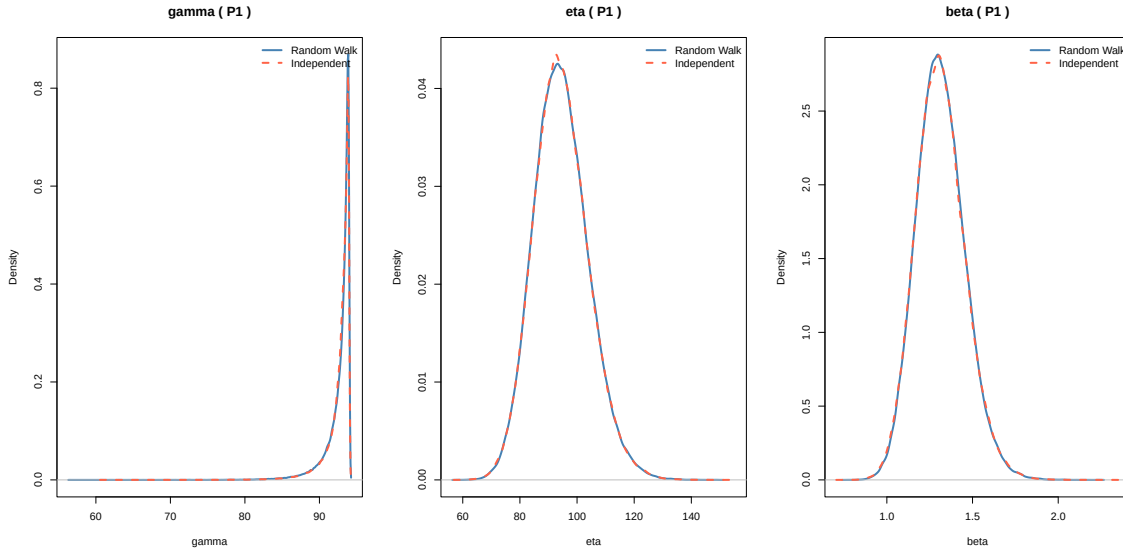


Figure 6.19: Posterior densities of γ , η , and β for P1 in the semi-simulated alloy fatigue study.

6.8.3 Posterior Selection Probabilities

Table 6.26 gives the posterior probabilities of having the largest value of γ , η , and β in the semi-simulated alloy fatigue study. As in the fully simulated study, the formal Bayesian subset selection rule is based on the threshold parameter γ . The posterior probabilities for η and β are reported only as supplementary summaries, since they help describe how the fitted populations compare in terms of scale and shape.

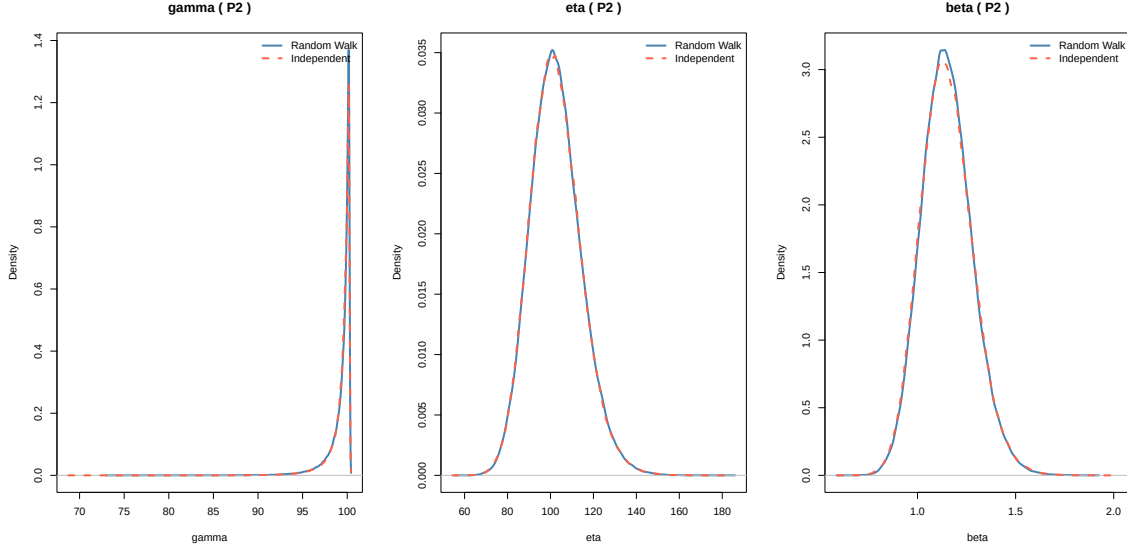


Figure 6.20: Posterior densities of γ , η , and β for P2 in the semi-simulated alloy fatigue study.

For the threshold parameter, P5 has the largest posterior probability of being best, with posterior probability 0.6702 under both samplers. P3 has the second-largest posterior probability, followed by P4. Thus, the posterior evidence favors P5 as the best population under the threshold-based criterion, but the evidence is not concentrated on P5 alone.

The supplementary probabilities for η and β show a different pattern. For the scale parameter η , the largest posterior probabilities are assigned to P4 and P2, rather than P5. For the shape parameter β , the largest posterior probabilities are assigned mainly to P1, P5, and P3. This indicates that the semi-simulated populations are not ordered in the same way for scale and shape as they are for the threshold parameter. This is consistent with the construction of the example, where the threshold parameters were designed to increase across populations, while the scale and shape parameters were kept close to the Alloy T7987 baseline values.

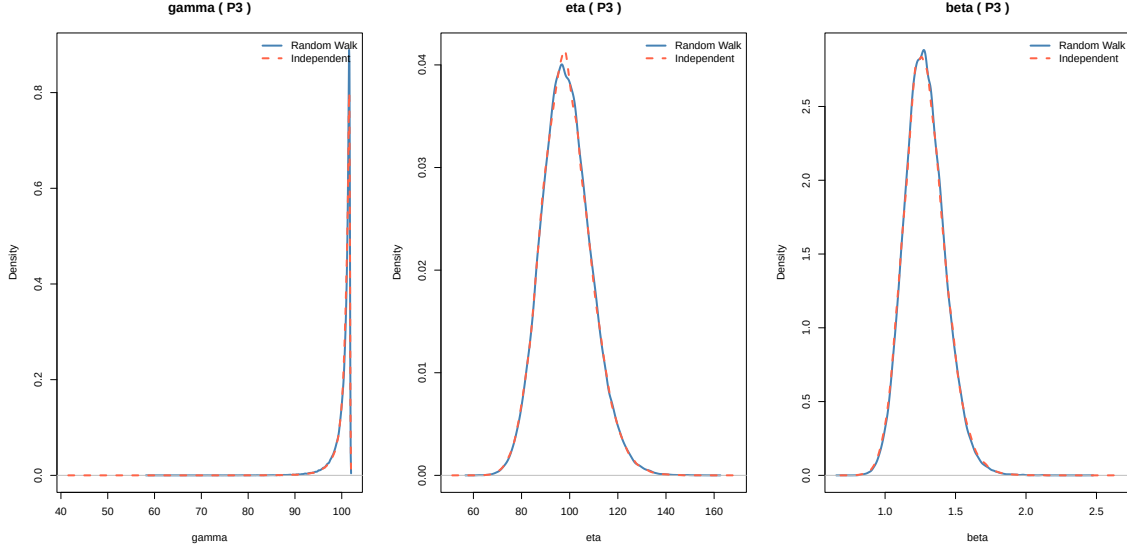


Figure 6.21: Posterior densities of γ , η , and β for P3 in the semi-simulated alloy fatigue study.

The random-walk and independence samplers give nearly identical posterior probabilities for all three parameters. This agreement supports the numerical stability of the posterior ranking summaries. The probabilities for η and β are interpreted descriptively only; the selected subset is still determined by the posterior probabilities for γ .

The Bayesian selected subset is formed using the posterior-probability rule described in Subsection 6.2.5: populations are ordered by their posterior probabilities of being best and added to the subset until the cumulative posterior probability reaches or exceeds the target level P^* .

Table 6.27 gives the ranked posterior probabilities and cumulative probabilities for the threshold parameter γ . Unlike Table 6.26, which also includes supplementary

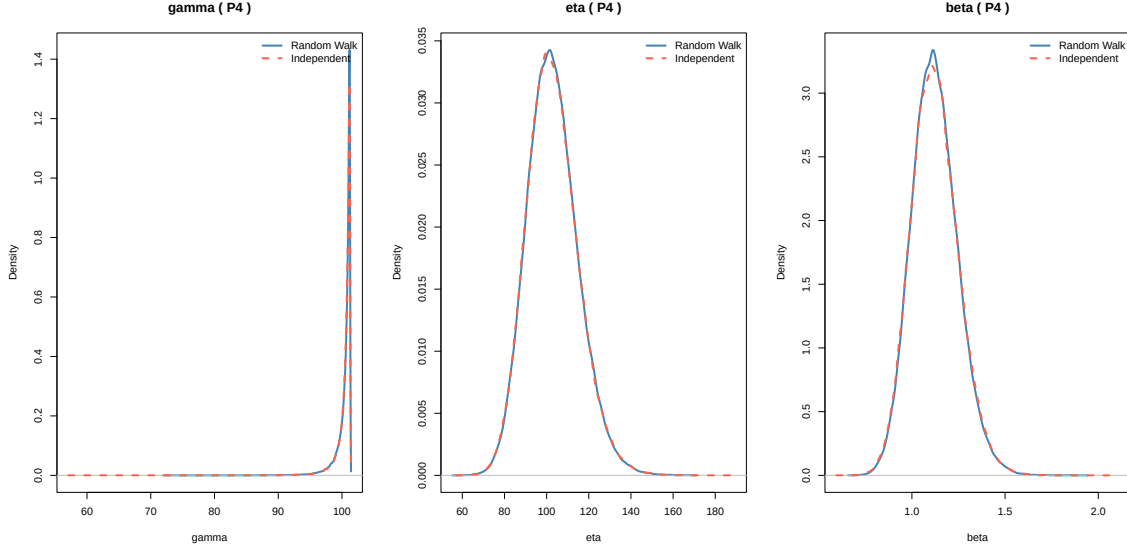


Figure 6.22: Posterior densities of γ , η , and β for P4 in the semi-simulated alloy fatigue study.

summaries for η and β , this table is restricted to γ because the formal selected subset is based on the threshold parameter.

For $P^* = 0.95$, P5 alone is not sufficient because its posterior probability of having the largest threshold parameter is 0.6702. Adding P3 increases the cumulative probability to 0.8880 under the random-walk sampler and 0.8885 under the independence sampler, which is still below 0.95. Adding P4 increases the cumulative probability to 0.9942 under the random-walk sampler and 0.9949 under the independence sampler.

Therefore, for $P^* = 0.95$, the Bayesian selected subset is

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

The same selected subset is obtained under both samplers, indicating that the subset selection result is stable with respect to the proposal mechanism.

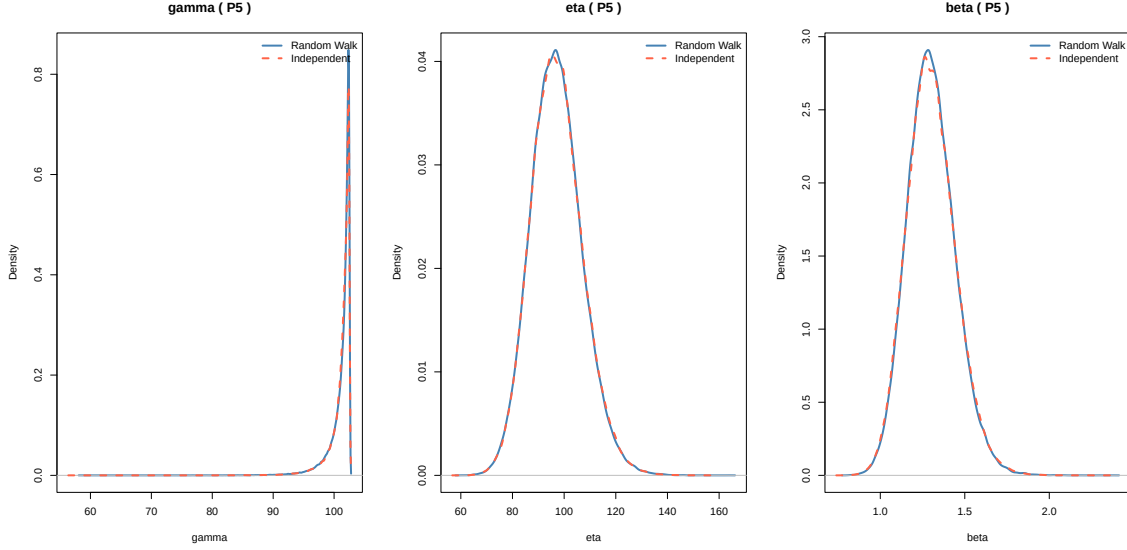


Figure 6.23: Posterior densities of γ , η , and β for P5 in the semi-simulated alloy fatigue study.

This result illustrates an important feature of Bayesian subset selection. In the fully simulated exact-data study, P5 was selected alone because the posterior probability that P5 had the largest threshold parameter was essentially one. In the semi-simulated censored study, P5 is still the most likely best population, but the posterior uncertainty is larger because the populations are intentionally similar and the observations include right-censoring. The selected subset therefore expands to include P3 and P4 in order to reach the posterior confidence level $P^* = 0.95$.

The small posterior probability for P2 and the rounded zero probability for P1 indicate that these populations are rarely, or never, identified as having the largest threshold parameter among the retained posterior draws. Thus, although the credible intervals for some threshold parameters overlap, see Table 6.25, the posterior ranking still places most of the evidence on P5, P3, and P4.

Table 6.26: Posterior probabilities of having the largest parameter value in the semi-simulated alloy fatigue study.

Parameter	Population	Random-Walk Probability	Independence Probability
γ	P1	0.0000	0.0000
γ	P2	0.0058	0.0052
γ	P3	0.2178	0.2183
γ	P4	0.1062	0.1064
γ	P5	0.6702	0.6702
η	P1	0.0841	0.0847
η	P2	0.3005	0.2992
η	P3	0.1712	0.1689
η	P4	0.3130	0.3125
η	P5	0.1311	0.1347
β	P1	0.3443	0.3401
β	P2	0.0552	0.0568
β	P3	0.2578	0.2575
β	P4	0.0346	0.0358
β	P5	0.3081	0.3099

Table 6.27: Posterior selection probabilities and cumulative probabilities for γ in the semi-simulated alloy fatigue study.

Rank	Population	Random-Walk Sampler		Independence Sampler	
		Probability	Cumulative	Probability	Cumulative
1	P5	0.6702	0.6702	0.6702	0.6702
2	P3	0.2178	0.8880	0.2183	0.8885
3	P4	0.1062	0.9942	0.1064	0.9949
4	P2	0.0058	1.0000	0.0052	1.0000
5	P1	0.0000	1.0000	0.0000	1.0000

6.8.4 Posterior Weibull Percentiles

The posterior summaries for $t_{0.10}$, $t_{0.50}$, and $t_{0.90}$ are shown in Table 6.28. These percentiles provide a complementary reliability interpretation to the threshold-based analysis. The threshold parameter γ describes the lower endpoint of the lifetime distribution, while the Weibull percentiles describe early, median, and upper-tail failure-time behavior.

For each retained posterior draw, the Weibull percentile was computed as

$$t_p = \gamma + \eta[-\log(1 - p)]^{1/\beta},$$

for $p = 0.10, 0.50$, and 0.90 . The posterior summaries in Table 6.28 were then obtained from the resulting posterior samples of each percentile. Thus, the table reflects uncertainty in all three Weibull parameters, not only uncertainty in the threshold parameter.

The random-walk and independence samplers give nearly identical posterior percentile summaries, which is consistent with the agreement observed in the parameter summaries and posterior selection probabilities. The posterior means and medians for the 10th percentile generally increase from P1 to P5, with P5 having the largest estimated early-life percentile. This agrees with the construction of the semi-simulated populations, where P5 has the largest threshold parameter.

For the median percentile $t_{0.50}$, the posterior means for P3, P4, and P5 are very close. This shows that the median lifetime behavior is not determined by γ alone. Although P5 has the largest threshold parameter, the scale and shape parameters also contribute to the percentile values, producing substantial overlap among the posterior summaries for the middle of the distribution.

The difference is more pronounced for the upper-tail percentile $t_{0.90}$. The posterior means for P2 and P4 are larger than the posterior mean for P5, even though P5 has the

largest threshold parameter. This occurs because the upper-tail percentile is especially sensitive to η and β through the term $\eta[-\log(1-p)]^{1/\beta}$. Therefore, the population with the largest threshold parameter does not necessarily have the largest upper-tail percentile.

The credible intervals for the percentiles also widen as p increases from 0.10 to 0.90. This reflects greater uncertainty in the upper tail of the fitted Weibull distributions, especially in the presence of right-censoring. Thus, the percentile analysis provides a useful complement to the threshold-based selection analysis: the threshold parameter determines the formal definition of the best population in this chapter, while the percentiles describe how the fitted lifetime distributions compare across different regions of the distribution.

Table 6.28: Posterior summaries for Weibull percentiles in the semi-simulated alloy fatigue study.

Percentile	Pop.	RW Mean	RW Median	RW 2.5%	RW 97.5%	Indep Mean	Indep Median	Indep 2.5%	Indep 97.5%
$t_{0.10}$	P1	109.7321	109.5697	103.2518	117.0878	109.7159	109.5507	103.1463	117.2191
$t_{0.10}$	P2	113.8767	113.6704	107.6105	121.3511	113.8537	113.6409	107.5342	121.4328
$t_{0.10}$	P3	117.4985	117.3070	110.8216	125.2442	117.5023	117.3048	110.7174	125.3783
$t_{0.10}$	P4	114.3337	114.1122	108.2031	121.7494	114.3335	114.0969	108.1579	121.8942
$t_{0.10}$	P5	118.4326	118.2619	111.7844	126.0091	118.4606	118.2974	111.7294	126.1869
$t_{0.50}$	P1	163.9910	163.7715	149.7081	179.5748	163.9703	163.7468	149.5575	179.6216
$t_{0.50}$	P2	173.6953	173.4009	156.7573	192.4502	173.6260	173.3257	156.6944	192.5280
$t_{0.50}$	P3	174.3818	174.1452	159.0831	190.9779	174.3415	174.1373	158.9390	191.0467
$t_{0.50}$	P4	174.4447	174.1088	157.2479	193.6622	174.3966	173.9913	157.1266	193.8260
$t_{0.50}$	P5	174.2870	174.1015	159.4248	190.3783	174.3555	174.1803	159.4226	190.5887
$t_{0.90}$	P1	272.2456	270.0755	240.7136	316.3531	272.3907	270.3126	240.5135	316.6262
$t_{0.90}$	P2	313.3131	309.7197	267.9615	379.4890	313.4751	309.7728	267.7531	380.5075
$t_{0.90}$	P3	291.0458	288.6808	256.4270	339.4615	291.0628	288.6315	256.1233	340.3035
$t_{0.90}$	P4	318.8830	315.3333	272.8170	385.5267	318.8841	315.1879	272.6654	386.5038
$t_{0.90}$	P5	286.8809	284.7052	253.7826	332.6281	287.0838	284.8145	253.8047	333.1237

Table 6.29 gives posterior probabilities of having the largest Weibull percentile for $t_{0.10}$, $t_{0.50}$, and $t_{0.90}$. For each retained posterior draw, the population with the largest value of the corresponding percentile was identified. The posterior probability was then estimated as the proportion of retained posterior draws in which each population had the largest percentile value. The populations are ranked separately for each percentile and sampler, and the cumulative probabilities show how the Bayesian selected subset would be formed under a percentile-based selection rule.

Table 6.29: Ranked posterior selection probabilities and cumulative probabilities for largest Weibull percentiles in the semi-simulated alloy fatigue study.

Percentile	Rank	Random-Walk Sampler			Independence Sampler		
		Pop.	Prob.	Cumulative	Pop.	Prob.	Cumulative
$t_{0.10}$	1	P5	0.4670	0.4670	P5	0.4635	0.4635
$t_{0.10}$	2	P3	0.3431	0.8101	P3	0.3418	0.8053
$t_{0.10}$	3	P4	0.0971	0.9072	P4	0.1007	0.9060
$t_{0.10}$	4	P2	0.0827	0.9899	P2	0.0830	0.9890
$t_{0.10}$	5	P1	0.0100	1.0000	P1	0.0110	1.0000
$t_{0.50}$	1	P4	0.2636	0.2636	P4	0.2643	0.2643
$t_{0.50}$	2	P3	0.2428	0.5064	P3	0.2401	0.5044
$t_{0.50}$	3	P2	0.2339	0.7403	P5	0.2368	0.7412
$t_{0.50}$	4	P5	0.2329	0.9732	P2	0.2317	0.9729
$t_{0.50}$	5	P1	0.0268	1.0000	P1	0.0272	1.0000
$t_{0.90}$	1	P4	0.4603	0.4603	P4	0.4567	0.4567
$t_{0.90}$	2	P2	0.3612	0.8215	P2	0.3627	0.8194
$t_{0.90}$	3	P3	0.0926	0.9141	P3	0.0930	0.9124
$t_{0.90}$	4	P5	0.0650	0.9791	P5	0.0659	0.9783
$t_{0.90}$	5	P1	0.0210	1.0000	P1	0.0217	1.0000

For $t_{0.10}$, P5 has the largest posterior probability of having the largest early-life percentile under both samplers. However, the probability is only 0.4670 under the random-walk sampler and 0.4635 under the independence sampler. Therefore, P5 alone would not be sufficient to reach a target posterior probability of $P^* = 0.95$. The cumulative probabilities show that P5, P3, P4, and P2 would be needed to exceed the target level for $t_{0.10}$ under both samplers.

For $t_{0.50}$, the posterior probabilities are spread across P2, P3, P4, and P5. No single population dominates the posterior distribution of the largest median lifetime. The rankings from the two samplers are very similar, although P2 and P5 switch order between the two samplers. This small difference is not practically important because the posterior probabilities for these populations are nearly equal. For $P^* = 0.95$, the selected percentile-based subset for $t_{0.50}$ would include P2, P3, P4, and P5.

For $t_{0.90}$, P4 has the largest posterior probability of having the largest upper-tail percentile, followed by P2. This differs from the threshold-based selection result, where P5 has the largest posterior probability of being best. The difference occurs because upper-tail percentiles depend not only on γ , but also on η and β . Therefore, a population with the largest threshold parameter does not necessarily have the largest upper-tail lifetime percentile.

Overall, the percentile-based results provide a different reliability perspective from the threshold-based selection rule. The threshold-based analysis identifies the population most likely to have the largest lower endpoint, while the percentile-based analysis compares the populations at specific regions of the fitted lifetime distribution. In this censored study, the early-life percentile favors P5 most strongly, the median percentile shows substantial ambiguity among P2–P5, and the upper-tail percentile favors P4 and P2.

As in the fully simulated study, joint agreement was used to compare threshold-based and percentile-based selection in the semi-simulated alloy fatigue study. For each retained posterior draw, the population with the largest threshold parameter was compared with the population having the largest Weibull percentile t_p . The resulting agreement probability estimates the posterior probability that the two selection criteria identify the same population. Table 6.30 gives the agreement probabilities for $t_{0.10}$, $t_{0.50}$, and $t_{0.90}$.

Table 6.30: Joint agreement between threshold-based and percentile-based selection in the semi-simulated alloy fatigue study.

Percentile	RW Agreement Probability	Independence Agreement Probability
$t_{0.10}$	0.3804	0.3710
$t_{0.50}$	0.2095	0.2104
$t_{0.90}$	0.1281	0.1304

The agreement probability is much lower in the semi-simulated censored study than in the fully simulated exact-data study. For $t_{0.10}$, the agreement probability is approximately 0.38, meaning that the population with the largest threshold parameter is also the population with the largest early-life percentile in only about 38% of retained posterior draws. The agreement decreases to approximately 0.21 for $t_{0.50}$ and approximately 0.13 for $t_{0.90}$.

This decline shows that percentile-based selection gives a different reliability comparison from threshold-based selection in the censored semi-simulated study. The formal selection rule in this chapter is based on the threshold parameter γ , because γ

defines the best population. However, because the Weibull percentile depends on the threshold, scale, and shape parameters together, uncertainty in the scale and shape parameters can change the percentile rankings, especially for upper-tail percentiles such as $t_{0.90}$.

Overall, these results show that the threshold-based and percentile-based comparisons are less aligned in the semi-simulated censored study than in the fully simulated exact-data study. This is expected because the populations are intentionally similar, the observations include right-censoring, and the upper-tail behavior is more sensitive to uncertainty in the fitted Weibull scale and shape parameters.

6.8.5 MCMC Diagnostics

Representative trace plots are shown in Figures 6.24–6.26. Population P1 represents the Alloy-based baseline population, while P5 is the population with the largest true threshold parameter and the population with the largest posterior probability of being best. The full set of trace plots for all five populations under both samplers is provided in Appendix C, Section C.6. The trace plots are shown for the transformed parameters $\log(\beta)$, $\log(\eta)$, and $\log(\delta)$, since these are the parameters sampled directly by the MCMC algorithms. The threshold parameter γ is then recovered from δ through the constraint imposed by the minimum observed failure time.

The displayed trace plots show stable behavior across the four chains, with no visible long-term drift. The chains fluctuate around common posterior regions for both samplers, supporting the convergence assessment from the Gelman–Rubin diagnostics and effective sample sizes reported below.

The trace plots for $\log(\delta)$ show more variability than those for $\log(\beta)$ and $\log(\eta)$. This is expected because δ controls the distance between the threshold parameter and the

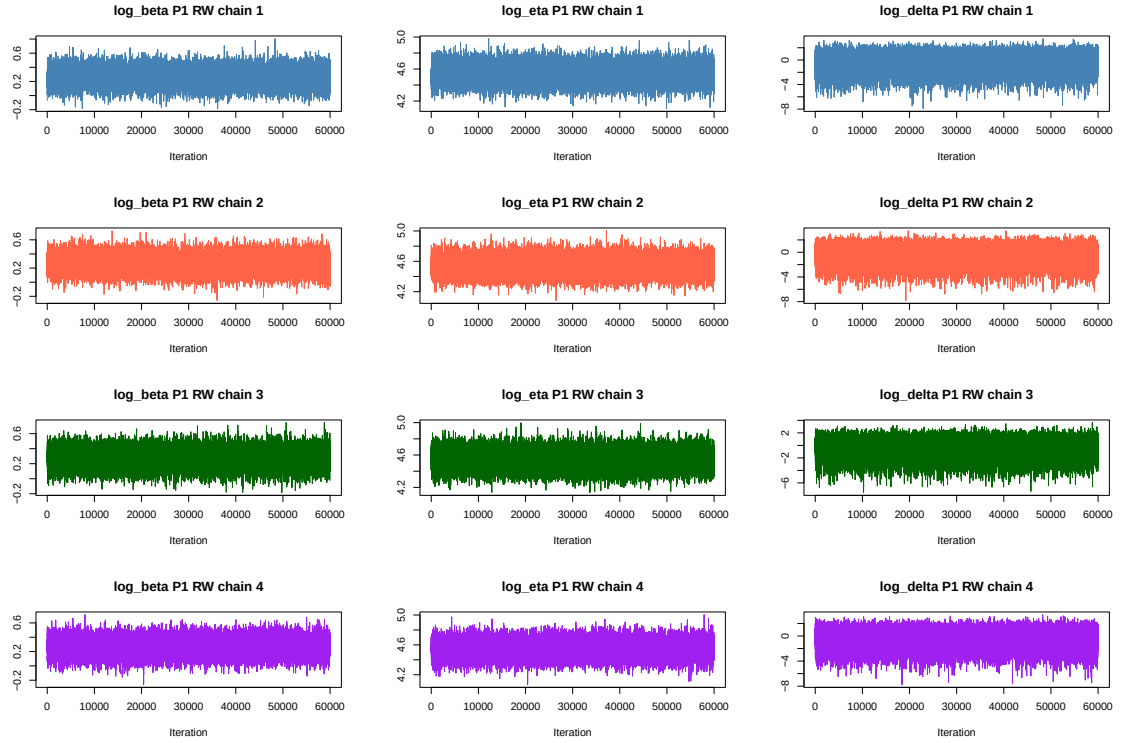


Figure 6.24: Random-walk trace plots for Population P1

smallest observed time, so it is more directly affected by the boundary constraint on γ . The behavior is similar for the baseline population and the selected population, and the random-walk and independence samplers show comparable mixing patterns for P5. Overall, the representative trace plots do not suggest serious convergence problems for the semi-simulated alloy fatigue study.

Figure 6.27 shows representative autocorrelation functions for γ , η , and β under both samplers for Population P5. The full set of autocorrelation plots for all five populations is provided in Appendix C, Section C.7. The autocorrelations decrease with lag for both samplers. As in the fully simulated study, the independence sampler generally

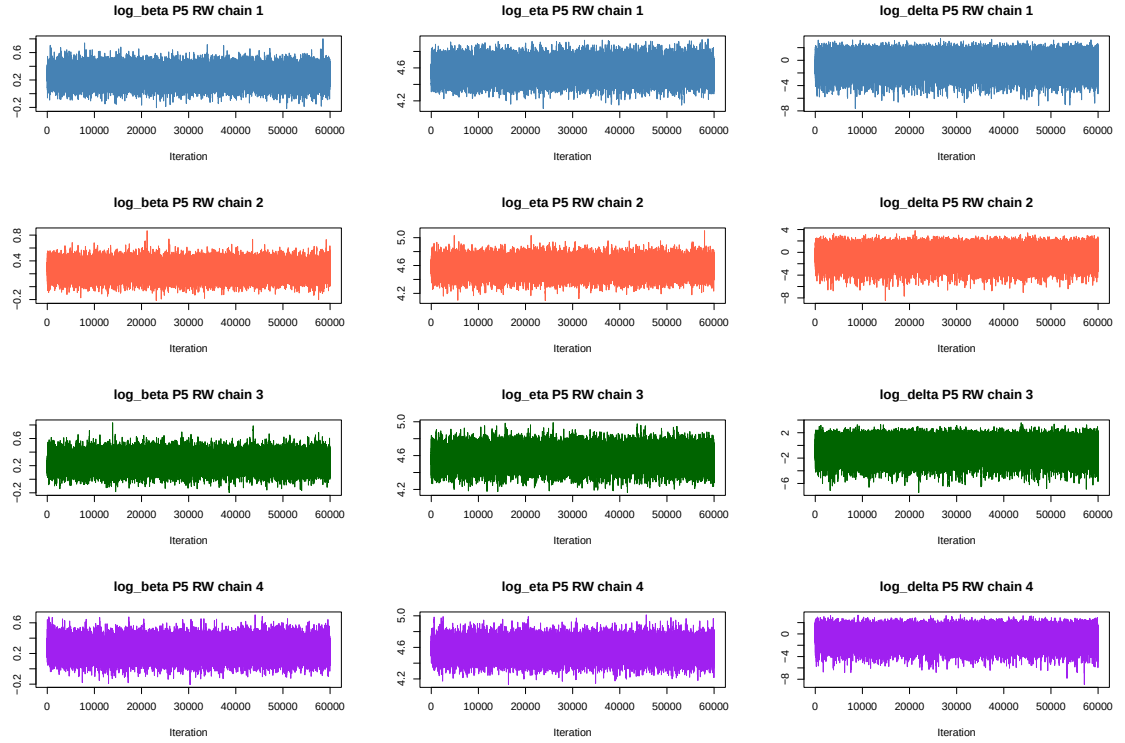


Figure 6.25: Random-walk trace plots for Population P5

shows lower autocorrelation than the random-walk sampler, indicating more efficient movement through the posterior distribution.

The autocorrelation functions decrease with increasing lag under both samplers, indicating that dependence between retained draws weakens as the lag increases. The independent Metropolis–Hastings sampler generally shows smaller autocorrelation at short lags than the random-walk Metropolis sampler. This is consistent with the purpose of the independent proposal, which uses a proposal distribution designed to approximate the posterior distribution more globally.

The reduction in autocorrelation is relevant for effective posterior sampling, since lower serial dependence corresponds to larger effective sample sizes. Although some

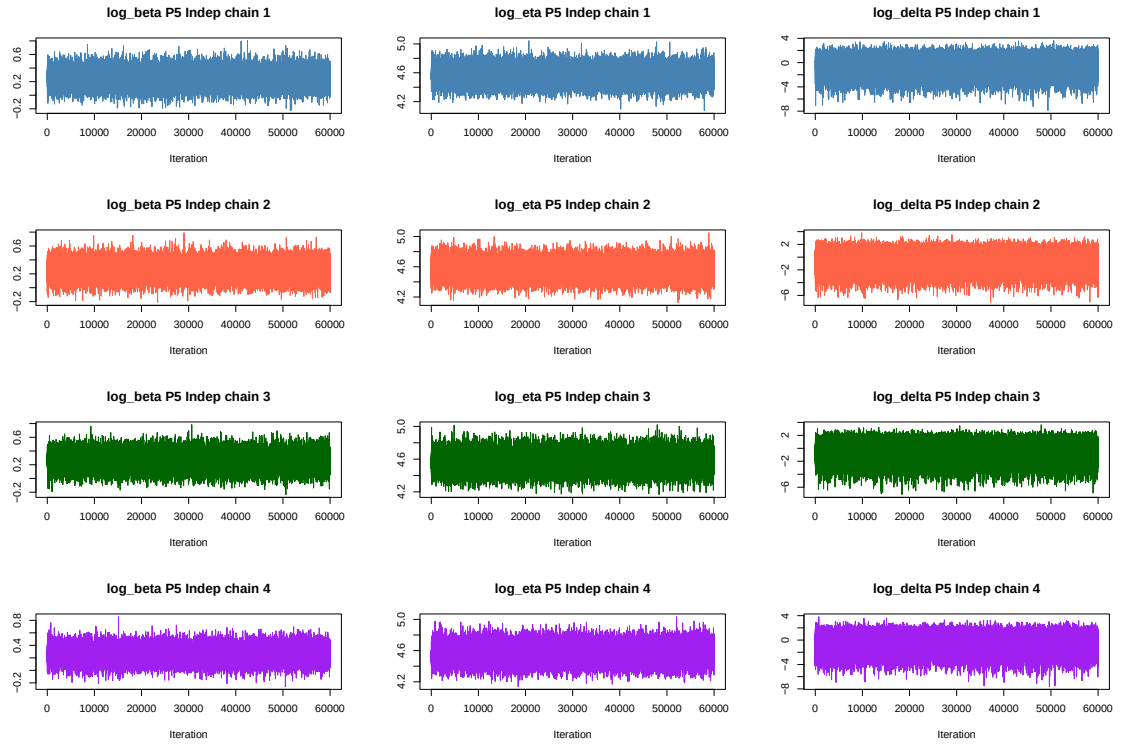


Figure 6.26: Independence-sampler trace plots for Population P5

autocorrelation is still visible at small lags, especially for parameters related to the threshold component, the overall autocorrelation pattern supports adequate mixing. This is relevant in the semi-simulated alloy fatigue study because the populations are intentionally similar and the data include right-censored observations, both of which can increase posterior uncertainty. Together with the trace plots, Gelman–Rubin diagnostics, and effective sample sizes, the ACF results support the stability of the MCMC output for the semi-simulated censored study.

The acceptance rates are shown in Table 6.31. The random-walk sampler has mean acceptance rates between 0.356 and 0.367, while the independence sampler has mean

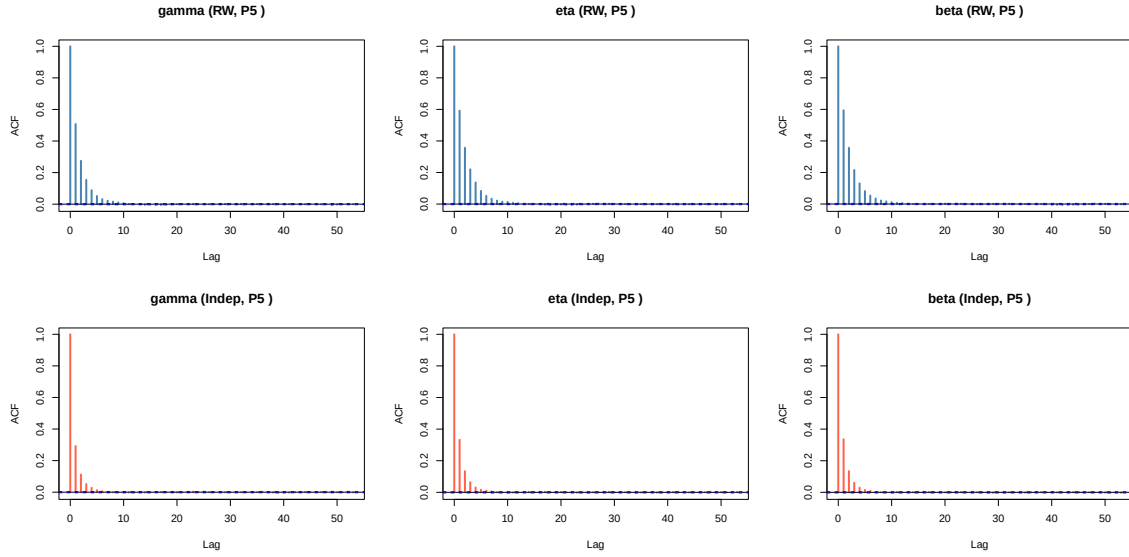


Figure 6.27: Autocorrelation functions for γ , η , and β for Population P5

acceptance rates between 0.241 and 0.247. These values are reasonable for the three-dimensional Metropolis–Hastings updates used here.

Table 6.31: Acceptance rates across chains for the semi-simulated alloy fatigue study.

Population	Random-Walk Sampler			Independence Sampler		
	Mean	Min	Max	Mean	Min	Max
P1	0.364	0.355	0.373	0.243	0.242	0.244
P2	0.367	0.351	0.374	0.245	0.243	0.247
P3	0.357	0.347	0.364	0.242	0.240	0.243
P4	0.367	0.362	0.371	0.247	0.245	0.249
P5	0.356	0.354	0.358	0.241	0.240	0.242

The minimum and maximum acceptance rates across the four chains are close to the corresponding population means, indicating stable tuning behavior across chains. The acceptance rates are also similar across the five populations, even though the populations differ slightly in their censoring patterns and fitted Weibull parameters. This suggests that the proposal distributions remained adequate for all five censored datasets.

As in the fully simulated study, the independence sampler has lower acceptance rates than the random-walk sampler. This does not indicate poorer performance by itself, because acceptance rates must be interpreted together with autocorrelation and effective sample size. In this study, the independence sampler also shows lower autocorrelation and larger effective sample sizes, so the lower acceptance rate is consistent with more efficient posterior exploration. Overall, the acceptance-rate results support the conclusion that both samplers are adequately tuned for the semi-simulated alloy fatigue study.

Table 6.32 gives the Gelman–Rubin $\hat{R}$ diagnostics for the transformed Weibull parameters in the semi-simulated alloy fatigue study. Since the diagnostic was defined in the fully simulated study, it is used here to assess whether the four chains for each sampler mixed into the same posterior region.

All $\hat{R}$ values are essentially equal to one for both samplers and for all transformed parameters. The largest point estimate is 1.0002, which occurs for $\log(\delta)$ in P2 under the random-walk sampler. The upper confidence limits from the diagnostic output are also very close to one, with all parameters flagged as acceptable. These results support convergence of the four chains for both the random-walk and independent Metropolis–Hastings samplers.

The agreement across populations is important in this semi-simulated censored study because the five populations are intentionally similar and include right-censored observations. Despite this added posterior uncertainty, the between-chain and

within-chain variability are nearly identical for the monitored transformed parameters. Together with the trace plots, autocorrelation functions, acceptance rates, and effective sample sizes, the Gelman–Rubin diagnostics support the stability of the MCMC output.

Table 6.32: Gelman–Rubin $\hat{R}$ diagnostics for the transformed Weibull parameters in the semi-simulated alloy fatigue study.

Population	Parameter	Random-Walk $\hat{R}$	Independence $\hat{R}$
P1	$\log(\beta)$	1.0000	1.0000
	$\log(\eta)$	1.0000	1.0000
	$\log(\delta)$	1.0001	1.0000
P2	$\log(\beta)$	1.0001	1.0000
	$\log(\eta)$	1.0000	1.0000
	$\log(\delta)$	1.0002	1.0000
P3	$\log(\beta)$	1.0000	1.0000
	$\log(\eta)$	1.0001	1.0001
	$\log(\delta)$	1.0000	1.0001
P4	$\log(\beta)$	1.0000	1.0000
	$\log(\eta)$	1.0001	1.0000
	$\log(\delta)$	1.0000	1.0000
P5	$\log(\beta)$	1.0000	1.0000
	$\log(\eta)$	1.0000	1.0000
	$\log(\delta)$	1.0000	1.0000

Effective sample sizes for the semi-simulated alloy fatigue study are shown in Table 6.33. Since effective sample size was defined in the fully simulated study, it is used here as a numerical measure of the amount of independent posterior information

contained in the retained MCMC draws. Large effective sample sizes indicate that the posterior summaries and selection probabilities are based on stable simulation output.

The effective sample sizes are large for all populations and Weibull parameters under both samplers. This is important in the semi-simulated alloy fatigue study because the populations are intentionally similar and the observations include right-censoring, both of which can increase posterior uncertainty. Despite this added complexity, the retained MCMC samples provide a large amount of effective posterior information.

As in the fully simulated study, the independence sampler produces larger effective sample sizes than the random-walk sampler for every population and parameter. This agrees with the ACF plots, where the independence sampler generally shows lower autocorrelation. Nevertheless, the random-walk sampler also produces sufficiently large effective sample sizes for stable posterior inference. Taken together with the trace plots, acceptance rates, and Gelman–Rubin diagnostics, the ESS results support the stability of the MCMC output for the semi-simulated censored study.

Table 6.33: Effective sample sizes for Weibull parameters in the semi-simulated alloy fatigue study.

Population	Random-Walk Sampler			Independence Sampler		
	γ	η	β	γ	η	β
P1	71,185.3	60,418.6	61,219.3	120,551.4	112,495.3	108,488.3
P2	73,088.3	61,920.3	61,814.6	125,931.8	111,193.8	111,992.1
P3	70,547.1	61,426.5	59,945.5	118,642.5	110,955.4	109,331.5
P4	75,599.0	61,831.1	63,010.4	127,908.4	114,748.1	114,623.0
P5	73,169.0	59,393.1	60,380.8	119,128.7	110,861.6	110,483.8

Table 6.34 gives effective sample sizes for the posterior Weibull percentiles $t_{0.10}$, $t_{0.50}$, and $t_{0.90}$. These values assess the Monte Carlo stability of the posterior percentile summaries and the percentile-based selection probabilities reported earlier.

The percentile effective sample sizes are large for all populations and percentile levels. The random-walk sampler produces ESS values of approximately 59,000 to 66,600, while the independence sampler produces ESS values of approximately 108,500 to 116,800. Thus, the posterior summaries for the early-life, median, and upper-tail percentiles are based on stable simulation output.

The upper-tail percentile $t_{0.90}$ has slightly smaller random-walk ESS values than $t_{0.10}$ and $t_{0.50}$ for several populations. This is consistent with the greater posterior uncertainty in upper-tail reliability quantities, especially in the presence of right-censoring. However, the ESS values for $t_{0.90}$ remain large, so the upper-tail percentile summaries and selection probabilities are not driven by Monte Carlo instability.

As with the parameter effective sample sizes, the independence sampler gives larger ESS values than the random-walk sampler for all percentile quantities. This agrees with the autocorrelation results and supports the conclusion that the independence sampler is more efficient in terms of effective posterior draws. Nevertheless, both samplers provide sufficiently large effective sample sizes for the percentile-based analysis.

6.8.6 Prior Sensitivity Analysis

The prior sensitivity analysis for the semi-simulated alloy fatigue study varies the prior standard deviation of

$$\theta_3 = \log(\delta)$$

over the values 1.0, 2.0, and 3.0. The purpose is to examine whether posterior inference for the threshold component is sensitive to the prior spread assigned to the threshold-gap

Table 6.34: Effective sample sizes for Weibull percentiles in the semi-simulated alloy fatigue study.

Percentile	Population	ESS (RW)	ESS (Independence)
$t_{0.10}$	P1	65,734.9	108,107.5
$t_{0.10}$	P2	66,606.1	112,703.8
$t_{0.10}$	P3	66,120.5	109,948.2
$t_{0.10}$	P4	66,358.9	112,416.5
$t_{0.10}$	P5	65,441.6	108,708.5
$t_{0.50}$	P1	63,157.7	111,019.3
$t_{0.50}$	P2	63,814.7	114,129.8
$t_{0.50}$	P3	62,890.6	110,830.2
$t_{0.50}$	P4	64,096.2	114,994.4
$t_{0.50}$	P5	61,209.0	108,569.0
$t_{0.90}$	P1	59,389.3	113,820.6
$t_{0.90}$	P2	60,102.4	111,196.6
$t_{0.90}$	P3	60,978.0	109,533.5
$t_{0.90}$	P4	60,959.2	116,827.8
$t_{0.90}$	P5	59,214.3	111,173.6

parameter. Since the Gelman–Rubin diagnostics, trace plots, and effective sample sizes support stable MCMC behavior, the sensitivity analysis focuses on prior influence rather than sampler instability.

Table 6.35 reports posterior means and 95% credible intervals for γ and δ under the three prior specifications. The posterior means of γ are fairly stable across the three prior specifications. The largest posterior mean of γ is assigned to P5 for all three values of σ_{θ_3} , which is consistent with the threshold-based construction of the semi-simulated populations. The ordering among P3 and P4 changes slightly across prior choices, reflecting the fact that these populations are close to each other and have overlapping

posterior uncertainty. However, the lower-threshold populations P1 and P2 remain separated from the populations with larger threshold values.

The sensitivity is more visible in the credible intervals than in the posterior means. As the prior standard deviation of $\log(\delta)$ increases, the lower credible limits for γ generally become smaller and the upper credible limits for δ become larger. This behavior is expected because a more diffuse prior on $\log(\delta)$ allows a wider range of threshold-gap values. Since γ is recovered from δ , greater uncertainty in δ translates into greater uncertainty in the lower tail of the posterior distribution of γ .

Overall, the prior sensitivity analysis indicates that the numerical uncertainty in the threshold component is affected by the prior spread assigned to $\log(\delta)$, especially through the credible interval widths. However, the main threshold-based conclusion is stable: P5 remains the population with the largest posterior mean threshold parameter under all three prior specifications. Thus, the semi-simulated alloy fatigue results do not appear to be driven by a single prior choice for the threshold-gap parameter.

Figure 6.28 displays the prior sensitivity results graphically. Each panel shows the posterior mean of γ together with the corresponding 95% credible interval under different prior standard deviations for $\log(\delta)$. The figure provides a visual summary of how the threshold estimates change as the prior on the threshold-gap parameter becomes more diffuse.

The graphical results are consistent with the numerical summaries in Table 6.35. The posterior means of γ remain fairly stable across the three prior settings, while the credible intervals become wider for some populations as the prior standard deviation increases. This effect is most visible in the lower credible limits for γ , reflecting the additional uncertainty introduced by allowing a wider range of values for δ . This behavior

Table 6.35: Prior sensitivity analysis for the prior standard deviation of $\log(\delta)$ in the semi-simulated alloy fatigue study.

σ_{θ_3}	Population	γ			δ		
		Mean	2.5%	97.5%	Mean	2.5%	97.5%
1.0	P1	92.7606	89.6122	93.8455	1.2394	0.1545	4.3878
1.0	P2	99.1787	96.4537	100.1380	1.0841	0.1247	3.8091
1.0	P3	100.4459	97.1532	101.5919	1.2964	0.1505	4.5891
1.0	P4	100.2029	97.4500	101.1515	1.0710	0.1224	3.8239
1.0	P5	101.2285	98.0709	102.3784	1.3051	0.1552	4.4627
2.0	P1	92.6324	87.4752	93.9744	1.3676	0.0256	6.5248
2.0	P2	99.2650	95.1171	100.2470	0.9978	0.0154	5.1457
2.0	P3	100.3599	94.8437	101.7151	1.3824	0.0272	6.8986
2.0	P4	100.3941	96.7059	101.2572	0.8798	0.0167	4.5680
2.0	P5	101.0613	95.3648	102.5068	1.4724	0.0268	7.1688
3.0	P1	92.7067	86.9282	93.9932	1.2933	0.0068	7.0718
3.0	P2	99.4497	95.3755	100.2606	0.8130	0.0022	4.8873
3.0	P3	100.3908	94.1172	101.7349	1.3515	0.0074	7.6252
3.0	P4	100.5443	96.8818	101.2720	0.7296	0.0018	4.3921
3.0	P5	101.2140	95.3336	102.5284	1.3196	0.0052	7.2000

is expected because the threshold parameter is constrained by the smallest observed time and is recovered through the threshold-gap parameter.

The figure also shows that the populations remain close to each other in the upper part of the threshold ordering, especially P3, P4, and P5. This is consistent with the construction of the semi-simulated study, where the populations were designed to be similar while having increasing threshold parameters. Although the credible intervals overlap, P5 retains the largest posterior mean of γ under all three prior settings. Thus, the

graphical sensitivity analysis supports the conclusion that the main threshold-based ordering is not driven by a single choice of prior standard deviation for $\log(\delta)$.

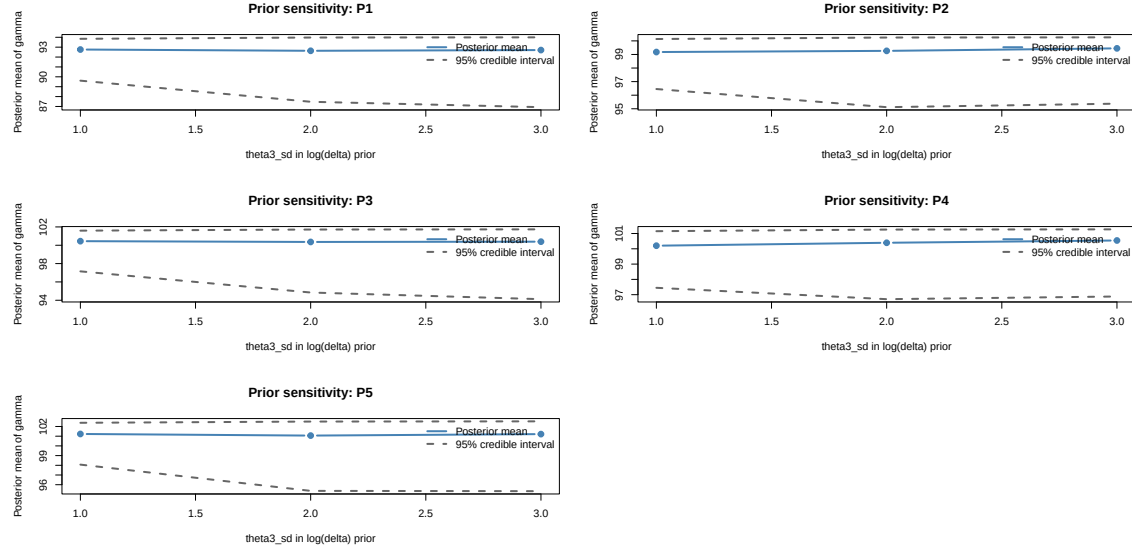


Figure 6.28: Prior sensitivity analysis for the semi-simulated alloy fatigue study. Each panel shows the posterior mean and 95% credible interval for γ under different prior standard deviations for $\log(\delta)$.

6.8.7 Posterior Predictive Checks

Posterior predictive checks were also performed for the semi-simulated censored alloy fatigue study. As in the fully generated study, replicated datasets were generated from the posterior predictive distribution and compared with the observed data through selected summary statistics. For this censored study, the replicated failure times were subjected to the same right-censoring mechanism used in the observed datasets. Thus, the

posterior predictive checks evaluate not only the fitted lifetime distribution, but also whether the fitted censored Weibull model can reproduce the observed censoring patterns.

The summary statistics considered were the minimum, mean, standard deviation, number of failures, number censored, 10th percentile, median, and 90th percentile. The minimum assesses the lower-end behavior of the fitted distribution, which is important for the threshold parameter. The mean, standard deviation, and median summarize the center and spread of the observed censored lifetimes. The 10th and 90th percentiles assess early-life and upper-tail behavior, while the numbers of failures and censored observations assess whether the fitted model reproduces the censoring structure of the data.

Table 6.36 reports the observed statistic, posterior predictive mean, posterior predictive standard deviation, and Bayesian posterior predictive p -value for each population. Since the definition of the posterior predictive p -value was given in the fully simulated study, the values here are interpreted in the same way: values close to 0 or 1 would suggest that the observed statistic is unusual under the fitted model, while values away from the extremes indicate that the replicated datasets are broadly consistent with the observed data.

Table 6.36: Posterior predictive checks for the semi-simulated alloy fatigue study.

Population	Statistic	Observed	Simulated Mean	Simulated SD	Bayesian p -value
P1	min	94.0000	95.9950	3.1391	0.768
P1	mean	175.3750	175.8072	9.6209	0.499
P1	sd	56.6112	57.8555	5.6218	0.594
P1	failures	67.0000	67.4880	2.5746	0.679
P1	censored	5.0000	4.5120	2.5746	0.459

Continued on next page

Table 6.36: Posterior predictive checks for the semi-simulated alloy fatigue study continued.

Population	Statistic	Observed	Simulated Mean	Simulated SD	Bayesian p -value
P1	$q_{0.10}$	114.3000	110.8919	6.0948	0.267
P1	median	168.0000	163.4865	11.9556	0.334
P1	$q_{0.90}$	270.8000	266.1134	22.6558	0.420
P2	min	100.2628	101.9343	2.5220	0.797
P2	mean	186.1235	187.4811	10.0187	0.541
P2	sd	60.7892	63.1098	5.0235	0.680
P2	failures	63.0000	63.5760	3.6482	0.666
P2	censored	9.0000	8.4240	3.6482	0.442
P2	$q_{0.10}$	112.7832	115.1663	5.8706	0.630
P2	median	173.0520	173.3997	14.0769	0.489
P2	$q_{0.90}$	300.0000	289.9691	15.1365	0.442
P3	min	101.7424	103.6314	3.0323	0.781
P3	mean	187.0320	186.9945	10.0105	0.478
P3	sd	58.7341	58.8626	5.3535	0.516
P3	failures	67.0000	65.5480	3.3967	0.450
P3	censored	5.0000	6.4520	3.3967	0.681
P3	$q_{0.10}$	117.2868	119.0195	6.2141	0.591
P3	median	179.9863	175.4707	12.8185	0.353
P3	$q_{0.90}$	270.3756	280.6171	19.9685	0.698
P4	min	101.2739	102.6960	2.4912	0.770

Continued on next page

Table 6.36: Posterior predictive checks for the semi-simulated alloy fatigue study continued.

Population	Statistic	Observed	Simulated Mean	Simulated SD	Bayesian p -value
P4	mean	190.0673	188.8168	10.3807	0.438
P4	sd	65.0918	63.3217	5.1311	0.370
P4	failures	65.0000	63.2070	3.8855	0.396
P4	censored	7.0000	8.7930	3.8855	0.696
P4	$q_{0.10}$	111.8123	115.9132	5.6355	0.761
P4	median	172.4624	174.9584	14.5296	0.543
P4	$q_{0.90}$	296.5872	290.8386	15.0168	0.627
P5	min	102.5337	104.5633	3.3021	0.765
P5	mean	186.2949	186.0054	9.8349	0.462
P5	sd	58.2943	57.7230	5.0729	0.459
P5	failures	67.0000	66.2290	3.0446	0.507
P5	censored	5.0000	5.7710	3.0446	0.638
P5	$q_{0.10}$	123.2893	119.8504	6.4697	0.271
P5	median	168.3278	174.4247	12.5942	0.685
P5	$q_{0.90}$	272.7212	277.4630	19.8809	0.587

The posterior predictive summaries show that the observed statistics are generally close to the corresponding posterior predictive means. The Bayesian posterior predictive p -values are not close to 0 or 1 for any of the statistics considered, suggesting that the observed semi-simulated datasets do not appear unusual under the fitted censored Weibull models.

The checks for the numbers of failures and censored observations are particularly important in this study because each population contains right-censored observations. The observed numbers of failures and censored observations are consistent with the replicated datasets across all five populations. This suggests that the fitted models reproduce the overall censoring behavior reasonably well.

The posterior predictive checks also show reasonable agreement for the mean, standard deviation, and median, indicating that the fitted models capture the center and spread of the censored lifetime distributions. The 10th and 90th percentile checks provide additional support for the lower-tail and upper-tail behavior. Some values are farther from 0.5, such as the minimum for P2 and the 10th percentile for P5, but these values are not extreme and do not provide strong evidence of lack of fit.

Overall, the posterior predictive checks support the adequacy of the fitted censored three-parameter Weibull models for the semi-simulated alloy fatigue study. The fitted models reproduce the main features of the observed data, including the lower endpoint, central tendency, variability, censoring counts, and tail behavior.

Figure 6.29 displays representative posterior predictive distributions for selected statistics for Population P5. Population P5 is shown in the main chapter because it has the largest true threshold parameter and the largest posterior probability of being best under the threshold-based selection rule. The full set of posterior predictive check plots for all five populations is provided in Appendix C, Section C.8.

The figure provides a graphical complement to the numerical posterior predictive summaries in Table 6.36. Within each panel, the histogram shows the distribution of the statistic computed from posterior predictive replicated datasets, and the vertical line marks the observed value. For P5, the observed minimum, mean, and standard deviation fall within the main body of the simulated posterior predictive distributions, indicating that the

fitted censored Weibull model can reproduce the lower endpoint, central tendency, and variability of the selected population.

The numerical posterior predictive summaries in Table 6.36 show a similar pattern across all five populations. The Bayesian posterior predictive p -values are not close to 0 or 1 for any of the statistics considered, including the numbers of failures and censored observations. Thus, the posterior predictive checks do not provide evidence of substantial lack of fit for the semi-simulated censored datasets. Overall, the fitted censored three-parameter Weibull models reproduce the main features of the observed data, including the lower endpoint, center, spread, censoring behavior, and tail summaries.

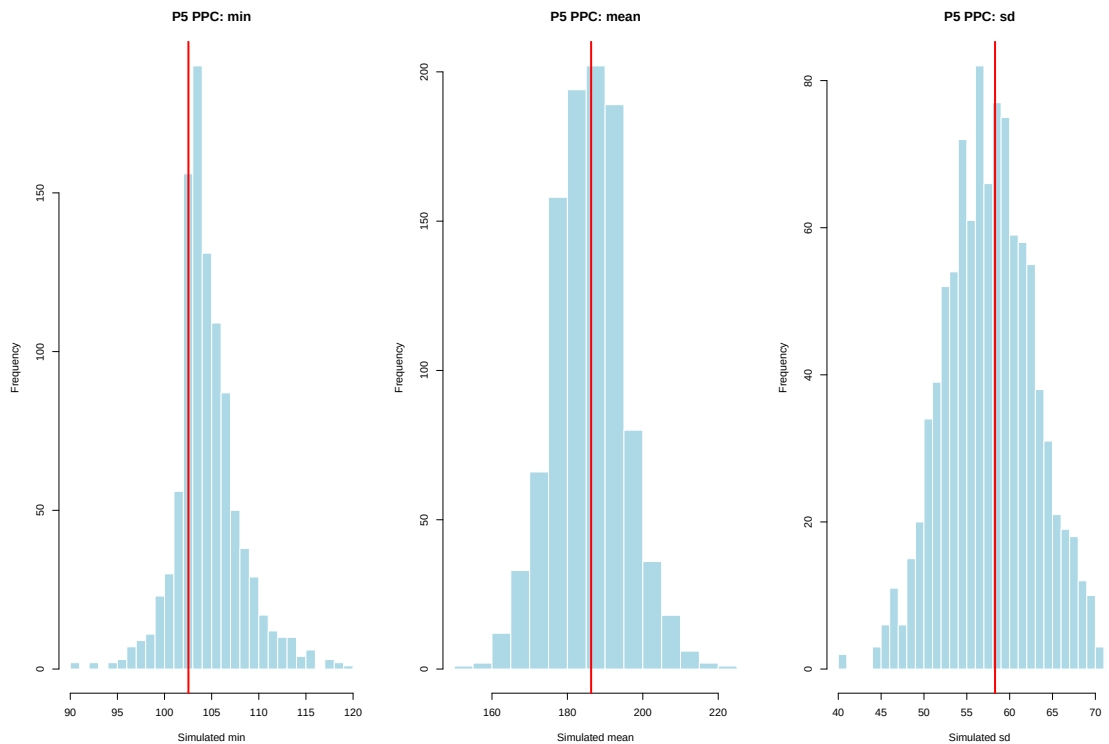


Figure 6.29: Representative posterior predictive checks for Population P5

6.8.8 Summary of the Semi-Simulated Alloy Fatigue Study

The semi-simulated alloy fatigue study shows how the Bayesian subset selection procedure behaves in a more realistic censored-data setting. Compared with the fully simulated exact-data study, the populations are more similar, the observations include right-censoring, and the posterior distributions of the threshold parameters show more overlap. As a result, the posterior evidence is less concentrated on a single population.

The posterior probability that P5 has the largest threshold parameter is 0.6702 under both samplers. Since this value is below the target level $P^* = 0.95$, P5 alone is not sufficient for the Bayesian selected subset. Adding P3 and P4 increases the cumulative posterior probability to 0.9942 under the random-walk sampler and 0.9949 under the independence sampler. Therefore, for $P^* = 0.95$, the Bayesian selected subset is

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

The same selected subset is obtained under both samplers, and the subset contains the true best population P5.

The percentile-based analysis gives a more nuanced comparison. For the 10th percentile, P5 has the largest posterior probability of being best. For the median and 90th percentile, the posterior probabilities are more dispersed, and P4 or P2 may appear more favorable. This difference occurs because Weibull percentiles depend on γ , η , and β together, while the formal best-population definition in this chapter is based only on γ .

The MCMC diagnostics, prior sensitivity analysis, and posterior predictive checks support the stability of the posterior inference and the adequacy of the fitted censored Weibull models. Overall, the semi-simulated study illustrates the practical role of Bayesian subset selection: when posterior uncertainty prevents selecting a single population with the desired confidence level, the procedure expands the selected subset to

preserve the target posterior probability. For the results of the semi-simulated study, use the R-code in Appendix B.20.

6.9 Results and Takeaways

This chapter developed a Bayesian subset selection procedure for comparing independent three-parameter Weibull populations. The best population was defined as the population with the largest threshold parameter γ , and posterior simulation was used to estimate the probability that each population was best. The Bayesian selected subset was then formed by ordering these posterior probabilities and including populations until the cumulative posterior probability reached the desired level P^* .

The fully simulated study provided a controlled setting in which the true parameter values were known. In this study, the posterior estimates preserved the true ordering of the threshold parameters, and P5 was correctly identified as the best population. The posterior probability that P5 had the largest threshold parameter was 1.0000 under both the random-walk Metropolis sampler and the independence Metropolis–Hastings sampler. Therefore, for $P^* = 0.95$, the Bayesian selected subset was

$$\mathcal{S}_B = \{\text{P5}\}.$$

The percentile-based comparisons gave the same practical conclusion, since P5 also had the largest posterior probability of having the largest 10th, 50th, and 90th Weibull percentiles. Thus, in the fully simulated exact-data study, the threshold-based and percentile-based reliability comparisons were strongly aligned.

The original Alloy T7987 fatigue-life data provided a real censored reliability-data baseline. Both maximum likelihood and Bayesian analyses supported the use of a three-parameter Weibull model as a reasonable working model for the original data. The

fitted shape parameter was greater than one, indicating an increasing hazard rate, which is consistent with fatigue-life behavior. The Bayesian posterior estimates from the original alloy analysis were then used as the baseline for constructing the semi-simulated censored study.

The semi-simulated alloy fatigue study showed how the Bayesian subset selection procedure behaves in a more realistic and more uncertain setting. The five populations were intentionally similar, and the data included right-censored observations. In this case, P5 still had the largest posterior probability of having the largest threshold parameter, but this probability was only 0.6702 under both samplers. Since this was below $P^* = 0.95$, P5 alone was not sufficient for the Bayesian selected subset. Adding P3 and P4 increased the cumulative posterior probability above 0.99, giving

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

This subset contains the true best population P5 while reflecting the additional posterior uncertainty caused by similar populations and right-censoring.

The percentile-based analyses provide an important supplementary reliability interpretation. In the fully simulated study, the percentile rankings were consistent with the threshold-based ranking. In the semi-simulated censored study, the percentile rankings were less aligned with the threshold ranking. For the 10th percentile, P5 had the largest posterior probability of being best, but for the median and 90th percentile, the posterior probabilities were more dispersed and other populations, such as P4 or P2, could appear more favorable. This difference occurs because Weibull percentiles depend on γ , η , and β together, while the formal subset selection rule in this chapter is based only on γ . Therefore, the percentile analysis should be viewed as a complementary reliability comparison rather than a replacement for the threshold-based selection rule.

Across both studies, the random-walk and independence Metropolis–Hastings samplers produced nearly identical posterior summaries, selection probabilities, and selected subsets. The trace plots showed stable behavior across chains, the Gelman–Rubin diagnostics were essentially equal to one, and the effective sample sizes were large for both Weibull parameters and percentiles. The independence sampler generally produced lower autocorrelation and larger effective sample sizes, indicating greater sampling efficiency, but the random-walk sampler also provided stable posterior inference.

The prior sensitivity analyses showed that posterior inference for the threshold component can be affected by the prior spread assigned to $\log(\delta)$, especially through the credible intervals for γ and δ . This is expected because the threshold parameter is constrained by the smallest observed lifetime and is recovered through the threshold-gap parameter. However, the main selection conclusions were stable under the prior choices considered. In the fully simulated study, P5 remained the unique selected population. In the semi-simulated alloy study, P5 remained the population with the strongest threshold-based posterior evidence, while the selected subset expanded to include P3 and P4 because of posterior uncertainty.

The posterior predictive checks supported the adequacy of the fitted three-parameter Weibull models in both studies. For the fully simulated data, the replicated datasets reproduced the main features of the observed generated samples. For the semi-simulated alloy fatigue data, the replicated censored datasets reproduced the lower endpoint, center, spread, censoring counts, and tail summaries reasonably well. The posterior predictive p -values were not close to 0 or 1, so the checks did not indicate substantial lack of fit.

The main takeaway is that Bayesian subset selection provides a direct posterior-probability approach for selecting a subset of reliability populations. When the

posterior evidence strongly identifies one population as best, the selected subset may contain a single population, as in the fully simulated study. When posterior uncertainty is larger, the selected subset expands to include additional plausible best populations, as in the semi-simulated censored alloy study. Thus, the Bayesian procedure adapts naturally to the amount of posterior uncertainty while preserving the desired posterior probability of containing the best population.

Several extensions are possible from the work developed in this dissertation. On the frequentist side, the subset selection rules could be extended to settings with unknown or unequal scale parameters, unequal sample sizes, or more general censoring mechanisms. Additional work could also develop analytic comparisons of the operating characteristics beyond the two-population case. On the Bayesian side, future work could consider hierarchical Weibull models, alternative prior structures for the threshold parameter, and decision-theoretic loss functions that explicitly balance the posterior probability of including the best population against the cost of selecting larger subsets. These extensions would broaden the applicability of threshold-based subset selection to more complex reliability studies.

CHAPTER 7

CONCLUSIONS

7.1 Summary of the Dissertation

This dissertation developed and compared subset selection procedures for reliability populations in which the best population is defined as the one with the largest threshold parameter. The threshold parameter represents a lower endpoint of the lifetime distribution under the assumed model, and therefore provides a meaningful reliability criterion when early failures are of particular concern. Instead of requiring a single best population to be selected in every case, the subset selection framework allows the procedure to return a set of promising populations while maintaining a prescribed probability requirement.

The dissertation considered both frequentist and Bayesian approaches. In the frequentist part, exponential threshold populations with a common known scale parameter were studied. Selection rules based on sample minima and sample means were developed and calibrated to include the best population with probability at least P^* . A Gamma-based version of the mean rule was also considered to provide exact finite-sample operating characteristics for the mean-based approach. In addition, an indifference-zone formulation was included as a related single-best selection framework.

In the Bayesian part, the methodology was extended to three-parameter Weibull populations. The Bayesian procedure used posterior simulation to estimate the probability that each population had the largest threshold parameter. The selected subset was then formed by ordering these posterior probabilities and including populations until the cumulative posterior probability reached the target level P^* . This approach provided a

post-data probability interpretation and allowed uncertainty in the Weibull parameters, including censoring, to be incorporated directly into the selection decision.

7.2 Main Findings

The frequentist results show that the minimum-based rule R_1 is well aligned with the threshold-based reliability objective. Since the sample minimum is directly connected to the lower endpoint of the exponential threshold model, R_1 responds strongly to differences in threshold parameters. The mean-based rule R_2 is also valid under the common-scale assumption because the population means are ordered in the same way as the threshold parameters. However, the operating-characteristic comparisons show that procedures calibrated to satisfy the same probability of correct selection requirement can behave very differently in terms of expected subset size.

Across the simulation-based operating-characteristic studies, the minimum-based rule R_1 generally produced smaller selected subsets than the mean-based rule R_2 once the best population became separated from the others. This pattern appeared under both slippage and equi-spaced configurations. The results show that both rules can satisfy the same PCS requirement, but R_1 tends to eliminate non-best populations more quickly. Thus, R_1 is usually more decisive in the exponential threshold setting, especially when the goal is to include the best population while avoiding unnecessarily large selected subsets.

The Gamma-based rule R_3 helped separate two issues: the use of sample means and the use of the CLT approximation. Since R_3 uses the exact Gamma distribution of the sample sums, it provides a more reliable finite-sample comparison for the mean-based procedure. The Gamma-based operating characteristics agreed closely with simulation results, supporting the accuracy of the simulation approach. In small samples, the Gamma

formulation also avoided some of the approximation issues observed for the CLT-based rule.

The exact two-population analysis clarified the mechanisms behind the numerical findings. For $k = 2$, the probability of correct selection, probability of incorrect selection, and expected subset size can be expressed as functions of the separation

$$\delta = \gamma_{[2]} - \gamma_{[1]}.$$

The exact formulas show that all three rules can be calibrated to attain the desired probability of correct selection at the least favorable configuration, where $\delta = 0$. The main practical difference is how quickly each rule removes the inferior population when $\delta > 0$. The results show that R_1 generally has a smaller probability of incorrect selection and therefore a smaller expected subset size than the mean-based rules over most of the configurations studied.

The Bayesian results show that posterior-probability-based subset selection provides a flexible alternative for Weibull reliability populations. In the fully generated Weibull study, the posterior evidence clearly identified population P5 as the best population according to the threshold parameter. The posterior probability that P5 had the largest threshold was 1.0000 under both samplers, so the Bayesian selected subset for $P^* = 0.95$ was

$$\mathcal{S}_B = \{P5\}.$$

In this controlled exact-data setting, the threshold-based selection and the percentile-based reliability comparisons were strongly aligned.

The semi-simulated Alloy T7987 fatigue-life study illustrated a more realistic and uncertain case. The populations were intentionally similar, and the data included right-censored observations. In this setting, P5 still had the largest posterior probability of

having the largest threshold parameter, but that probability was only 0.6702. Since this was below $P^* = 0.95$, selecting P5 alone was not sufficient. The Bayesian selected subset expanded to

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

which reflected the additional uncertainty caused by similar populations and censoring. This result illustrates a key advantage of Bayesian subset selection: the selected subset automatically expands when the posterior evidence is not concentrated on a single population.

The percentile-based analyses also showed that the threshold parameter and Weibull percentiles can lead to different reliability interpretations. In the fully generated study, the population with the largest threshold also tended to have the largest 10th, 50th, and 90th percentiles. In the semi-simulated censored study, the percentile rankings were less aligned with the threshold ranking. This is because Weibull percentiles depend jointly on γ , η , and β , while the formal subset selection rule in this dissertation is based only on γ . Therefore, percentile-based comparisons are useful supplementary summaries, but they should not be viewed as replacements for the threshold-based selection rule.

7.3 Practical Implications

The results of this dissertation have several practical implications for reliability-based selection problems. First, when the main concern is protection against early failures, the threshold parameter provides a natural definition of the best population. In that setting, rules based on sample minima are especially relevant because they directly reflect the lower-tail behavior of the distribution.

Second, subset selection is useful when selecting a single best population may be unstable. In many reliability settings, populations may be close together, sample sizes

may be limited, or data may be censored. A selected subset provides a more cautious decision by retaining plausible best populations while still reducing the set of candidates. This is especially important when an incorrect single-best decision could be costly.

Third, the frequentist and Bayesian procedures provide complementary interpretations. The frequentist rules are calibrated to satisfy repeated-sampling probability guarantees. They are useful when the goal is to design a procedure with long-run operating-characteristic performance. The Bayesian procedure, by contrast, provides a posterior probability statement conditional on the observed data and the assumed model. It is useful when the goal is to quantify current uncertainty about which population is best after observing the data.

Fourth, the results suggest that finite-sample distributional calculations should be used when available. The Gamma-based rule provides a useful alternative to the CLT-based mean rule, especially in small samples. More generally, exact or numerically stable finite-sample calculations can help avoid conclusions that are driven by approximation error rather than by the behavior of the selection rules themselves.

7.4 Limitations and Future Work

Several limitations of this dissertation suggest directions for future research. The frequentist procedures were developed for exponential threshold populations under a common known scale parameter. This assumption made it possible to compare threshold parameters directly and to calibrate the selection rules in a tractable way. In practice, scale parameters may be unknown or unequal across populations. Extending the frequentist subset selection rules to allow unknown or heterogeneous scale parameters would broaden the applicability of the methods.

The exact operating-characteristic analysis focused on the two-population case. This setting provided useful insight and allowed direct comparison of the minimum-based, CLT-based mean, and Gamma-based mean rules. A natural next step would be to develop additional analytic results for ($k > 2$), or to identify conditions under which formal ordering results for PCS, PICS, and ESS can be established.

The Bayesian analysis used independent Weibull models for the populations. Future work could consider hierarchical Bayesian models that allow information to be shared across populations. This may be especially useful when the populations represent related materials, systems, or treatment groups. Hierarchical modeling could improve estimation when sample sizes are small, although it would also require careful attention to prior specification and borrowing of information.

Another direction for future work is to consider alternative definitions of the best population. This dissertation defined the best population as the one with the largest threshold parameter. That choice is appropriate when early-failure protection is the primary goal. In other applications, however, the best population may be defined by a reliability percentile, survival probability at a specified time, mean lifetime, or another reliability functional. Extending the subset selection framework to these alternative criteria would allow the method to be tailored to a wider range of reliability objectives.

Finally, additional real-data applications would be useful for evaluating the practical performance of the proposed procedures. The Alloy T7987 study demonstrated how the Bayesian method can be applied in a censored fatigue-life setting, but further applications involving different censoring mechanisms, sample sizes, and Weibull parameter configurations would help assess the robustness and general usefulness of the proposed methods.

7.5 Final Remarks

Overall, this dissertation provides a threshold-based framework for subset selection in reliability populations. The frequentist results show how selection rules can be calibrated to satisfy probability guarantees and how their operating characteristics differ across parameter configurations. The exact two-population results explain why the minimum-based rule is often more decisive than mean-based alternatives in exponential threshold models. The Bayesian results extend the framework to three-parameter Weibull populations and show how posterior uncertainty can be incorporated into subset construction, especially in the presence of censoring.

The main conclusion is that subset selection provides a practical and interpretable approach for reliability problems in which the best population is defined through a threshold or lower-endpoint parameter. When the data clearly identify one best population, the selected subset can be small. When uncertainty remains, the subset expands to preserve the desired probability of containing the best population. This balance between probability protection and decision usefulness is the central contribution of the procedures studied in this dissertation.

APPENDIX A
PROOFS AND DERIVATIONS

A.1 Deriving the $\Pr(CS)$ $\Pr(ICS)$ for rule R_1 when $k = 2$

We have two 2-parameter Exponential distributions, with scale parameters $\eta_1 = \eta_2 = 1$, and threshold parameters $\gamma_{[1]} = 0$ and $\gamma_{[2]} = \delta \geq 0$, respectively.

Let

$$Y_i = \min_{1 \leq j \leq n} X_{ij}, \quad i = 1, 2, \dots, k. \quad (\text{A.1})$$

Then rule R_1 says:

$$R_1 : \text{ Select } \Pi_i \text{ iff } Y_i \geq \max_{1 \leq j \leq k} Y_j - d, \quad d \geq 0. \quad (\text{A.2})$$

The cumulative distribution function and the probability density function of Y_i are given as follows,

$$\Pr(Y_i \leq y) = 1 - e^{-n(y-\gamma_i)}, \quad y \geq \gamma_i \quad (\text{A.3})$$

$$f_{Y_i}(y) = ne^{-n(y-\gamma_i)}. \quad (\text{A.4})$$

Let $X_i \sim \exp(n)$ for $i = 1, 2$, be two independent identically distributed exponential random variables, with PDF

$$f_{X_i}(x) = \begin{cases} ne^{-nx}, & x \geq 0, \\ 0, & x < 0. \end{cases} \quad (\text{A.5})$$

Here $n > 0$ is the rate parameter of these distributions. Then, we can write $Y_i = \gamma_i + X_i$;

$$Y_{(1)} = \gamma_{[1]} + X_1$$

$$Y_{(2)} = \gamma_{[2]} + X_2$$

If X_1 and X_2 are independent and identically distributed exponential random variables with rate n , then $X_1 - X_2$ has a Laplace distribution with location 0 and scale $1/n$.

Theorem 1.1 *Let $X_1, X_2 \stackrel{iid}{\sim} \text{Exp}(n)$, and assume that X_1 and X_2 are independent. Then*

$$X_1 - X_2 \sim \text{Laplace}(0, 1/n),$$

with density

$$f_{X_1 - X_2}(x) = \begin{cases} \frac{n}{2} e^{-nx}, & x \geq 0, \\ \frac{n}{2} e^{nx}, & x < 0. \end{cases}$$

Proof. Since X_1 and X_2 are iid then, their joint PDF is given by the following,

$$f_{X_1 X_2}(x_1, x_2) = n e^{-nx_1} \cdot n e^{-nx_2} = n^2 e^{-n(x_1 + x_2)}, \quad x_1, x_2 \geq 0.$$

Let $U = X_1 - X_2$, and $V = X_1 + X_2$. From here we can see that $u \in \mathbb{R}$, and $v \geq 0$.

Then,

$$\begin{cases} U = X_1 - X_2, \\ V = X_1 + X_2 \end{cases} \implies \begin{cases} X_1 = \frac{U + V}{2} = h_1(u, v), \\ X_2 = \frac{V - U}{2} = h_2(u, v). \end{cases} \quad (\text{A.6})$$

$$X_1 = \frac{V + U}{2} \geq 0 \quad \text{and} \quad X_2 = \frac{V - U}{2} \geq 0 \quad \text{if and only if} \quad V \geq |U|.$$

Computing the Jacobian of the inverse transformation A.6 :

$$J = \frac{\partial(x_1, x_2)}{\partial(u, v)} = \begin{vmatrix} \frac{\partial x_1}{\partial u} & \frac{\partial x_1}{\partial v} \\ \frac{\partial x_2}{\partial u} & \frac{\partial x_2}{\partial v} \end{vmatrix} = \begin{vmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{vmatrix} = \frac{1}{2}.$$

Now, the joint PDF of the random variables U and V is given by,

$$\begin{aligned}
f_{U,V}(u,v) &= f_{X_1,X_2}(h_1(u,v), h_2(u,v)) \cdot |J| \\
&= n^2 e^{-n(\frac{u+v}{2} + \frac{v-u}{2})} \cdot \frac{1}{2} \\
&= \frac{n^2}{2} e^{-nv}, \quad v \geq |u|.
\end{aligned} \tag{A.7}$$

To find the marginal distribution of $U = X_1 - X_2$, we need to integrate out the random variable V from the joint PDF given by A.8.

$$\begin{aligned}
f_U(u) &= \int_{|u|}^{\infty} \frac{n^2}{2} e^{-nv} dv \\
&= \frac{n^2}{2} \left[-\frac{1}{n} e^{-nv} \right]_{|u|}^{\infty} \\
&= \frac{n^2}{2} \left[0 - \left(-\frac{1}{n} e^{-n|u|} \right) \right] \\
&= \frac{n}{2} e^{-n|u|}, \quad u \in \mathbb{R}.
\end{aligned} \tag{A.8}$$

This is the Laplace distribution with $b = \frac{1}{n}$, and $\mu = 0$.

$$f_Z(z) = \begin{cases} \frac{n}{2} e^{-nz}, & z \geq 0 \\ \frac{n}{2} e^{nz}, & z < 0. \end{cases} \tag{A.9}$$

□

The CDF of a Laplace random variable is found by integrating A.9.

For $z < 0$,

$$\begin{aligned}
 F_Z(z) &= \int_{-\infty}^z \frac{n}{2} e^{nt} dt \\
 &= \frac{n}{2} \left[\left(\frac{1}{n} \right) e^{nt} \right]_{-\infty}^z \\
 &= \frac{1}{2} e^{nz}
 \end{aligned} \tag{A.10}$$

For $z \geq 0$,

$$\begin{aligned}
 F_Z(z) &= F(z < 0) + \int_0^z \frac{n}{2} e^{nt} dt \\
 &= \int_{-\infty}^0 \frac{n}{2} e^{nz} dz + \frac{n}{2} \left[\left(-\frac{1}{n} \right) e^{nt} \right]_0^z \\
 &= \frac{1}{2} - \frac{1}{2} (e^{-nz} - e^0) \\
 &= 1 - \frac{1}{2} e^{-nz}
 \end{aligned} \tag{A.11}$$

Thus, the Laplace CDF when $b = \frac{1}{n}$ and $m = 0$, is given by:

$$F_Z(z) = \begin{cases} 1 - \frac{1}{2} e^{-nz}, & z \geq 0 \\ \frac{1}{2} e^{nz}, & z < 0 \end{cases} \tag{A.12}$$

Having the results above, now we can compute the $\Pr(CS_{R_1})$ and the $\Pr(ICS_{R_1})$ for rule R_1 .

$$\begin{aligned}
\Pr(CS_{R_1}) &= \Pr(Y_2 \geq Y_1 - d) \\
&= \Pr(\delta + X_2 \geq X_1 - d) \\
&= \Pr(X_1 - X_2 \leq \delta - d) \\
&= \Pr(Z \leq \delta + d), \quad \delta + d \geq 0
\end{aligned} \tag{A.13}$$

where Z is a Laplace random variable as in A.12. Then,

$$\Pr(CS_{R_1}) = 1 - \frac{1}{2}e^{-n(\delta+d)}.$$

Similarly, the probability of an incorrect selection $\Pr(ICS_{R_1})$,

$$\begin{aligned}
\Pr(ICS_{R_1}) &= \Pr(Y_1 \geq Y_2 - d) \\
&= \Pr(X_1 \geq X_2 + \delta - d) \\
&= \Pr(X_1 - X_2 \geq \delta - d) \\
&= \Pr(-(X_2 - X_1) \geq \delta - d) \\
&= \Pr(-Z \geq \delta - d) \\
&= \Pr(Z \leq d - \delta) \\
&= \begin{cases} \frac{1}{2}e^{-n(\delta-d)}, & d < \delta \\ 1 - \frac{1}{2}e^{-n(d-\delta)}, & d \geq \delta \end{cases}
\end{aligned}$$

A.2 Deriving Probability of Incorrect Selection $\Pr(ICS)$ for Rule R_3 when $k = 2$

Under the slippage configuration $(0, \delta)$, population Π_2 is the best population and population Π_1 is the non-best population. For rule R_3 , the selection rule is

$$S_i \geq \max(S_1, S_2) - nb,$$

where

$$S_i = \sum_{j=1}^n X_{ij}.$$

The event of incorrect selection is the event that the non-best population Π_1 is included in the selected subset. For $k = 2$, this event is

$$S_1 \geq S_2 - nb.$$

Under the slippage configuration $(0, \delta)$,

$$S_1 = G_1, \quad S_2 = n\delta + G_2,$$

where G_1 and G_2 are independent $\text{Gamma}(n, 1)$ random variables with common density $g(\cdot)$ and cdf $G(\cdot)$. Therefore,

$$\begin{aligned} \Pr(ICS_3) &= \Pr(S_1 \geq S_2 - nb) \\ &= \Pr(G_1 \geq n\delta + G_2 - nb) \\ &= \Pr\{G_1 \geq G_2 + n(\delta - b)\}. \end{aligned} \tag{A.14}$$

The form of this probability depends on the sign of $\delta - b$.

A.2.1 Case 1: $b \geq \delta$.

If $b \geq \delta$, define

$$h = n(b - \delta) \geq 0.$$

Then

$$n(\delta - b) = -h,$$

and the event in (A.14) becomes

$$G_1 \geq G_2 - h.$$

Equivalently,

$$G_2 \leq G_1 + h.$$

Conditioning on $G_1 = x$, we obtain

$$\Pr(G_2 \leq G_1 + h \mid G_1 = x) = \Pr(G_2 \leq x + h) = G(x + h).$$

Averaging over the density of G_1 gives

$$\begin{aligned} \Pr(\text{ICS}_3) &= \int_0^\infty G(x + h)g(x) dx \\ &= \int_0^\infty G(x + n(b - \delta))g(x) dx. \end{aligned} \tag{A.15}$$

Thus, when $b \geq \delta$,

$$\Pr(\text{ICS}_3) = \int_0^\infty G(x + n(b - \delta))g(x) dx.$$

A.2.2 Case 2: $\delta \geq b$.

If $\delta \geq b$, define

$$h = n(\delta - b) \geq 0.$$

Then the event in (A.14) becomes

$$G_1 \geq G_2 + h.$$

Conditioning on $G_2 = x$, we obtain

$$\Pr(G_1 \geq G_2 + h \mid G_2 = x) = \Pr(G_1 \geq x + h).$$

Since G_1 has cdf $G(\cdot)$,

$$\Pr(G_1 \geq x + h) = 1 - G(x + h).$$

Averaging over the density of G_2 gives

$$\begin{aligned} \Pr(\text{ICS}_3) &= \int_0^\infty \{1 - G(x + h)\} g(x) dx \\ &= \int_0^\infty \{1 - G(x + n(\delta - b))\} g(x) dx. \end{aligned} \quad (\text{A.16})$$

Thus, when $\delta \geq b$,

$$\Pr(\text{ICS}_3) = \int_0^\infty \{1 - G(x + n(\delta - b))\} g(x) dx.$$

Combining (A.15) and (A.16), the incorrect-selection probability for rule R_3 when $k = 2$ is

$$\Pr(\text{ICS}_3) = \begin{cases} \int_0^\infty G(x + n(b - \delta)) g(x) dx, & b \geq \delta, \\ \int_0^\infty \{1 - G(x + n(\delta - b))\} g(x) dx, & \delta \geq b. \end{cases} \quad (\text{A.17})$$

At $\delta = b$, both expressions reduce to the same probability because G_1 and G_2 are independent and identically distributed continuous random variables. In particular,

$$\Pr(G_1 \geq G_2) = \Pr(G_2 \geq G_1) = \frac{1}{2}.$$

Therefore, the two cases agree at the boundary $\delta = b$.

APPENDIX B
SOURCE CODE FOR CONSTANTS AND SIMULATIONS

B.1 Source Code for Constant d

```
#calculating the constant d. exp.sim.10

#set for k = 10 populations; n samples; N simulations
k=10;n=25;N=200000

set.seed(17) #repeat same numbers for each program simulation
Y1<-rep(0,N);Y2<-rep(0,N);Y3<-rep(0,N);Y4<-rep(0,N)
Y5<-rep(0,N);Y6<-rep(0,N);Y7<-rep(0,N);Y8<-rep(0,N)
Y9<-rep(0,N);Y10<-rep(0,N);S<-rep(0,N)

for (i in 1:N){
  X1<-rexp(n);X2<-rexp(n);X3<-rexp(n);X4<-rexp(n);X5<-rexp(n)
  X6<-rexp(n);X7<-rexp(n);X8<-rexp(n);X9<-rexp(n);X10<-rexp(n)
  Y1[i]<-min(X1);Y2[i]<-min(X2);Y3[i]<-min(X3);Y4[i]<-min(X4)
  Y5[i]<-min(X5);Y6[i]<-min(X6);Y7[i]<-min(X7);Y8[i]<-min(X8)
  Y9[i]<-min(X9);Y10[i]<-min(X10)
  S[i]<-max(Y1[i],Y2[i],Y3[i],Y4[i],Y5[i],Y6[i],
           Y7[i],Y8[i],Y9[i],Y10[i])-Y10[i]
}

length(S)

summary(S)

df<-data.frame(Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10,S)

head(df,10)

SS<-round(sort(S),4)

head(SS,20)

CUSUM<-cumsum(SS)

df1<-data.frame(SS,CUSUM)

tail(df1,20)
```

```

df2<-data.frame(table(SS))
#df2
df3<-data.frame(df2,cumsum(df2$Freq))
df3
Freq<-df3$Freq
CDF<-df3$cumsum.df2.Freq./N
df4<-data.frame(df2,CDF)
#tail(df4,200)
plot(df4$SS,df4$CDF)
quant<-c(0.75,0.90,0.95,0.975,0.990)
quantile(SS,quant)

```

B.2 Source Code for Constant b

```

#exp.sim.10
#set for k = 10 populations; n samples; N simulations
k=10;n=25;N=200000
set.seed(17) #repeat same numbers for each program simulation
Y1<-rep(0,N);Y2<-rep(0,N);Y3<-rep(0,N);Y4<-rep(0,N)
Y5<-rep(0,N);Y6<-rep(0,N);Y7<-rep(0,N);Y8<-rep(0,N)
Y9<-rep(0,N);Y10<-rep(0,N);S<-rep(0,N)
for (i in 1:N){
  X1<-rexp(n);X2<-rexp(n);X3<-rexp(n);X4<-rexp(n);X5<-rexp(n)
  X6<-rexp(n);X7<-rexp(n);X8<-rexp(n);X9<-rexp(n);X10<-rexp(n)
  Y1[i]<-mean(X1);Y2[i]<-mean(X2);Y3[i]<-mean(X3);Y4[i]<-mean(X4)
  Y5[i]<-mean(X5);Y6[i]<-mean(X6);Y7[i]<-mean(X7);Y8[i]<-mean(X8)
  Y9[i]<-mean(X9);Y10[i]<-mean(X10)
}

```

```

S[i]<-max(Y1[i],Y2[i],Y3[i],Y4[i],Y5[i],Y6[i],
          Y7[i],Y8[i],Y9[i],Y10[i])-Y10[i]
}
length(S)
summary(S)
df<-data.frame(Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10,S)
head(df,10)
SS<-round(sort(S),4)
head(SS,20)
CUSUM<-cumsum(SS)
df1<-data.frame(SS,CUSUM)
tail(df1,20)
df2<-data.frame(table(SS))
#df2
df3<-data.frame(df2,cumsum(df2$Freq))
df3
Freq<-df3$Freq
CDF<-df3$cumsum.df2.Freq./N
df4<-data.frame(df2,CDF)
#tail(df4,200)
plot(df4$SS,df4$CDF)
quant<-c(0.75,0.90,0.95,0.975,0.990)
round(quantile(SS,quant),4)

```

B.3 Source Code for the Sample Statistics and the Selected Populations

```
#exp.sim.subset
```

```

#Subset selection for exponential distributions
#differencing in threshold parameters
#11_20_2024
#k populations with samples of size n
#d and b must be determined for a given  $P^*=P$ 
rm(list=ls())
#Input the value of  $P^*$  as P along with k and n
k<-10; n<-25; P<-0.95
if (P==0.75){d<-0.1088;b<-0.4816}
if (P==0.90){d<-0.1500;b<-0.6274}
if (P==0.95){d<-0.1791;b<-0.7144}
if (P==0.975){d<-0.2074;b<-0.7906}
if (P==0.99){d<-0.2432;b<-0.8810}
set.seed(17) #insure same simulated values on repeat
#gamma<-seq(from = 0, to = 1.8, by = 0.2)
#gamma <-c(rep(0,9), 1)
delta <- 0.2
gamma <- (0:9)*delta
M<-matrix(0,nrow=n,ncol=k)
for (i in 1:k){
  M[,i]<-rexp(n) + gamma[i]
}
M
#####
y<-rep(0,k)
for (i in 1:k){

```

```

    y[i]<-min(M[,i])
  }
  print(y)
  max.y<-max(y)
  s<-rep(0,k)
  for (i in 1:k){
    if (y[i]>=max.y-d){s[i]<-1}
  }
  print(y-max.y)
  print(s)
  #####
  z<-rep(0,k)
  for (i in 1:k){
    z[i]<-mean(M[,i])
  }
  print(z)
  max.z<-max(z)
  t<-rep(0,k)
  for (i in 1:k){
    if (z[i]>=max.z-b){t[i]<-1}
  }
  print(z-max.z)
  print(t)
  #####
  w<-seq(1:k)
  df<-data.frame(w,y,z)

```

```

colnames(df)<-c("populations","minimums","means")
round(df,4)
#####
df1<-data.frame(s,t)
colnames(df1)<-c("minsel","meansel")
message("k = ",k," n = ",n," P* = ",P)
print('The selected populations denoted by 1')
df1

```

B.4 Source code for Indifference Zone Example

```

# P(CS) expon.pops.given.d
# P* with given d for k exponential populations
# Selection based on min order statistics with n samples
n<-12
k<-5
d<-0.25
exp.sub <- function(x){
  n*((1 - exp(-n*(x+d)))^(k-1))*exp(-(n*x))}
ans <- integrate(exp.sub, lower = 0, upper = Inf)
ans
ans$value

```

B.5 Source code for OCs under Slippage Configuration

```

#Matrix_mins_SC
#Simulation of exponential distributions matrix format
#Selection Rule R1(mins) with slippage configuration

```



```

#02/06/2025

rm(list=ls())

set.seed(17)

#Input values of P* as P;number of simulations N

#k=number of pops;n=sample size per pop

#Use delta = 0 to obtain d-values for R1 given as quantiles

#at end of program enter them in the 5 if statements below

k<-10;n<-25;delta<-0;P<-0.95

N<-200000; T<-rep(0,k);W<-rep(-1, N); mini<-rep(-1, k)

mini.exp<-rep(0,k);x<-rep(-1, n*k)

M<-matrix(x, ncol=k, nrow=n)

if(P==0.75){d<-0.1088}

if(P==0.90){d<-0.1500}

if(P==0.95){d<-0.1791}

if(P==0.975){d<-0.2074}

if(P==0.99){d<-0.2432}

for (j in 1:N){

  for (i in 1:k){M[,i]<-rexp(n,rate = 1)}

  M

  for (i in 1:k){mini.exp[i]<- min(M[,i])}

  mini.exp

  S<-rep(0,k)

  M[,k] <- M[,k] + delta

for (i in 1:k){mini[i]<-min(M[,i])}

  mini.max <- max(mini)

  diff<- mini.max-mini

```

```

W[j]<-diff[k]
for (i in 1:k){
  if (diff[i]<= d){S[i]<-1}
}
T<-T+S
}
message("k=",k,"n=",n,"delta=",delta,"P* =",P,"d=",d,"N=", N)
CS<-T[k]/N
CS<-round(CS,4)
ICS<-sum(T[1:k-1])/N
ICS<-round(ICS,4)
ESS<-sum(T)/N
ESS<-round(ESS,4)
message("ICS=", ICS,"Pr(CS) =", CS,"ESS=", ESS)
#For use with delta = 0 to determine d-values
length(W)
quan<-c(0.75,0.90,0.95,0.975, 0.99)
round(quantile(W, quan),4)
#Matrix_means_SC
#Simulation of exponential distributions matrix format
#Selection Rule R2(means) with slippage configuration (SC)
#02/06/2025
rm(list=ls())
set.seed(17)
#Input values P* as P; number of simulations as N
#k=number of pops; n=sample size per pop

```

```

#Use delta = 0 to obtain b-values for R2 given as quantiles
#at end of rprogram and enter them in the 5 statements below
k<-10;n<-25;delta<-0;P<-0.95
N<-200000; T<-rep(0,k);W<-rep(-1, N);avg<-rep(-1,k)
mean.exp<-rep(0,k);x<-rep(-1, n*k)
M<-matrix(x, ncol=k, nrow=n)
if(P==0.75){b<-0.48159}
if(P==0.90){b<-0.62742}
if(P==0.95){b<-0.71436}
if(P==0.975){b<-0.79057}
if(P==0.99){b<-0.88100}
for(j in 1:N){
  for(i in 1:k){M[,i]<-rexp(n,rate=1)}
  M
  for(i in 1:k){mean.exp[i]<-mean(M[,i])}
  mean.exp
  S<-rep(0,k)
  M[,k]<-M[,k]+delta
  for(i in 1:k){avg[i]<-mean(M[,i])}
  avg.max <- max(avg)
  diff<-avg.max-avg
W[j]<-diff[k]
for(i in 1:k){
  if (diff[i]<= b){S[i] <- 1}
}
T <- T + S

```

```

}

message("k=", k, ",n=", n, ",delta =", delta, ",P* =",P, ",b=",b,"N=", N)

CS<-T[k]/N

CS<-round(CS,4)

ICS<-sum(T[1:k-1])/N

ICS<-round(ICS,4)

ESS<-sum(T)/N

ESS<-round(ESS,4)

message("ICS=", ICS, ",Pr(CS) =", CS,"ESS =", ESS)

#For use with delta = 0 to determine b-values

length(W)

quan<-c(0.75, 0.90, 0.95, 0.975, 0.99)

round(quantile(W,quan), 5)

```

B.6 Source code for OCs under Equi-Spaced Configuration

```

#Matrix_mins_ES

#Simulation of exponential distribution matrix format

#Selection Rule R1 (mins) with equi-spaced configuration

#02_09_2025

rm(list=ls())

set.seed(17)

#Input values of P* as P;number of simulations as N

#k=number of pops; n=sample size per pop

#Use delta = 0 to obtain d-values for R1 given as quantiles

k<-10;n<-25;delta<-0;P<-0.95

N<-200000; T<-rep(0,k);W<-rep(-1, N);mini<-rep(-1,k)

```

```

min.exp<-rep(0,k);x<-rep(-1, n*k)
M<-matrix(x, ncol=k, nrow=n)
#Enter proper b-values for k,n,P*, delta = 0
if(P==0.75){d<-0.1088}
if(P==0.90){d<-0.1500}
if(P==0.95){d<-0.1791}
if(P==0.975){d<-0.2074}
if(P==0.99){d<-0.2432}
for (j in 1:N){
  for (i in 1:k){M[,i] <- rexp(n,rate = 1)}
  M
  for (i in 1:k){min.exp[i]<-min(M[,i])}
  min.exp
  S<-rep(0,k)
  for (i in 1:k){M[,i] <- M[,i] +(i-1)*delta}
  for (i in 1:k){mini[i] <- min(M[,i])}
  mini.max <- max(mini)
  diff<- mini.max - mini
  W[j] <- diff[k]
  for (i in 1:k){
    if (diff[i]<=d){S[i]<-1}
  }
  T<-T+S
}
round(T/N,4)
ESS <- sum(T)/N

```

```

round(ESS,4)

#For use with delta = 0 to determine d-values
length(W)

quan<-c(0.75, 0.90, 0.95, 0.975, 0.99)
round(quantile(W,quan), 4)

#Matrix_means_ES

#Simulation of exponential distribution matrix format
#Selection Rule R2 (means) with equi-spaced configuration
#02_09_2025

rm(list=ls())

set.seed(17)

#Input values of P* as P;number of simulations as N
#k=number of pops; n=sample size per pop
#Use delta = 0 to obtain b-values for R2 given as quantiles
k<-10;n<-25;delta<-0;P<-0.95
N<-200000; T<-rep(0,k);W<-rep(-1, N);avg<-rep(-1,k)
mean.exp<-rep(0,k);x<-rep(-1, n*k)
M<-matrix(x, ncol=k, nrow=n)

#Enter proper b-values for k,n,P*, delta = 0
if(P==0.75){b<-0.4816}
if(P==0.90){b<-0.6274}
if(P==0.95){b<-0.7144}
if(P==0.975){b<-0.7906}
if(P==0.99){b<-0.8810}
for (j in 1:N){
  for (i in 1:k){M[,i] <- rexp(n,rate = 1)}

```

```

M
for (i in 1:k){mean.exp[i]<-mean(M[,i])}
mean.exp
S<-rep(0,k)
for (i in 1:k){M[,i] <- M[,i] +(i-1)*delta}
for (i in 1:k){avg[i] <- mean(M[,i])}
avg.max <- max(avg)
diff<- avg.max - avg
W[j] <- diff[k]
for (i in 1:k){
  if (diff[i]<=b){S[i]<-1}
}
T<-T+S
}
round(T/N,4)
ESS <- sum(T)/N
round(ESS,4)
#For use with delta = 0 to determine b-values
length(W)
quan<-c(0.75, 0.90, 0.95, 0.975, 0.99)
round(quantile(W,quan), 4)

```

B.7 Source Code for Expected Subset Size Plots

```

#plot of ESS for R1 and R2 slippage configuration
#01_08_2025
#k=25; n=10; P*=0.95

```

```

#set.seed(17)

rm(list=ls())

delta<-c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0)

ESS.R2<-c(9.4935,9.4577,9.3272,9.0515,8.5702,7.8288,6.8172,5.6201,
4.3822,3.2658, 2.3806)

ESS.R1<-c(9.5022,9.0118,3.6245,1.2189,1.0180,1.0013,1.0001,1.0000,
1.0000,1.0000, 1.0000)

df<-data.frame(delta,ESS.R1,ESS.R2)

df

data=matrix(c(ESS.R1,ESS.R2),ncol=11,byrow=TRUE)

colnames(data)=c('0','0.1','0.2','0.3','0.4','0.5','0.6','0.7',
'0.8','0.9','1.0')

rownames(data)=c('ESS.mins','ESS.means')

#data

final=as.table(data)

final

barplot(final,beside=TRUE,col=c("red","blue"),xlab="delta",
ylab="Expected Subset Size",
      main="Slippage Configuration",ylim=c(0,10))

legend("right",box.col="brown",bg="yellow",legend=c("mins(R1)",
"means(R2)"),fill=c("red","blue"))

#plot of ESS for R1 and R2 equi-spaced configuration

#01_02_202

#k=25; n=10; P*=0.95

#set.seed(17)

```



```

rm(list=ls())

delta<-c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0)

ESS.R2<-c(9.4935,6.5053,3.8512,2.8169,2.2692,1.9281,1.7006,1.5272,
1.3870,1.2522,1.1529)

ESS.R1<-c(9.5022,2.2448,1.2987,1.0244,1.0020,1.0001,1.0000,1.0000,
1.0000,1.0000,1.0000)

df<-data.frame(delta,ESS.R1,ESS.R2)

df

data=matrix(c(ESS.R1,ESS.R2),ncol=11,byrow=TRUE)

colnames(data)=c('0','0.1','0.2','0.3','0.4','0.5','0.6','0.7',
'0.8','0.9','1.0')

rownames(data)=c('ESS.mins','ESS.means')

#data

final=as.table(data)

final

barplot(final,beside=TRUE,col=c("red","blue"),xlab="delta",
ylab="Expected Subset Size",
        main="Equi-Spaced Configuration",ylim=c(0,10))

legend("right",box.col="brown",bg="yellow",legend=c("mins(R1)",
"means(R2)"),fill=c("red","blue"))

```

B.8 Source Code for CLT Approach of R2

```

#P(CS) for k populations with slippage configuration

#Assuming sample means are normally distributed: CLT

rm(list=ls())

#input model values

```

```

#see "Selection Rules for Exponential Population Threshold
#Parameters, Sections 2.2 and 2.3, Means Rule R2
#Input value of P* as P
k<-10; delta<-0; n<-25; P<-0.95
if (P==0.75){b<-0.4528}
if (P==0.90){b<-0.5970}
if (P==0.95){b<-0.6836}
if (P==0.975){b<-0.7598}
if (P==0.99){b<-0.8500}
c<-(sqrt(n))*(b+delta)
int<-function(x){
  ((pnorm(x+c))^(k-1))*dnorm(x)
}
PCS<-integrate(int,lower = -Inf,upper = Inf)
message("k = ",k," n = ",n," delta = ",delta," P* = ",P,
        ", b = ",b)
PCS
round(PCS$value,4)
u<-(sqrt(n))*b
v<-(sqrt(n))*(b-delta)
int1<-function(x){
  ((pnorm(x+u))^(k-2))*(pnorm(x+v))*dnorm(x)
}
P1<-integrate(int1,lower = -Inf,upper = Inf)
P1
round(P1$value,4)

```

```

#PICS is the probability of an incorrect selection
PICS<-(k-1)*P1$value
round(PICS,4)

#ESS is the expected subset size
ESS<-PCS$value+(k-1)*P1$value
message("The expected subset size is ",round(ESS,4))

```

B.9 Source Code for Gamma Distribution Approach of R2

```

#gamma.int.k
#OCs for R3 using gamma distribution for sum of exp rv's
#input k, n, b, and delta
rm(list=ls())
k=10; n=25; b=0.71559; delta=0.1
fun1<-function(x){
  ((pgamma(x+n*b+n*delta,shape=n,scale=1))^(k-1))*dgamma(x,shape=n,scale=1)
}
PrCS<-integrate(fun1,lower=0,upper=Inf)
PrCS
fun2<-function(x){
  ((pgamma(x+n*b,shape=n,scale=1))^(k-2))*
  pgamma(x+n*b-n*delta,shape=n,scale=1)*
  dgamma(x,shape=n,scale=1)
}
#Pr1 = Pr(choosing pop1) =, ..., = Pr(choosing pop(k-1))
Pr1<-integrate(fun2,lower=0,upper=Inf)
Pr1

```

```

PrICS<-(k-1)*Pr1$value
ESS<-PrCS$value+PrICS
df<-data.frame(delta,PrCS$value,PrICS,ESS)
df<-round(df,4)
df

```

B.10 Source Code for OCs for 5 Populations and Small Sample Size, Slippage

Configuration

```

#Matrix_mins_SC
#Simulation of exponential distributions matrix format
#Selection Rule R1 (mins) with slippage configuration
rm(list=ls())
set.seed(17)

#Input values of P* as P; number of simulations as N
#k=number of pops;n=sample size per pop
#Use delta=0 to obtain d-values for R1 given as quantiles
#at end of program and enter them in the 5 if statements below
k<-5;n<-10; delta<-0; P<-0.95
N<-200000; T<-rep(0,k); W<-rep(-1,N); mini<-rep(-1,k)
mini.exp<-rep(0,k); x<-rep(-1,n*k)
M<-matrix(x,ncol=k,nrow=n)

if (P==0.75){d<-0.1933}
if (P==0.90){d<-0.2943}
if (P==0.95){d<-0.3652}
if (P==0.975){d<-0.4367}
if (P==0.99){d<-0.5301}

```

```

for (j in 1:N){
  for (i in 1:k){M[,i]<-rexp(n,rate=1)}
  M
  for (i in 1:k){mini.exp[i]<-min(M[,i])}
  mini.exp
  S<-rep(0,k)
  M[,k]<-M[,k]+delta
  for (i in 1:k){mini[i]<-min(M[,i])}
  mini.max<-max(mini)
  diff<-mini.max-mini
  W[j]<-diff[k]
  for (i in 1:k){
    if (diff[i]<=d){S[i]<-1}
  }
  T<-T+S
}
message("k= ",k," n= ",n," delta= ",delta," P*= ",P,
        " d= ",d," N= ",N)
CS<-T[k]/N
CS<-round(CS,4)
ICS<-sum(T[1:k-1])/N
ICS<-round(ICS,4)
ESS<-sum(T)/N
ESS<-round(ESS,4)
message("ICS =",ICS," ,Pr(CS) =",CS," ,ESS =",ESS)
#For use with delta=0 to determine d-values

```

```

length(W)

quan<-c(0.75,0.90,0.95,0.975,0.99)

round(quantile(W,quan),5)


#Matrix_means_SC

#Simulation of exponential distributions matrix format

#Selection Rule R2 (means) with slippage configuration (SC)

rm(list=ls())

set.seed(17)

#Input values of P* as P; number of simulations as N

#k=number of pops;n=sample size per pop

#Use delta=0 to obtain b-values for R2 given as quantiles

#at end of program and enter them in the 5 if statements below

k<-5;n<-10; delta<-0; P<-0.95

N<-200000; T<-rep(0,k); W<-rep(-1,N); avg<-rep(-1,k)

mean.exp<-rep(0,k); x<-rep(-1,n*k)

M<-matrix(x,ncol=k,nrow=n)

if (P==0.75){b<-0.6148}

if (P==0.90){b<-0.8636}

if (P==0.95){b<-1.0160}

if (P==0.975){b<-1.1559}

if (P==0.99){b<-1.3240}

for (j in 1:N){

  for (i in 1:k){M[,i]<-rexp(n,rate=1)}

  M

  for (i in 1:k){mean.exp[i]<-mean(M[,i])}

```

```

mean.exp
S<-rep(0,k)
M[,k]<-M[,k]+delta
for (i in 1:k){avg[i]<-mean(M[,i])}
avg.max<-max(avg)
diff<-avg.max-avg
W[j]<-diff[k]
for (i in 1:k){
  if (diff[i]<=b){S[i]<-1}
}
T<-T+S
}
message("k= ",k," n= ",n," delta= ",delta," P*= ",P,
        " b= ",b," N= ",N)
CS<-T[k]/N
CS<-round(CS,4)
ICS<-sum(T[1:k-1])/N
ICS<-round(ICS,4)
ESS<-sum(T)/N
ESS<-round(ESS,4)
message("ICS =",ICS," ,Pr(CS) =",CS," ,ESS =",ESS)
#For use with delta=0 to determine b-values
length(W)
quan<-c(0.75,0.90,0.95,0.975,0.99)
round(quantile(W,quan),5)

```

B.11 Source Code for OCs for 5 Populations and Small Sample Size, Equi-Spaced

Configuration

```
#Simulation of exponential distributions matrix format
#Selection Rule R1 (mins) with equi-spaced configuration
#01_08_2025

rm(list=ls())

set.seed(17)

#Input values of P* as P; number of simulations as N
k<-5;n<-10; delta<-0.5; P<-0.95

N<-200000; T<-rep(0,k); W<-rep(-1,N); mini<-rep(-1,k)

min.exp<-rep(0,k); x<-rep(-1,n*k)

M<-matrix(x,ncol=k,nrow=n)

if (P==0.75){d<-0.1933}
if (P==0.90){d<-0.2943}
if (P==0.95){d<-0.3652}
if (P==0.975){d<-0.4367}
if (P==0.99){d<-0.5301}

for (j in 1:N){
  for (i in 1:k){M[,i]<-rexp(n,rate=1)}
  M
  for (i in 1:k){min.exp[i]<-min(M[,i])}
  min.exp
  S<-rep(0,k)
  for (i in 1:k){M[,i]<-M[,i]+(i-1)*delta}
  for (i in 1:k){mini[i]<-min(M[,i])}
  mini.max<-max(mini)
```



```

diff<-mini.max-mini
W[j]<-diff[k]
for (i in 1:k){
  if (diff[i]<=d){S[i]<-1}
}
T<-T+S
}
round(T/N,4)
ESS<-sum(T)/N
round(ESS,4)
#For use with delta=0 to determine d-values
length(W)
quan<-c(0.75,0.90,0.95,0.975,0.99)
round(quantile(W,quan),4)

#Simulation of exponential distributions matrix format
#Selection Rule R2 (means) with equi-spaced configuration
#01_16_2025
rm(list=ls())
set.seed(17)

#Input values of P* as P; number of simulations as N
k<-5;n<-10; delta<-0.5; P<-0.95
N<-200000; T<-rep(0,k); W<-rep(-1,N); avg<-rep(-1,k)
mean.exp<-rep(0,k); x<-rep(-1,n*k)
M<-matrix(x,ncol=k,nrow=n)
if (P==0.75){b<-0.6148}

```

```

if (P==0.90){b<-0.8636}
if (P==0.95){b<-1.0160}
if (P==0.75){b<-1.1559}
if (P==0.99){b<-1.3240}
for (j in 1:N){
  for (i in 1:k){M[,i]<-rexp(n,rate=1)}
  M
  for (i in 1:k){mean.exp[i]<-mean(M[,i])}
  mean.exp
  S<-rep(0,k)
  for (i in 1:k){M[,i]<-M[,i]+(i-1)*delta}
  for (i in 1:k){avg[i]<-mean(M[,i])}
  avg.max<-max(avg)
  diff<-avg.max-avg
  W[j]<-diff[k]
  for (i in 1:k){
    if (diff[i]<=b){S[i]<-1}
  }
  T<-T+S
}
round(T/N,4)
ESS<-sum(T)/N
round(ESS,4)
#For use with delta=0 to determine b-values
length(W)
quan<-c(0.75,0.90,0.95,0.975,0.99)

```

```
round(quantile(W,quan),4)
```

B.12 Source Code for OCs for 2 Populations and Various Sample Sizes

```
#OCs for R1 and R2, k=2
#R1 values end in 1; R2 end in 2
#Set n, delta, and P
n<-25; delta<-c(0,0.05,0.15,0.20,0.25,0.30,0.35,0.40,0.45,0.50,0.55); P<-0.95
len<-length(delta)
PCS1<-rep(0,len);PCS2<-rep(0,len);PICS1<-rep(0,len)
PICS2<-rep(0,len)
d<-log(2*(1-P))/n
b<-sqrt(2/n)*qnorm(P)
params<-c(d,b)
params
for (i in 1:len){
  PCS1[i]<-1-0.5*exp(-n*delta[i]-n*d)
  PCS2[i]<-pnorm((b+delta[i])/sqrt(2/n))
  if(delta[i]>=d){PICS1[i]<-(0.5)*exp(-n*(delta[i]-d))}
  if(d>delta[i]){PICS1[i]<-1-0.5*exp(-n*(d-delta[i]))}
  PICS2[i]<-pnorm((b-delta[i])/sqrt(2/n))
}
ESS1<-PCS1+PICS1
ESS2<-PCS2+PICS2
df1<-data.frame(n,d,b,P,delta,PCS1,PCS2,PICS1,PICS2,ESS1,ESS2)
df1<-round(df1,5)
df1
```

```
length(delta)
length(PCS1)
```

B.13 Source Code for Exploring the Differences of OCs of Rule 1 and Rule 2, when

$$\underline{k = 2}$$

```
# OCs for k = 2, n = 5, 25, 100, P = 0.95, 0.99
# b calculated using CLT approximation
# 05/27/2026
rm(list = ls())
k <- 2
n <- 5
P <- 0.95
b <- sqrt(2/n) * qnorm(P)
del <- seq(from = 0, to = 2, by = 0.1)
len.d <- length(del)
d <- -(log(2 * (1 - P))) / n
PCS1 <- matrix(rep(0, 2 * len.d),
nrow = len.d, ncol = 2, byrow = TRUE)
PICS1 <- matrix(rep(0, 2 * len.d),
nrow = len.d, ncol = 2, byrow = TRUE)
PCS2 <- matrix(rep(0, 2 * len.d),
nrow = len.d, ncol = 2, byrow = TRUE)
PICS2 <- matrix(rep(0, 2 * len.d),
nrow = len.d, ncol = 2, byrow = TRUE)
diff.CS <- matrix(rep(0, 2 * len.d),
nrow = len.d, ncol = 2, byrow = TRUE)
```

```

diff.ICS <- matrix(rep(0, 2 * len.d),
nrow = len.d, ncol = 2, byrow = TRUE)
for (i in 1:len.d) {
  PCS1[i, 2] <- 1 - 0.5 * exp(-n * (d + del[i]))
  if (del[i] >= d) {
    PICS1[i, 2] <- 0.5 * exp(-n * (del[i] - d))
  }
  if (d > del[i]) {
    PICS1[i, 2] <- 1 - 0.5 * exp(-n * (d - del[i]))
  }
}
df1 <- data.frame(del = del, PCS1 = PCS1[, 2],
PICS1 = PICS1[, 2])
for (i in 1:len.d) {
  PCS2[i, 2] <- pnorm((b + del[i]) / sqrt(2/n))
  PICS2[i, 2] <- pnorm((b - del[i]) / sqrt(2/n))
}
df2 <- data.frame(del = del, PCS1 = PCS1[, 2],
PICS1 = PICS1[, 2], PCS2 = PCS2[, 2], PICS2 = PICS2[, 2])
diff.CS <- PCS1 - PCS2
diff.ICS <- PICS1 - PICS2
df3 <- data.frame(del = del, diff.CS = diff.CS[, 2],
diff.ICS = diff.ICS[, 2])
ESS1 <- PCS1[, 2] + PICS1[, 2]
ESS2 <- PCS2[, 2] + PICS2[, 2]
diff.ESS <- ESS1 - ESS2

```

```

df4 <- data.frame(del = del, ESS1 = ESS1, ESS2 = ESS2,
diff.ESS = diff.ESS)

results <- data.frame(del = del, PCS1 = PCS1[, 2],
PCS2 = PCS2[, 2], PICS1 = PICS1[, 2], PICS2 = PICS2[, 2],
ESS1 = ESS1, ESS2 = ESS2)

results$diff_PCS <- results$PCS1 - results$PCS2
results$diff_PICS <- results$PICS1 - results$PICS2
results$diff_ESS <- results$ESS1 - results$ESS2

tol <- 1e-10

results$PCS_ineq_holds <- results$diff_PCS >= -tol
results$PICS_ineq_holds <- results$diff_PICS <= tol
results$ESS_ineq_holds <- results$diff_ESS <= tol

check_table <- results[, c("del", "diff_PCS", "PCS_ineq_holds",
"diff_PICS", "PICS_ineq_holds", "diff_ESS", "ESS_ineq_holds")]

violations <- subset(check_table, PCS_ineq_holds == FALSE |
PICS_ineq_holds == FALSE | ESS_ineq_holds == FALSE)

summary_all <- data.frame(Case = "All delta values",
Min_diff_PCS = min(results$diff_PCS),
Max_diff_PICS = max(results$diff_PICS),
Max_diff_ESS = max(results$diff_ESS))

summary_positive_delta <- data.frame(Case = "Only delta > 0",
Min_diff_PCS = min(results$diff_PCS[results$del > 0]),
Max_diff_PICS = max(results$diff_PICS[results$del > 0]),
Max_diff_ESS = max(results$diff_ESS[results$del > 0]))

summary_ineq <- rbind(summary_all, summary_positive_delta)

oc_outputs <- list(constants = data.frame(k = k, n = n,

```

```

P = P, d = d, b = b),
rule1 = df1, rule1_rule2 = df2,
differences = df3,
expected_subset_size = df4,
results = results, check_table = check_table,
violations = violations, summary_ineq = summary_ineq)

```

B.14 Source Code for Exploring the Differences of OCs of Rule 1 and Rule 3, when $k = 2$

```

# OCs for k = 2, n = 5, 25, 100, P = 0.95, 0.99
# Rule 3: b calculated from Gamma-based rule
# 05/27/2026
rm(list = ls())
k <- 2
n <- 5
P <- 0.95
b_gamma <- function(k, n, P, tol = 1e-10) {
  pcs_at_b <- function(b) {
    integrand <- function(x) {
      pgamma(x + n*b, shape = n, scale = 1)^(k - 1)
      * dgamma(x, shape = n, scale = 1)
    }
    integrate(integrand, lower = 0,
      upper = Inf, rel.tol = 1e-10)$value
  }
  equation <- function(b) {
    pcs_at_b(b) - P
  }
}

```

```

}

lower_b <- 0

upper_b <- 1

while (equation(upper_b) < 0) {
  upper_b <- 2 * upper_b
}

b_hat <- uniroot(equation,
  lower = lower_b, upper = upper_b, tol = tol)$root
return(b_hat)
}

b <- b_gamma(k = k, n = n, P = P)

del <- seq(from = 0, to = 2, by = 0.1)

len.d <- length(del)

d <- -(log(2*(1 - P))) / n

PCS1 <- matrix(0, nrow = len.d, ncol = 2)
PICS1 <- matrix(0, nrow = len.d, ncol = 2)
PCS3 <- matrix(0, nrow = len.d, ncol = 2)
PICS3 <- matrix(0, nrow = len.d, ncol = 2)
diff.CS <- matrix(0, nrow = len.d, ncol = 2)
diff.ICS <- matrix(0, nrow = len.d, ncol = 2)

for (i in 1:len.d) {
  PCS1[i, 2] <- 1 - 0.5 * exp(-n * (d + del[i]))
  if (del[i] >= d) {
    PICS1[i, 2] <- 0.5 * exp(-n * (del[i] - d))
  }
  if (d > del[i]) {

```



```

    PICS1[i, 2] <- 1 - 0.5 * exp(-n * (d - del[i]))
  }
}

df1 <- data.frame(del = del, PCS1 = PCS1[, 2],
PICS1 = PICS1[, 2])
for (i in 1:len.d) {
  fun_pcs3 <- function(x) {
    pgamma(x + n*(b + del[i]), shape = n, scale = 1)^(k - 1)
    * dgamma(x, shape = n, scale = 1)
  }
  int_pcs3 <- integrate(fun_pcs3, lower = 0,
upper = Inf, rel.tol = 1e-10)
  PCS3[i, 2] <- int_pcs3$value
}

for (i in 1:len.d) {
  fun_pics3 <- function(x) {
    pgamma(x + n*b, shape = n, scale = 1)^(k - 2)
    * pgamma(x + n*(b - del[i]), shape = n, scale = 1)
    * dgamma(x, shape = n, scale = 1)
  }
  int_pics3 <- integrate(fun_pics3, lower = 0,
upper = Inf, rel.tol = 1e-10)
  PICS3[i, 2] <- int_pics3$value
}

df2 <- data.frame(del = del, PCS1 = PCS1[, 2],
PICS1 = PICS1[, 2], PCS3 = PCS3[, 2], PICS3 = PICS3[, 2])

```

```

diff.CS <- PCS1 - PCS3
diff.ICS <- PICS1 - PICS3
df3 <- data.frame(del = del, diff.CS = diff.CS[, 2],
diff.ICS = diff.ICS[, 2])
ESS1 <- PCS1[, 2] + PICS1[, 2]
ESS3 <- PCS3[, 2] + PICS3[, 2]
diff.ESS <- ESS1 - ESS3
df4 <- data.frame(del = del, ESS1 = ESS1,
ESS3 = ESS3, diff.ESS = diff.ESS)
results <- data.frame(del = del, PCS1 = PCS1[, 2],
PCS3 = PCS3[, 2], PICS1 = PICS1[, 2],
PICS3 = PICS3[, 2], ESS1 = ESS1, ESS3 = ESS3)
results$diff_PCS <- results$PCS1 - results$PCS3
results$diff_PICS <- results$PICS1 - results$PICS3
results$diff_ESS <- results$ESS1 - results$ESS3
tol <- 1e-10
results$PCS_ineq_holds <- results$diff_PCS >= -tol
results$PICS_ineq_holds <- results$diff_PICS <= tol
results$ESS_ineq_holds <- results$diff_ESS <= tol
check_table <- results[, c("del", "diff_PCS",
"PCS_ineq_holds", "diff_PICS",
"PICS_ineq_holds", "diff_ESS", "ESS_ineq_holds")]
violations <- subset(check_table,
PCS_ineq_holds == FALSE | PICS_ineq_holds == FALSE |
ESS_ineq_holds == FALSE)
summary_all <- data.frame(Case = "All delta values",

```

```

Min_diff_PCS = min(results$diff_PCS),
Max_diff_PICS = max(results$diff_PICS),
Max_diff_ESS = max(results$diff_ESS))
summary_positive_delta <- data.frame(Case = "Only delta > 0",
Min_diff_PCS = min(results$diff_PCS[results$del > 0]),
Max_diff_PICS = max(results$diff_PICS[results$del > 0]),
Max_diff_ESS = max(results$diff_ESS[results$del > 0]))
summary_ineq <- rbind(summary_all, summary_positive_delta)
oc_outputs <- list(constants = data.frame(k = k, n = n,
P = P, d = d, b = b), rule1 = df1, rule1_rule3 = df2,
differences = df3, expected_subset_size = df4,
results = results, check_table = check_table,
violations = violations, summary_ineq = summary_ineq)

```

B.15 Source Code for Heat Map of Differences of OCs of Rule 1 and Rule 2, when $k = 2$

```

rm(list = ls())
P <- 0.99
n_grid <- seq(5, 100, by = 5)
delta_grid <- seq(0, 0.5, by = 0.01)
heat_clt <- expand.grid(n = n_grid,
delta = delta_grid)
heat_clt$PCS1 <- NA
heat_clt$PICS1 <- NA
heat_clt$PCS2 <- NA
heat_clt$PICS2 <- NA
heat_clt$ESS1 <- NA

```

```

heat_clt$ESS2 <- NA
heat_clt$diff_PCS <- NA
heat_clt$diff_PICS <- NA
heat_clt$diff_ESS <- NA
for (r in 1:nrow(heat_clt)) {
  n <- heat_clt$n[r]
  delta <- heat_clt$delta[r]
  d <- -(log(2 * (1 - P))) / n
  b <- sqrt(2 / n) * qnorm(P)
  PCS1 <- 1 - 0.5 * exp(-n * (d + delta))
  if (delta >= d) {
    PICS1 <- 0.5 * exp(-n * (delta - d))
  } else {
    PICS1 <- 1 - 0.5 * exp(-n * (d - delta))
  }
  PCS2 <- pnorm((b + delta) / sqrt(2 / n))
  PICS2 <- pnorm((b - delta) / sqrt(2 / n))
  ESS1 <- PCS1 + PICS1
  ESS2 <- PCS2 + PICS2
  heat_clt$PCS1[r] <- PCS1
  heat_clt$PICS1[r] <- PICS1
  heat_clt$PCS2[r] <- PCS2
  heat_clt$PICS2[r] <- PICS2
  heat_clt$ESS1[r] <- ESS1
  heat_clt$ESS2[r] <- ESS2
  heat_clt$diff_PCS[r] <- PCS1 - PCS2

```

```

    heat_clt$diff_PICS[r] <- PICS1 - PICS2
    heat_clt$diff_ESS[r] <- ESS1 - ESS2
  }

  tol <- 1e-10

  PCS_holds <- all(heat_clt$diff_PCS >= -tol)
  PICS_holds <- all(heat_clt$diff_PICS <= tol)
  ESS_holds <- all(heat_clt$diff_ESS <= tol)

  inequality_summary <- data.frame(Inequality = c("P(CS1) >= P(CS2)",
    "P(ICS1) <= P(ICS2)", "ESS1 <= ESS2"),
  Holds_over_grid = c(PCS_holds, PICS_holds, ESS_holds))

  violations_heat_clt <- subset(heat_clt,
    diff_PCS < -tol | diff_PICS > tol | diff_ESS > tol)

  heat_clt_outputs <- list(results = heat_clt,
    inequality_summary = inequality_summary,
    violations = violations_heat_clt)

```

B.16 Source Code for Heat Map of Differences of OCs of Rule 1 and Rule 3, when $k = 2$

```

rm(list = ls())

k <- 2
P <- 0.99

n_grid <- seq(5, 100, by = 5)
delta_grid <- seq(0, 0.5, by = 0.01)

b_gamma <- function(k, n, P, tol = 1e-10) {
  pcs_at_b <- function(b) {
    integrand <- function(x) {
      pgamma(x + n*b, shape = n, scale = 1)^(k - 1)

```

```

      * dgamma(x, shape = n, scale = 1)
    }

    integrate(integrand, lower = 0, upper = Inf,
      rel.tol = 1e-10)$value
  }

equation <- function(b) {
  pcs_at_b(b) - P
}

lower_b <- 0
upper_b <- 1
while (equation(upper_b) < 0) {
  upper_b <- 2 * upper_b
}

b_hat <- uniroot(equation, lower = lower_b,
  upper = upper_b, tol = tol)$root
return(b_hat)
}

b_table <- data.frame(n = n_grid, b = NA)
for (j in 1:nrow(b_table)) {
  b_table$b[j] <- b_gamma(k = k, n = b_table$n[j], P = P)
}

heat_gamma <- expand.grid(n = n_grid, delta = delta_grid)
heat_gamma$b <- b_table$b[match(heat_gamma$n, b_table$n)]
heat_gamma$PCS1 <- NA
heat_gamma$PICS1 <- NA
heat_gamma$PCS3 <- NA

```

```

heat_gamma$PICS3 <- NA
heat_gamma$ESS1 <- NA
heat_gamma$ESS3 <- NA
heat_gamma$diff_PCS <- NA
heat_gamma$diff_PICS <- NA
heat_gamma$diff_ESS <- NA
for (r in 1:nrow(heat_gamma)) {
  n <- heat_gamma$n[r]
  delta <- heat_gamma$delta[r]
  b <- heat_gamma$b[r]
  d <- -(log(2 * (1 - P))) / n
  PCS1 <- 1 - 0.5 * exp(-n * (d + delta))
  if (delta >= d) {
    PICS1 <- 0.5 * exp(-n * (delta - d))
  } else {
    PICS1 <- 1 - 0.5 * exp(-n * (d - delta))
  }
  fun_pcs3 <- function(x) {
    pgamma(x + n*(b + delta), shape = n, scale = 1)^(k - 1)
    * dgamma(x, shape = n, scale = 1)
  }
  PCS3 <- integrate(fun_pcs3, lower = 0,
    upper = Inf, rel.tol = 1e-10)$value
  fun_pics3 <- function(x) {
    pgamma(x + n*b, shape = n, scale = 1)^(k - 2) *
    pgamma(x + n*(b - delta), shape = n, scale = 1) *

```

```

    dgamma(x, shape = n, scale = 1)
  }
  PICS3 <- integrate(fun_pics3, lower = 0,
    upper = Inf, rel.tol = 1e-10)$value
  ESS1 <- PCS1 + PICS1
  ESS3 <- PCS3 + PICS3
  heat_gamma$PCS1[r] <- PCS1
  heat_gamma$PICS1[r] <- PICS1
  heat_gamma$PCS3[r] <- PCS3
  heat_gamma$PICS3[r] <- PICS3
  heat_gamma$ESS1[r] <- ESS1
  heat_gamma$ESS3[r] <- ESS3
  heat_gamma$diff_PCS[r] <- PCS1 - PCS3
  heat_gamma$diff_PICS[r] <- PICS1 - PICS3
  heat_gamma$diff_ESS[r] <- ESS1 - ESS3
}

tol <- 1e-10

PCS_holds <- all(heat_gamma$diff_PCS >= -tol)
PICS_holds <- all(heat_gamma$diff_PICS <= tol)
ESS_holds <- all(heat_gamma$diff_ESS <= tol)

inequality_summary <- data.frame(Inequality = c("P(CS1) >= P(CS3)",
"P(ICS1) <= P(ICS3)", "ESS1 <= ESS3"),
  Holds_over_grid = c(PCS_holds, PICS_holds, ESS_holds))

violations_heat_gamma <- subset(heat_gamma,
  diff_PCS < -tol | diff_PICS > tol | diff_ESS > tol)

heat_gamma_outputs <- list(b_table = b_table,

```



```
results = heat_gamma,  
inequality_summary = inequality_summary,  
violations = violations_heat_gamma)
```

B.17 Source Code for the Fully Simulated Weibull Study

```
library(LearnBayes)  
library(doParallel)  
library(foreach)  
library(MASS)  
library(dplyr)  
library(coda)  
set.seed(17)  
PAD <- 1e-6  
n_chains <- 4  
iters_long <- 200000  
burn_in <- 20000  
thin <- 3  
rw_scale_main <- 1.2  
indep_inflate <- 4  
use_mixture_indep <- TRUE  
mixture_df <- 5  
mixture_p <- 0.20  
main_a_beta <- 6  
main_b_beta <- 0.25  
main_a_eta <- 20  
main_b_eta <- 0.05
```

```

main_theta3_mean <- log(0.05)
main_theta3_sd <- 1.0
theta3_sd_grid <- c(0.5, 1.0, 1.5)
perc <- c(0.10, 0.50, 0.90)
z_p <- -log(1 - perc)
ppc_nrep <- 1000
n_per_pop <- 150
true_gamma <- c(5.0, 5.3, 5.6, 5.9, 6.2)
true_eta <- c(0.9, 0.95, 1.0, 1.05, 1.1)
true_beta <- c(1.2, 1.35, 1.5, 1.65, 1.8)
pop_names <- paste0("P", seq_along(true_gamma))
K <- length(pop_names)
datasets <- lapply(seq_len(K), function(i) {
  true_gamma[i] + stats::rweibull(
    n_per_pop,
    shape = true_beta[i],
    scale = true_eta[i]
  )
})
names(datasets) <- pop_names
true_params <- data.frame(
  Population = pop_names,
  True_Gamma = true_gamma,
  True_Eta = true_eta,
  True_Beta = true_beta,
  check.names = FALSE

```

```

)

true_percentiles <- data.frame(
  Population = pop_names,
  True_Gamma = true_gamma,
  True_Eta = true_eta,
  True_Beta = true_beta,
  t10 = true_gamma + true_eta *
    ((-log(1 - 0.10))^(1 / true_beta)),
  t50 = true_gamma + true_eta *
    ((-log(1 - 0.50))^(1 / true_beta)),
  t90 = true_gamma + true_eta *
    ((-log(1 - 0.90))^(1 / true_beta)),
  check.names = FALSE
) |>

mutate(
  across(where(is.numeric), ~ round(.x, 4))
)

logpost_gamma <- 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])
t1   <- min(data)
delta <- exp(theta[3])
gamma <- t1 - delta - pad
if (!is.finite(beta) || !is.finite(eta)
|| beta <= 0 || eta <= 0) {
  return(-Inf)
}
if (any(data <= gamma)) return(-Inf)
y <- data - gamma
ll <- sum(
  log(beta) - beta * log(eta) +
  (beta - 1) * log(y) -
  (y / eta)^beta
)
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]
lp_theta3 <- dnorm(theta[3], mean = theta3_mean,
sd = theta3_sd, log = TRUE)
ll + lp_beta + lp_eta + lp_theta3
}
make_logpost_gamma <- 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_gamma(
    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
  )
}
}

get_start <- function(x, epsilon = PAD) {
  t1 <- min(x)
  deltas <- exp(seq(
    log(1e-5),
    log(max(0.8 * sd(x), 1e-3)),
    length.out = 25
  ))
})

```

```

best <- NULL
best_ll <- -Inf
for (dlt in deltas) {
  g <- t1 - dlt - PAD
  y <- x - g
  if (any(y <= 0)) next
  fit <- try(
    suppressWarnings(MASS::fitdistr(y, "weibull")),
    silent = TRUE
  )
  if (inherits(fit, "try-error")) next
  beta0 <- fit$estimate["shape"]
  eta0 <- fit$estimate["scale"]
  ll <- sum(
    log(beta0) - beta0 * log(eta0) +
    (beta0 - 1) * log(y) -
    (y / eta0)^beta0
  )
  if (is.finite(ll) && ll > best_ll) {
    best_ll <- ll
    best <- c(beta0, eta0, dlt)
  }
}
if (is.null(best)) {
  g <- t1 * 0.95
  y <- x - g

```

```

wb <- suppressWarnings(MASS::fitdistr(y, "weibull"))$estimate
best <- c(wb["shape"], wb["scale"], max(t1 - g - epsilon, 1e-4))
}

c(log(best[1]), log(best[2]), log(best[3]))
}

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))
  }
  out <- LearnBayes::rwmetrop(
    logpost,
    list(var = fitlap$var, scale = scale),
    fitlap$mode,
    iters,
    data
  )
  list(
    mode = colMeans(out$par),
    var = cov(out$par)
  )
}

```

```

check_laplace <- function(fitlap, label = "") {
  cond_num <- tryCatch(kappa(fitlap$var),
    error = function(e) Inf)
  chol_ok <- tryCatch({chol(fitlap$var); TRUE},
    error = function(e) FALSE)
  all(is.finite(fitlap$var)) && is.finite(cond_num)
  && cond_num <= 1e6 && chol_ok
}

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))
}

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) * log1p(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_, iters, length(th))
  acc <- 0L
  for (t in 1: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
)
}

cl <- makeCluster(max(1, parallel::detectCores() - 1))
registerDoParallel(cl)
export_fns <- c(
  "logpost_gamma", "make_logpost_gamma", "get_start",
  "rw_pilot", "check_laplace", "regularize_covariance",
  "dnorm_mv", "dt_mv", "rmvt1", "rprop_mix1",
  "dprop_mix1", "indepmetrop_mix", "n_chains",
  "iters_long", "burn_in", "thin", "rw_scale_main",

```

```

    "indep_inflate", "use_mixture_indep", "mixture_df",
    "mixture_p", "PAD", "main_a_beta", "main_b_beta",
    "main_a_eta", "main_b_eta", "main_theta3_mean",
    "main_theta3_sd"
  )
results <- foreach(
  i = 1:K,
  .packages = c("LearnBayes", "MASS"),
  .export = export_fns
) %dopar% {
  data_i <- datasets[[i]]
  logpost_i <- make_logpost_gamma(theta3_sd = main_theta3_sd)
  start0 <- get_start(data_i)
  chain_results <- lapply(1:n_chains, function(ch) {
    start_ch <- if (ch == 1) {
      start0
    } else {
      start0 + rnorm(3, 0, 0.3)
    }
    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))
    )
    pilot$var <- regularize_covariance(pilot$var)
    r.walk <- LearnBayes::rwmetrop(
      logpost_i,

```

```

    list(var = pilot$var, scale = rw_scale_main),
    pilot$mode,
    iters_long,
    data_i
  )
  fitlap <- tryCatch(
LearnBayes::laplace(logpost_i, pilot$mode, data_i),
error = function(e) list(mode = pilot$mode, var = pilot$var)
  )
  fitlap$var <- regularize_covariance(fitlap$var)
  laplace_ok <- check_laplace(
    fitlap,
    paste0("Population ", i, ", chain ", ch)
  )
  Sigma_prop <- regularize_covariance(
    if (laplace_ok) fitlap$var else pilot$var
  )
  mu_prop <- if (laplace_ok) fitlap$mode else pilot$mode
  if (!use_mixture_indep) {
    in.walk <- LearnBayes::indepmetrop(
      logpost_i,
      list(var = Sigma_prop * indep_inflate, mu = mu_prop),
      mu_prop,
      iters_long,
      data_i
    )
  }

```

```

    acc_in <- if (!is.null(in.walk$accept)) {
      in.walk$accept
    } else {
mean(apply(diff(in.walk$par), 1, function(x) any(x != 0)))
    }
  } else {
    prop <- list(
      mu = as.numeric(mu_prop),
      var = Sigma_prop * indep_inflate,
      df = mixture_df,
      mix_p = mixture_p
    )
    in.walk <- indepmetrop_mix(
      logpost_i,
      mu_prop,
      data_i,
      iters_long,
      prop
    )
    acc_in <- in.walk$acc
  }
trim_thin <- function(m) {
  m <- m[-(1:burn_in), , drop = FALSE]
  m[seq(1, nrow(m), by = thin), , drop = FALSE]
}
list(

```

```

    rw = trim_thin(r.walk$par),
    indep = trim_thin(in.walk$par),
    acc_rw = if (!is.null(r.walk$accept)) {
      r.walk$accept
    } else {
      mean(apply(diff(r.walk$par), 1, function(x) any(x != 0)))
    },
    acc_in = acc_in
  )
})
list(
  chains = chain_results,
  data = data_i
)
}

stopCluster(cl)

get_rhat_for_sampler <- function(i, sampler = "rw") {
  mclist <- coda::mcmc.list(
    lapply(results[[i]]$chains, function(ch) {
      coda::mcmc(ch[[sampler]])
    })
  )

  rhat <- tryCatch(
    coda::gelman.diag(
      mclist,
      multivariate = FALSE,

```

```

    autoburnin = FALSE
  )$psrf,
  error = function(e) {
    matrix(
      NA_real_,
      nrow = 3,
      ncol = 2,
      dimnames = list(
        c("log_beta", "log_eta", "log_delta"),
        c("Point est.", "Upper C.I.")
      )
    )
  }
)

rownames(rhat) <- c("log_beta", "log_eta", "log_delta")
rhat
}

rhat_rw_results <- lapply(1:K, function(i)
  get_rhat_for_sampler(i, sampler = "rw"))
rhat_in_results <- lapply(1:K, function(i)
  get_rhat_for_sampler(i, sampler = "indep"))
names(rhat_rw_results) <- pop_names
names(rhat_in_results) <- pop_names
pool_chains <- function(results_i, sampler = "rw") {
  do.call(rbind, lapply(results_i$chains, function(ch) {
    if (sampler == "rw") ch$rw else ch$indep
  }

```



```

    )))
  }
  recover_params <- function(draws_mat, data_i) {
    t1 <- min(data_i)
    list(
      beta = exp(draws_mat[, 1]),
      eta  = exp(draws_mat[, 2]),
      gamma = t1 - exp(draws_mat[, 3]) - PAD
    )
  }
  params_rw <- lapply(1:K, function(i) {
    recover_params(pool_chains(results[[i]], "rw"),
      datasets[[i]])
  })
  params_in <- lapply(1:K, function(i) {
    recover_params(pool_chains(results[[i]], "indep"),
      datasets[[i]])
  })
  assemble_mat <- function(params_list, param_name) {
    do.call(cbind, lapply(params_list, '[' + param_name))
  }
  g.rw.s <- assemble_mat(params_rw, "gamma")
  g.in.s <- assemble_mat(params_in, "gamma")
  eta.rw.s <- assemble_mat(params_rw, "eta")
  eta.in.s <- assemble_mat(params_in, "eta")
  beta.rw.s <- assemble_mat(params_rw, "beta")

```

```

beta.in.s <- assemble_mat(params_in, "beta")

colnames(g.rw.s) <- pop_names
colnames(g.in.s) <- pop_names
colnames(eta.rw.s) <- pop_names
colnames(eta.in.s) <- pop_names
colnames(beta.rw.s) <- pop_names
colnames(beta.in.s) <- pop_names

N_eff <- nrow(g.rw.s)

tp_from_draws <- function(g_mat, eta_mat, beta_mat, j) {
  g_mat + eta_mat * (z_p[j]^(1 / beta_mat))
}

tp_rw_list <- lapply(seq_along(perc), function(j) {
  tp_from_draws(g.rw.s, eta.rw.s, beta.rw.s, j)
})

tp_in_list <- lapply(seq_along(perc), function(j) {
  tp_from_draws(g.in.s, eta.in.s, beta.in.s, j)
})

names(tp_rw_list) <- names(tp_in_list) <-
paste0("t_", perc * 100, "%")

post_means <- bind_rows(
  data.frame(
    Parameter = "Gamma",
    Population = pop_names,
    'Random Walk Mean' = round(colMeans(g.rw.s), 4),
    'Independent Walk Mean' = round(colMeans(g.in.s), 4),
    check.names = FALSE

```

```

),
data.frame(
  Parameter = "Eta",
  Population = pop_names,
  'Random Walk Mean' = round(colMeans(eta.rw.s), 4),
  'Independent Walk Mean' = round(colMeans(eta.in.s), 4),
  check.names = FALSE
),
data.frame(
  Parameter = "Beta",
  Population = pop_names,
  'Random Walk Mean' = round(colMeans(beta.rw.s), 4),
  'Independent Walk Mean' = round(colMeans(beta.in.s), 4),
  check.names = FALSE
)
) |>
  arrange(match(Parameter, c("Gamma", "Eta", "Beta")))
make_ci_rows <- function(mat_rw, mat_in, criterion) {
  bind_rows(lapply(1:K, function(i) {
    data.frame(
      Criterion = criterion,
      Population = pop_names[i],
      'Post Mean (RW)' = round(mean(mat_rw[, i]), 4),
      '2.5% (RW)' = round(quantile(mat_rw[, i], 0.025), 4),
      '97.5% (RW)' = round(quantile(mat_rw[, i], 0.975), 4),
      'Post Mean (In)' = round(mean(mat_in[, i]), 4),

```

```

    '2.5% (In)' = round(quantile(mat_in[, i], 0.025), 4),
    '97.5% (In)' = round(quantile(mat_in[, i], 0.975), 4),
    check.names = FALSE
  )
}))
}

ci_table <- bind_rows(
  make_ci_rows(g.rw.s, g.in.s, "Gamma"),
  make_ci_rows(eta.rw.s, eta.in.s, "Eta"),
  make_ci_rows(beta.rw.s, beta.in.s, "Beta"),
  bind_rows(lapply(seq_along(perc), function(j) {
    make_ci_rows(
      tp_rw_list[[j]],
      tp_in_list[[j]],
      paste0("t_", perc[j] * 100, "%")
    )
  })))
)

tp_means_tbl <- bind_rows(lapply(seq_along(perc), function(j) {
  data.frame(
    Percentile = paste0(perc[j] * 100, "%"),
    Population = pop_names,
    'Random Walk Mean' = round(colMeans(tp_rw_list[[j]]), 4),
    'Independent Walk Mean' = round(colMeans(tp_in_list[[j]]), 4),
    check.names = FALSE
  )
}))

```

```

}))

posterior_medians <- bind_rows(
  data.frame(
    Parameter = "Gamma",
    Population = pop_names,
    'Random Walk Median' = round(apply(g.rw.s, 2, median), 4),
    'Independent Walk Median' = round(apply(g.in.s, 2, median), 4),
    check.names = FALSE
  ),
  data.frame(
    Parameter = "Eta",
    Population = pop_names,
    'Random Walk Median' = round(apply(eta.rw.s, 2, median), 4),
    'Independent Walk Median' = round(apply(eta.in.s, 2, median), 4),
    check.names = FALSE
  ),
  data.frame(
    Parameter = "Beta",
    Population = pop_names,
    'Random Walk Median' = round(apply(beta.rw.s, 2, median), 4),
    'Independent Walk Median' = round(apply(beta.in.s, 2, median), 4),
    check.names = FALSE
  )
) |>
  arrange(match(Parameter, c("Gamma", "Eta", "Beta")))

truth_comparison <- bind_rows(

```

```

data.frame(
  Parameter = "Gamma",
  Population = pop_names,
  True = true_gamma,
  'RW Mean' = colMeans(g.rw.s),
  'RW Median' = apply(g.rw.s, 2, median),
  'Indep Mean' = colMeans(g.in.s),
  'Indep Median' = apply(g.in.s, 2, median),
  check.names = FALSE
),
data.frame(
  Parameter = "Eta",
  Population = pop_names,
  True = true_eta,
  'RW Mean' = colMeans(eta.rw.s),
  'RW Median' = apply(eta.rw.s, 2, median),
  'Indep Mean' = colMeans(eta.in.s),
  'Indep Median' = apply(eta.in.s, 2, median),
  check.names = FALSE
),
data.frame(
  Parameter = "Beta",
  Population = pop_names,
  True = true_beta,
  'RW Mean' = colMeans(beta.rw.s),
  'RW Median' = apply(beta.rw.s, 2, median),

```

```

    'Indep Mean' = colMeans(beta.in.s),
    'Indep Median' = apply(beta.in.s, 2, median),
    check.names = FALSE
  )
) |>

mutate(
  'RW Mean Error' = 'RW Mean' - True,
  'RW Median Error' = 'RW Median' - True,
  'Indep Mean Error' = 'Indep Mean' - True,
  'Indep Median Error' = 'Indep Median' - True,
  across(where(is.numeric), ~ round(.x, 4))
) |>

arrange(match(Parameter, c("Gamma", "Eta", "Beta")), Population)

sel_prob <- function(mat) {
  idx <- apply(mat, 1, which.max)
  round(tabulate(idx, nbins = K) / nrow(mat), 4)
}

selection_probs <- bind_rows(
  data.frame(
    Parameter = "Gamma",
    Population = pop_names,
    'Random Walk P' = sel_prob(g.rw.s),
    'Independent Walk P' = sel_prob(g.in.s),
    check.names = FALSE
  ),
  data.frame(

```

```

    Parameter = "Eta",
    Population = pop_names,
    'Random Walk P' = sel_prob(eta.rw.s),
    'Independent Walk P' = sel_prob(eta.in.s),
    check.names = FALSE
  ),
  data.frame(
    Parameter = "Beta",
    Population = pop_names,
    'Random Walk P' = sel_prob(beta.rw.s),
    'Independent Walk P' = sel_prob(beta.in.s),
    check.names = FALSE
  )
) |>
  arrange(match(Parameter, c("Gamma", "Eta", "Beta")))
tp_selprob_tbl <- bind_rows(lapply(seq_along(perc), function(j)
{
  data.frame(
    Percentile = paste0(perc[j] * 100, "%"),
    Population = pop_names,
    'Random Walk P' = sel_prob(tp_rw_list[[j]]),
    'Independent Walk P' = sel_prob(tp_in_list[[j]]),
    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)
  gamma_winner_in <- apply(g.in.s, 1, which.max)
  tp_winner_in  <- apply(tp_in_list[[j]], 1, which.max)
  data.frame(
    Percentile = paste0(perc[j] * 100, "%"),
    'RW P(gamma & t_p agree)' =
      round(mean(gamma_winner_rw == tp_winner_rw), 4),
    'Indep P(gamma & t_p agree)' =
      round(mean(gamma_winner_in == tp_winner_in), 4),
    check.names = FALSE
  )
}))
safe_ess <- function(x) {
  x <- as.numeric(x[is.finite(x)])
  if (length(x) < 100 || sd(x) == 0) return(NA_real_)
  as.numeric(coda::effectiveSize(coda::mcmc(x)))
}
chains_rw <- lapply(1:K, function(i) {
  coda::mcmc(
    cbind(
      gamma = g.rw.s[, i],
      eta = eta.rw.s[, i],
      beta = beta.rw.s[, i]
    )
  )
})

```

```

    )
  })
chains_in <- lapply(1:K, function(i) {
  coda::mcmc(
    cbind(
      gamma = g.in.s[, i],
      eta = eta.in.s[, i],
      beta = beta.in.s[, i]
    )
  )
})
ess_summary <- data.frame(
  Population = pop_names,
  'Gamma (RW)' = round(sapply(chains_rw,
    function(x) safe_ess(x[, "gamma"])), 1),
  'Eta (RW)' = round(sapply(chains_rw,
    function(x) safe_ess(x[, "eta"])), 1),
  'Beta (RW)' = round(sapply(chains_rw,
    function(x) safe_ess(x[, "beta"])), 1),
  'Gamma (Indep)' = round(sapply(chains_in,
    function(x) safe_ess(x[, "gamma"])), 1),
  'Eta (Indep)' = round(sapply(chains_in,
    function(x) safe_ess(x[, "eta"])), 1),
  'Beta (Indep)' = round(sapply(chains_in,
    function(x) safe_ess(x[, "beta"])), 1),
  check.names = FALSE

```

```

)
tp_ess_tbl <- bind_rows(lapply(seq_along(perc),
function(j) {
  data.frame(
    Percentile = paste0(perc[j] * 100, "%"),
    Population = pop_names,
    't_p ESS (RW)' = round(sapply(1:K, function(i)
      safe_ess(tp_rw_list[[j]][, i])), 1),
    't_p ESS (Indep)' = round(sapply(1:K, function(i)
      safe_ess(tp_in_list[[j]][, i])), 1),
    check.names = FALSE
  )
}))
acceptance_df <- bind_rows(lapply(1:K, function(i) {
  rw_acc <- sapply(results[[i]]$chains,
function(ch) ch$acc_rw)
  in_acc <- sapply(results[[i]]$chains,
function(ch) ch$acc_in)
  data.frame(
    Population = pop_names[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),
    'Indep Acceptance Mean' = round(mean(in_acc), 3),
    'Indep Acceptance Min' = round(min(in_acc), 3),
    'Indep Acceptance Max' = round(max(in_acc), 3),

```

```

      check.names = FALSE
    )
  )))
rhat_tbl <- bind_rows(lapply(1:K, function(i) {
  rh_rw <- rhat_rw_results[[i]]
  rh_in <- rhat_in_results[[i]]
  bind_rows(
    data.frame(
      Population = pop_names[i],
      Sampler = "Random Walk",
      Parameter = rownames(rh_rw),
      'R-hat' = round(rh_rw[, "Point est."], 4),
      'Upper CI' = round(rh_rw[, "Upper C.I."], 4),
      check.names = FALSE
    ),
    data.frame(
      Population = pop_names[i],
      Sampler = "Independent Walk",
      Parameter = rownames(rh_in),
      'R-hat' = round(rh_in[, "Point est."], 4),
      'Upper CI' = round(rh_in[, "Upper C.I."], 4),
      check.names = FALSE
    )
  )
}))
rhat_tbl <- rhat_tbl |>

```

```

mutate(
  Flag = ifelse(
    is.na('R-hat'),
    "Not computed",
    ifelse(
      'R-hat' > 1.1,
      "WARNING",
      ifelse('R-hat' > 1.05, "Check", "OK")
    )
  )
)

sensitivity_results <- bind_rows(lapply(theta3_sd_grid,
function(sd_val)
{ bind_rows(lapply(1:K, function(i) {
  data_i <- datasets[[i]]
  logpost_sd <- make_logpost_gamma(theta3_sd = sd_val)
  start <- get_start(data_i)
  pilot <- tryCatch(
    rw_pilot(logpost_sd, start, data_i, iters = 5000),
    error = function(e) list(mode = start, var = diag(0.2, 3))
  )
  pilot$var <- regularize_covariance(pilot$var)
  r.walk <- LearnBayes::rwmetrop(
    logpost_sd,
    list(var = pilot$var, scale = rw_scale_main),
    pilot$mode,

```

```

    30000,
    data_i
  )
  post <- r.walk$par[-(1:5000), , drop = FALSE]
  t1 <- min(data_i)
  delta_draws <- exp(post[, 3])
  g_draws <- t1 - delta_draws - PAD
  data.frame(
    theta3_sd = sd_val,
    Population = pop_names[i],
    gamma_mean = round(mean(g_draws), 4),
    gamma_2.5 = round(quantile(g_draws, 0.025), 4),
    gamma_97.5 = round(quantile(g_draws, 0.975), 4),
    delta_mean = round(mean(delta_draws), 4),
    delta_2.5 = round(quantile(delta_draws, 0.025), 4),
    delta_97.5 = round(quantile(delta_draws, 0.975), 4),
    check.names = FALSE
  )
})))
}))
rweibull3 <- function(n, beta, eta, gamma) {
  gamma + stats::rweibull(n, shape = beta, scale = eta)
}
ppc_exact <- function(x, beta_draws, eta_draws, gamma_draws,
n_rep = 1000)
{n <- length(x)

```

```

obs_stats <- c(
  min = min(x),
  mean = mean(x),
  sd = sd(x),
  q10 = as.numeric(quantile(x, 0.10)),
  median = median(x),
  q90 = as.numeric(quantile(x, 0.90))
)

idx <- sample(seq_along(beta_draws), n_rep, replace = TRUE)

sim_stats <- t(apply(idx, function(s) {
  xs <- rweibull3(
    n,
    beta_draws[s],
    eta_draws[s],
    gamma_draws[s]
  )
  c(
    min = min(xs),
    mean = mean(xs),
    sd = sd(xs),
    q10 = as.numeric(quantile(xs, 0.10)),
    median = median(xs),
    q90 = as.numeric(quantile(xs, 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(1:K, function(i) {
  ppc <- ppc_exact(
    datasets[[i]],
    beta_draws = beta.rw.s[, i],
    eta_draws = eta.rw.s[, i],
    gamma_draws = g.rw.s[, i],
    n_rep = ppc_nrep
  )
  ppc$Population <- pop_names[i]
  ppc |> relocate(Population)
})))

```

B.18 Source Code for the MLE Fit of Alloy T7987 Data

```

# 3-Parameter Weibull MLE for Alloy T7987 Data
# Right-censored reliability data from Meeker SMRD2 data
# Model:

```



```

# X = gamma + W
# W ~ Weibull(shape = beta, scale = eta)
# Units:
# Time is measured in thousands of cycles
rm(list = ls())
library(survival)
library(MASS)
# 1. Enter the Alloy T7987 data
# 67 failures and 5 right-censored observations at
# 300 thousand cycles
alloy_raw <- data.frame(
  time = c(
    94, 96, 99, 99, 104, 108, 112, 114, 117, 117,
    118, 121, 121, 123, 129, 131, 133, 135, 136,
    139, 139, 140, 141, 141, 143, 144, 149, 149,
    152, 153, 159, 159, 159, 159, 162, 168, 168,
    169, 170, 170, 171, 172, 173, 176, 177, 180,
    180, 184, 187, 188, 189, 190, 196, 197, 203,
    205, 211, 213, 224, 226, 227, 256, 257, 269,
    271, 274, 291, 300
  ),
  status = c(rep(1, 67), 0), # 1 = failed,
  0 = right-censored
  count = c(rep(1, 67), 5)
)
# Expand counts so each specimen is one row

```

```

alloy <- alloy_raw[
  rep(seq_len(nrow(alloy_raw)), alloy_raw$count),
  c("time", "status")
]

# -----
# 2. Negative log-likelihood for right-censored 3P Weibull
# -----

# Parameterization:
#   theta[1] = log(beta)
#   theta[2] = log(eta)
#   theta[3] = log(delta)
# where
#   delta = min(time) - gamma - PAD
#   gamma = min(time) - delta - PAD
# This guarantees gamma < min(time).
PAD <- 1e-6

neg_loglik_3p_weibull_cens <- function(theta, data, pad = PAD){
  beta <- exp(theta[1])
  eta  <- exp(theta[2])
  delta <- exp(theta[3])
  t_min <- min(data$time)
  gamma <- t_min - delta - pad
  t <- data$time
  d <- data$status
  y <- t - gamma
  if (!is.finite(beta) || !is.finite(eta) || !is.finite(gamma)) {

```

```

    return(1e20)
  }
  if (beta <= 0 || eta <= 0 || any(y <= 0)) {
    return(1e20)
  }
  # Log density contribution for failures
  log_f <- log(beta) - beta * log(eta) +
    (beta - 1) * log(y) -
    (y / eta)^beta
  # Log survival contribution for censored observations
  log_S <- - (y / eta)^beta
  loglik <- sum(d * log_f + (1 - d) * log_S)
  return(-loglik)
}

# -----
# 3. Starting values and optimization
# -----

start_grid <- expand.grid(
  beta_start = c(0.8, 1.0, 1.2, 1.5, 2.0, 3.0),
  eta_start  = c(30, 50, 75, 100),
  delta_start = c(1, 5, 10, 20, 40, 70)
)

fits <- vector("list", nrow(start_grid))
values <- rep(Inf, nrow(start_grid))
for (i in seq_len(nrow(start_grid))) {
  start <- c(

```

```

      log(start_grid$beta_start[i]),
      log(start_grid$eta_start[i]),
      log(start_grid$delta_start[i])
    )
  fit_i <- try(
    optim(
      par = start,
      fn = neg_loglik_3p_weibull_cens,
      data = alloy,
      method = "BFGS",
      hessian = TRUE,
      control = list(maxit = 10000, reltol = 1e-10)
    ),
    silent = TRUE
  )
  if (!inherits(fit_i, "try-error") && is.finite(fit_i$value))
  { fits[[i]] <- fit_i
    values[i] <- fit_i$value
  }
}

best_index <- which.min(values)

best_fit <- fits[[best_index]]

# -----
# 4. Extract MLEs
# -----

theta_hat <- best_fit$par

```

```

beta_hat <- exp(theta_hat[1])
eta_hat  <- exp(theta_hat[2])
delta_hat <- exp(theta_hat[3])
gamma_hat <- min(alloy$time) - delta_hat - PAD
mle_table <- data.frame(
  Parameter = c("gamma", "eta", "beta", "delta"),
  Estimate = c(gamma_hat, eta_hat, beta_hat, delta_hat))
# -----
# 5. Approximate standard errors using Hessian
# -----
cov_theta <- try(solve(best_fit$hessian), silent = TRUE)
if (!inherits(cov_theta, "try-error") &&
all(is.finite(cov_theta))) {
  J <- matrix(0, nrow = 4, ncol = 3)
  rownames(J) <- c("gamma", "eta", "beta", "delta")
  colnames(J) <- c("log_beta", "log_eta", "log_delta")
  J["gamma", "log_delta"] <- -delta_hat
  J["eta", "log_eta"] <- eta_hat
  J["beta", "log_beta"] <- beta_hat
  J["delta", "log_delta"] <- delta_hat
  cov_orig <- J %*% cov_theta %*% t(J)
  se_orig <- sqrt(diag(cov_orig))
  se_table <- data.frame(
    Parameter = c("gamma", "eta", "beta", "delta"),
    Estimate = c(gamma_hat, eta_hat, beta_hat, delta_hat),
    SE = se_orig )}

```

```

# -----
# 6. Fitted Weibull functions
# -----

dweibull_3p <- function(t, gamma, eta, beta){
  ifelse(
    t > gamma,
    (beta / eta) *
      ((t - gamma) / eta)^(beta - 1) *
      exp(-((t - gamma) / eta)^beta),
    0 )}

pweibull_3p <- function(t, gamma, eta, beta){
  ifelse(
    t > gamma,
    1 - exp(-((t - gamma) / eta)^beta),
    0 )}

sweibull_3p <- function(t, gamma, eta, beta){
  ifelse(
    t > gamma,
    exp(-((t - gamma) / eta)^beta),
    1 )}

hweibull_3p <- function(t, gamma, eta, beta){
  ifelse(
    t > gamma,
    (beta / eta) * ((t - gamma) / eta)^(beta - 1),
    0 )}

# -----

```

```

# 7. Fitted quantiles
# -----
p <- c(0.10, 0.25, 0.50, 0.75, 0.90)
q_hat <- gamma_hat+eta_hat*(-log(1 - p))^(1 / beta_hat)
quantile_table <- data.frame(
  p = p,
  Fitted_Quantile_Thousands_Cycles = q_hat)
# -----

# 8. Kaplan-Meier estimate and Weibull
# probability plot setup
# -----

fail_times <- alloy$time[alloy$status == 1]
km_fit <- survfit(Surv(time, status) ~ 1, data = alloy)

# Kaplan-Meier summary at event times
km_sum <- summary(km_fit)
km_df <- data.frame(
  time = km_sum$time,
  surv = km_sum$surv,
  lower = km_sum$lower,
  upper = km_sum$upper,
  n_risk = km_sum$n.risk,
  n_event = km_sum$n.event
)

# Keep only valid event times for Weibull
#probability plot
km_df <- km_df[

```

```

km_df$n_event > 0 &
  km_df$time > gamma_hat &
  km_df$surv > 0 & km_df$surv < 1 &
  km_df$lower > 0 & km_df$lower < 1 &
  km_df$upper > 0 & km_df$upper < 1,
]

km_df$x <- log(km_df$time - gamma_hat)
km_df$y <- log(-log(km_df$surv))

# Confidence interval transformation
# lower survival gives upper transformed value
km_df$y_lower <- log(-log(km_df$upper))
km_df$y_upper <- log(-log(km_df$lower))

# Fitted Weibull probability plot line:
#  $\log(-\log(S(t))) = \beta * [\log(t - \gamma) - \log(\eta)]$ 
x_grid <- seq(min(km_df$x), max(km_df$x),
length.out = 200)
y_fit <- beta_hat * (x_grid - log(eta_hat))

# Optional fitted-line confidence band from
# MLE covariance
make_fit_line <- function(theta, x_grid, t_min,
pad = PAD) {
  beta <- exp(theta[1])
  eta <- exp(theta[2])
  beta * (x_grid - log(eta))

```



```

}

fit_band_available <- FALSE

if (exists("cov_theta") &&
    !inherits(cov_theta, "try-error") &&
    all(is.finite(cov_theta))) {
  set.seed(123)

  theta_sims <- try(
    MASS::mvrnorm(
      n = 2000,
      mu = theta_hat,
      Sigma = cov_theta
    ),
    silent = TRUE )

  if (!inherits(theta_sims, "try-error")){
    y_sims <- apply(
      theta_sims,
      1,
      make_fit_line,
      x_grid = x_grid,
      t_min = min(alloy$time),
      pad = PAD )

    y_band_lower <- apply(y_sims, 1, quantile,
      probs = 0.025, na.rm = TRUE)
    y_band_upper <- apply(y_sims, 1, quantile,
      probs = 0.975, na.rm = TRUE)
    fit_band_available <- TRUE }}

```

B.19 Source Code for the Bayesian Fit of Alloy T7987 Data

```
# =====  
# Bayesian 3-Parameter Weibull Fit in R  
# Fatigue Life Data for Alloy T7987  
# =====  
  
rm(list = ls())  
  
library(MASS)  
  
library(coda)  
  
library(dplyr)  
  
set.seed(123)  
  
# -----  
# 1. Data  
# -----  
  
alloy_raw <- data.frame(  
  time = c(  
    94, 96, 99, 99, 104, 108, 112, 114, 117, 117,  
    118, 121, 121, 123, 129, 131, 133, 135, 136,  
    139, 139, 140, 141, 141, 143, 144, 149, 149,  
    152, 153, 159, 159, 159, 159, 162, 168, 168,  
    169, 170, 170, 171, 172, 173, 176, 177, 180,  
    180, 184, 187, 188, 189, 190, 196, 197, 203,  
    205, 211, 213, 224, 226, 227, 256, 257, 269,  
    271, 274, 291, 300  
  ),  
  status = c(rep(1, 67), 0), # 1 = failure, 0 = right-censored  
  count = c(rep(1, 67), 5)
```

```

)

# Expand the censored observation at 300 into five rows
alloy <- alloy_raw[rep(seq_len(nrow(alloy_raw)), alloy_raw$count),
c("time", "status")]

N <- nrow(alloy)

# -----

# 2. Log-posterior

# -----

PAD <- 1e-6

logpost_3p_weibull_cens <- function(theta, data, pad = PAD) {
  # Transform back to original scale
  beta <- exp(theta[1])
  eta <- exp(theta[2])
  delta <- exp(theta[3])
  tmin <- min(data$time)
  gamma <- tmin - delta - pad
  t <- data$time
  d <- data$status
  y <- t - gamma

  # Validity checks
  if (!is.finite(beta) || !is.finite(eta) || !is.finite(delta))
    return(-Inf)
  if (beta <= 0 || eta <= 0 || delta <= 0 || any(y <= 0))
    return(-Inf)

  # Log density for exact failures
  log_f <- log(beta) - beta * log(eta) +

```

```

    (beta - 1) * log(y) -
    (y / eta)^beta
# Log survival for censored observations
log_S <- - (y / eta)^beta
# Censored log-likelihood
loglik <- sum(d * log_f + (1 - d) * log_S)
# Priors:
# beta ~ Gamma(6, scale = 0.25), mean = 1.5
# eta ~ Gamma(0.01, scale = 100), weak positive prior
# theta3 = log(delta) ~ Normal(log(20), sd = 10)
lp_beta <- dgamma(beta, shape = 6, scale = 0.25, log = TRUE)
+ theta[1]
lp_eta <- dgamma(eta, shape = 0.01, scale = 100, log = TRUE)
+ theta[2]
lp_t3 <- dnorm(theta[3], mean = log(20), sd = 10, log = TRUE)
loglik + lp_beta + lp_eta + lp_t3
}
# -----
# 3. MLE fit to get starting values
# -----
neg_loglik_3p_weibull_cens <- function(theta, data, pad = PAD){
  beta <- exp(theta[1])
  eta <- exp(theta[2])
  delta <- exp(theta[3])
  tmin <- min(data$time)
  gamma <- tmin - delta - pad

```

```

t <- data$time
d <- data$status
y <- t - gamma
if (!is.finite(beta) || !is.finite(eta) || !is.finite(delta))
return(1e20)
if (beta <= 0 || eta <= 0 || delta <= 0 || any(y <= 0))
return(1e20)
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)
-loglik
}

start_grid <- expand.grid(
  beta_start = c(0.8, 1.0, 1.2, 1.5, 2.0, 3.0),
  eta_start = c(30, 50, 75, 100),
  delta_start = c(1, 5, 10, 20, 40, 70)
)

fits <- vector("list", nrow(start_grid))
values <- rep(Inf, nrow(start_grid))
for (i in seq_len(nrow(start_grid))) {
  start <- c(
    log(start_grid$beta_start[i]),
    log(start_grid$eta_start[i]),
    log(start_grid$delta_start[i])
  )

```

```

)
fit_i <- try(
  optim(
    par = start,
    fn = neg_loglik_3p_weibull_cens,
    data = alloy,
    method = "BFGS",
    hessian = TRUE,
    control = list(maxit = 10000, reltol = 1e-10)
  ),
  silent = TRUE )
if (!inherits(fit_i, "try-error") && is.finite(fit_i$value)){
  fits[[i]] <- fit_i
  values[i] <- fit_i$value }}
best_fit <- fits[[which.min(values)]]
theta_mle <- best_fit$par
beta_hat <- exp(theta_mle[1])
eta_hat <- exp(theta_mle[2])
delta_hat <- exp(theta_mle[3])
gamma_hat <- min(alloy$time) - delta_hat - PAD
mle_table <- data.frame(
  Parameter = c("gamma", "eta", "beta", "delta"),
  MLE = c(gamma_hat, eta_hat, beta_hat, delta_hat)
)
# -----
# 4. Random-walk Metropolis-Hastings

```

```

# -----
rw_mh <- function(logpost, start, data, n_iter = 50000,
proposal_cov){
  p <- length(start)
  chain <- matrix(NA, nrow = n_iter, ncol = p)
  colnames(chain) <- c("log_beta", "log_eta", "log_delta")
  current <- start
  current_lp <- logpost(current, data)
  accept <- 0
  for (m in 1:n_iter) {
    proposal <- as.numeric(MASS::mvrnorm(1, mu = current,
Sigma = proposal_cov))
    proposal_lp <- logpost(proposal, data)
    log_alpha <- proposal_lp - current_lp
    if (log(runif(1)) < log_alpha) {
      current <- proposal
      current_lp <- proposal_lp
      accept <- accept + 1 }
    chain[m, ] <- current }
  list(
    chain = chain,
    accept_rate = accept / n_iter )}
# Proposal covariance from inverse Hessian, regularized
cov_theta <- try(solve(best_fit$hessian), silent = TRUE)
if (inherits(cov_theta, "try-error") || any(!is.finite(cov_theta))){
  cov_theta <- diag(c(0.05, 0.05, 0.10)^2)}

```

```

cov_theta <- (cov_theta + t(cov_theta)) / 2
proposal_cov <- cov_theta * 1.5
# Multiple chains
n_chains <- 4
n_iter <- 200000
burn_in <- 20000
thin <- 5
starts <- list(
  theta_mle + c(0.05, 0.05, 0.10),
  theta_mle + c(-0.05, 0.05, -0.10),
  theta_mle + c(0.05, -0.05, 0.20),
  theta_mle + c(-0.05, -0.05, -0.20))
mh_fits <- vector("list", n_chains)
for (ch in 1:n_chains) {
  mh_fits[[ch]] <- rw_mh(
    logpost = logpost_3p_weibull_cens,
    start = starts[[ch]],
    data = alloy,
    n_iter = n_iter,
    proposal_cov = proposal_cov
  )}

# -----
# 5. Convert posterior samples to gamma, eta, beta, delta
# -----

convert_chain <- function(theta_chain, data, burn_in, thin,
pad = PAD) {

```



```

keep <- seq(burn_in + 1, nrow(theta_chain), by = thin)
theta_keep <- theta_chain[keep, , drop = FALSE]
beta <- exp(theta_keep[, "log_beta"])
eta <- exp(theta_keep[, "log_eta"])
delta <- exp(theta_keep[, "log_delta"])
gamma <- min(data$time) - delta - pad
q10 <- gamma + eta * (-log(1 - 0.10))^(1 / beta)
q25 <- gamma + eta * (-log(1 - 0.25))^(1 / beta)
q50 <- gamma + eta * (-log(1 - 0.50))^(1 / beta)
q75 <- gamma + eta * (-log(1 - 0.75))^(1 / beta)
q90 <- gamma + eta * (-log(1 - 0.90))^(1 / beta)
out <- cbind(
  gamma = gamma,
  eta = eta,
  beta = beta,
  delta = delta,
  q10 = q10,
  q25 = q25,
  q50 = q50,
  q75 = q75,
  q90 = q90
)
as.mcmc(out)
}

mcmc_list <- mcmc.list(
  lapply(mh_fits, function(fit) {

```

```

    convert_chain(fit$chain, alloy, burn_in, thin)
  )))
# -----
# 6. Posterior summaries
# -----

posterior_mat <- as.matrix(mcmc_list)

posterior_summary <- data.frame(
  Parameter = colnames(posterior_mat),
  Posterior_Mean = apply(posterior_mat, 2, mean),
  Posterior_SD = apply(posterior_mat, 2, sd),
  Q2.5 = apply(posterior_mat, 2, quantile, probs = 0.025),
  Median = apply(posterior_mat, 2, median),
  Q97.5 = apply(posterior_mat, 2, quantile, probs = 0.975),
  row.names = NULL)

# Rhat and ESS

gelman <- gelman.diag(
  mcmc_list,
  autoburnin = FALSE,
  multivariate = FALSE
)$psrf

ess <- effectiveSize(mcmc_list)

posterior_summary$Rhat <- gelman[posterior_summary$Parameter,
"Point est."]

posterior_summary$n_eff <- ess[posterior_summary$Parameter]

posterior_summary <- posterior_summary %>%
  mutate(

```

```

Posterior_Mean = round(Posterior_Mean, 4),
Posterior_SD   = round(Posterior_SD, 4),
Q2.5           = round(Q2.5, 4),
Median         = round(Median, 4),
Q97.5          = round(Q97.5, 4),
Rhat           = round(Rhat, 3),
n_eff          = round(n_eff, 0) )

# -----
# 7. MLE vs Bayesian comparison
# -----

bayes_main <- posterior_summary %>%
  filter(Parameter %in% c("gamma", "eta", "beta",
    "delta")) %>%
  dplyr::select(Parameter, Posterior_Mean, Median,
    Q2.5, Q97.5)

comparison_table <- merge(mle_table, bayes_main,
  by = "Parameter")

```

B.20 Source Code for the Bayesian Subset Selection of Semi-Simulated Alloy T7987

Study

```

# =====
# Bayesian 3-Parameter Weibull Model
# Semi-Simulated Alloy T7987 Five-Population
# Censored Example
# 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(coda)
set.seed(2026)

# -----
# settings
# -----

PAD <- 1e-6
n_chains <- 4
iters_long <- 200000
burn_in <- 20000
thin <- 3
rw_scale_main <- 1.2
# Independent sampler settings
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$Population))
K <- length(pop_names)
# Split into list by population
datasets <- split(
  five_pop_data[, c("time", "status")],
  five_pop_data$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 Table 0 parameters
# -----
# 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 prior:
# beta ~ Gamma(6, scale = 0.25), mean = 1.5

```

```

main_a_beta <- 6
main_b_beta <- 0.25
# eta prior:
# 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
theta3_sd_grid <- c(1.0, 2.0, 3.0)
# 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 3P Weibull
# -----
# 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_weibull3_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_weibull3_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. MCMC 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)) )}
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
  } else {
  }
  ok}
# -----
# 6. Independent sampler helper functions
# -----

```

```

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) * log1p(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 )}

# -----
# 7. Multiple-chain MCMC for each population: RW and Indep
# -----

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 RW
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$var)
# Random-walk Metropolis
r.walk <- LearnBayes::rwmetrop(
  logpost_i,
  list(var = Sigma_pilot, scale = rw_scale_main),
  pilot$mode,
  iters_long,
  data_i )
# Laplace approximation for independent proposal
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$mode
prop <- list(
    mu = as.numeric(mu_prop),
    var = Sigma_prop * indep_inflate,
    df = mixture_df,
    mix_p = mixture_p )
# Independent/t-mixture Metropolis
in.walk <- indepmetrop_mix(
    logpost = logpost_i,
    start = mu_prop,
    data = data_i,
    iters = iters_long,
    prop = prop )
trim_thin <- function(m) {
    m <- m[-seq_len(burn_in), , drop = FALSE]
    m[seq(1, nrow(m), by = thin), , drop = FALSE] }
chain_results[[ch]] <- list(
    rw = trim_thin(r.walk$par),
    indep = trim_thin(in.walk$par),
    acc_rw = if (!is.null(r.walk$accept)) {
        r.walk$accept
    } else {
        mean(apply(diff(r.walk$par), 1,

```

```

        function(x) any(x != 0))) },

    acc_in = in.walk$acc ) }

results[[pop_i]] <- list(

    chains = chain_results,

    data = data_i )}

# -----

# 8. Rhat diagnostics for both samplers

# -----

get_rhat_for_sampler <- function(pop_i, sampler = "rw") {

    mclist <- coda::mcmc.list(

        lapply(results[[pop_i]]$chains, function(ch) {

            coda::mcmc(ch[[sampler]]) }) )

    rhat <- coda::gelman.diag(

        mclist,

        multivariate = FALSE,

        autoburnin = FALSE

    )$psrf

    rownames(rhat) <- c("log_beta", "log_eta", "log_delta")

    rhat}

rhat_rw_results <- lapply(pop_names, function(pop_i) {

    rh <- get_rhat_for_sampler(pop_i, sampler = "rw")

    rh})

rhat_in_results <- lapply(pop_names, function(pop_i) {

    rh <- get_rhat_for_sampler(pop_i, sampler = "indep")

    rh})

names(rhat_rw_results) <- pop_names

```

```

names(rhat_in_results) <- pop_names
rhat_tbl <- bind_rows(lapply(pop_names, function(pop_i){
  rh_rw <- rhat_rw_results[[pop_i]]
  rh_in <- rhat_in_results[[pop_i]]
  bind_rows(
    data.frame(
      Population = pop_i,
      Sampler = "Random Walk",
      Parameter = rownames(rh_rw),
      Rhat = round(rh_rw[, "Point est."], 4),
      Upper_CI = round(rh_rw[, "Upper C.I."], 4),
      check.names = FALSE ),
    data.frame(
      Population = pop_i,
      Sampler = "Independent",
      Parameter = rownames(rh_in),
      Rhat = round(rh_in[, "Point est."], 4),
      Upper_CI = round(rh_in[, "Upper C.I."], 4),
      check.names = FALSE ) )
  )))
rhat_tbl <- rhat_tbl %>%
  mutate(
    Flag = ifelse(
      Rhat > 1.10, "WARNING",
      ifelse(Rhat > 1.05, "Check", "OK") ) )
# -----

```

```

# 9. Pool chains and recover natural parameters
# -----
pool_chains <- function(results_i, sampler = "rw") {
  do.call(rbind, lapply(results_i$chains, function(ch) {
    if (sampler == "rw") ch$rw else ch$indep
  })))
recover_params_cens <- function(draws_mat, data_i) {
  t1 <- min(data_i$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_chains(results[[pop_i]], "rw"),
    datasets[[pop_i]]))
params_in <- lapply(pop_names, function(pop_i) {
  recover_params_cens(pool_chains(results[[pop_i]], "indep"),
    datasets[[pop_i]]))
names(params_rw) <- pop_names
names(params_in) <- 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")
g.in.s <- assemble_mat(params_in, "gamma")
eta.rw.s <- assemble_mat(params_rw, "eta")
eta.in.s <- assemble_mat(params_in, "eta")
beta.rw.s <- assemble_mat(params_rw, "beta")
beta.in.s <- assemble_mat(params_in, "beta")
delta.rw.s <- assemble_mat(params_rw, "delta")
delta.in.s <- assemble_mat(params_in, "delta")

# -----
# 10. Posterior summaries for RW and independent samplers
# -----

make_summary_rows_both <- function(mat_rw, mat_in,
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),
    Indep_Mean = round(colMeans(mat_in), 4),

```

```

    Indep_Median = round(apply(mat_in, 2, median), 4),
    Indep_Q2.5 = round(apply(mat_in, 2,
quantile, probs = 0.025), 4),
    Indep_Q97.5 = round(apply(mat_in, 2,
quantile, probs = 0.975), 4),
    check.names = FALSE
  )
}

posterior_summary <- bind_rows(
  make_summary_rows_both(g.rw.s, g.in.s, "Gamma"),
  make_summary_rows_both(eta.rw.s, eta.in.s, "Eta"),
  make_summary_rows_both(beta.rw.s, beta.in.s, "Beta"),
  make_summary_rows_both(delta.rw.s, delta.in.s, "Delta")
)

# -----
# 11. Posterior probability each population is best by gamma
# -----

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),
  Indep_P_Best_Gamma = round(sel_prob(g.in.s), 4),
  check.names = FALSE)

# Posterior probability 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),
  Indep_P_Best_Eta = round(sel_prob(eta.in.s), 4),
  check.names = FALSE)

selection_probs_beta <- data.frame(
  Population = pop_names,
  RW_P_Best_Beta = round(sel_prob(beta.rw.s), 4),
  Indep_P_Best_Beta = round(sel_prob(beta.in.s), 4),
  check.names = FALSE)

# Combined table: posterior probabilities of having the la
# rgest parameter value

selection_probs_parameters <- bind_rows(
  data.frame(
    Parameter = "Gamma",
    Population = pop_names,
    RW_Probability = round(sel_prob(g.rw.s), 4),
    Indep_Probability = round(sel_prob(g.in.s), 4),
    check.names = FALSE ),
  data.frame(
    Parameter = "Eta",
    Population = pop_names,
    RW_Probability = round(sel_prob(eta.rw.s), 4),
    Indep_Probability = round(sel_prob(eta.in.s), 4),
    check.names = FALSE),

```

```

data.frame(
  Parameter = "Beta",
  Population = pop_names,
  RW_Probability = round(sel_prob(beta.rw.s), 4),
  Indep_Probability = round(sel_prob(beta.in.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))
selected_subset_rw <- selection_order_rw[
  seq_len(which(selection_order_rw$Cumulative_Probability >=
    P_star)[1]),
]

selection_order_in <- selection_probs_gamma %>%
  arrange(desc(Indep_P_Best_Gamma)) %>%
  mutate(Cumulative_Probability =
    cumsum(Indep_P_Best_Gamma))
selected_subset_in <- selection_order_in[
  seq_len(which(selection_order_in$Cumulative_Probability >=
    P_star)[1]),
]

# -----
# 12. 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)
})

tp_in_list <- lapply(perc, function(p) {
  tp_from_draws(g.in.s, eta.in.s, beta.in.s, p)
})

names(tp_rw_list) <- names(tp_in_list) <- paste0("t_", perc * 100)
tp_summary <- bind_rows(lapply(seq_along(perc), function(j) {
  make_summary_rows_both(
    tp_rw_list[[j]],
    tp_in_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),
    Indep_P_Largest = round(sel_prob(tp_in_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)
  gamma_winner_in <- apply(g.in.s, 1, which.max)
  tp_winner_in <- apply(tp_in_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),
    Indep_P_Gamma_and_tp_Agree = round(mean(gamma_winner_in ==
    tp_winner_in), 4),
    check.names = FALSE
  )
}))

# -----
# 13. 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)
  in_acc <- sapply(results[[pop_i]]$chains, function(ch) ch$acc_in)
  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),
Indep_Acceptance_Mean = round(mean(in_acc), 3),
Indep_Acceptance_Min = round(min(in_acc), 3),
Indep_Acceptance_Max = round(max(in_acc), 3),
check.names = FALSE
)
}))
# -----
# 14. Effective sample size for both samplers
# -----
safe_ess <- function(x) {
  x <- as.numeric(x[is.finite(x)])
  if (length(x) < 100 || sd(x) == 0) return(NA_real_)
  as.numeric(coda::effectiveSize(coda::mcmc(x)))
}
ess_summary <- data.frame(
  Population = pop_names,
  Gamma_RW = round(sapply(seq_len(K), function(i)
    safe_ess(g.rw.s[, i])), 1),
  Eta_RW = round(sapply(seq_len(K), function(i)
    safe_ess(eta.rw.s[, i])), 1),
  Beta_RW = round(sapply(seq_len(K), function(i)
    safe_ess(beta.rw.s[, i])), 1),
  Delta_RW = round(sapply(seq_len(K), function(i)
    safe_ess(delta.rw.s[, i])), 1),

```

```

Gamma_Indep = round(sapply(seq_len(K), function(i)
safe_ess(g.in.s[, i])), 1),
Eta_Indep  = round(sapply(seq_len(K), function(i)
safe_ess(eta.in.s[, i])), 1),
Beta_Indep = round(sapply(seq_len(K), function(i)
safe_ess(beta.in.s[, i])), 1),
Delta_Indep = round(sapply(seq_len(K), function(i)
safe_ess(delta.in.s[, i])), 1),
check.names = FALSE)

# Effective sample size for Weibull percentiles t10, t50, and t90
ess_tp_summary <- data.frame(
  Population = pop_names,
  t10_RW = round(sapply(seq_len(K), function(i)
safe_ess(tp_rw_list[[1]][, i])), 1),
  t50_RW = round(sapply(seq_len(K), function(i)
safe_ess(tp_rw_list[[2]][, i])), 1),
  t90_RW = round(sapply(seq_len(K), function(i)
safe_ess(tp_rw_list[[3]][, i])), 1),
  t10_Indep = round(sapply(seq_len(K), function(i)
safe_ess(tp_in_list[[1]][, i])), 1),
  t50_Indep = round(sapply(seq_len(K), function(i)
safe_ess(tp_in_list[[2]][, i])), 1),
  t90_Indep = round(sapply(seq_len(K), function(i)
safe_ess(tp_in_list[[3]][, i])), 1),
  check.names = FALSE)

# -----

```

```

# 15. 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$Population <- pop_names[i]
ppc |> relocate(Population)))))
# -----
# 16. Prior sensitivity analysis for theta3_sd
# -----
sensitivity_results <- bind_rows(lapply(theta3_sd_grid,
function(sd_val) {
  bind_rows(lapply(pop_names, function(pop_i) {
    data_i <- datasets[[pop_i]]
    logpost_sd <- make_logpost_cens(theta3_sd = sd_val)
    start <- get_start_cens(data_i)
    pilot <- tryCatch(
      rw_pilot(logpost_sd, start, data_i, iters = 5000),
      error = function(e) list(mode = start, var = diag(0.2, 3)))
    r.walk <- LearnBayes::rwmetrop(
      logpost_sd,
      list(var = pilot$var, scale = rw_scale_main),
      pilot$mode,
      30000,
      data_i )
    post <- r.walk$par[-seq_len(5000), , drop = FALSE]
    t1 <- min(data_i$time)
    delta_draws <- exp(post[, 3])

```

```

gamma_draws <- t1 - delta_draws - PAD
data.frame(
  theta3_sd = sd_val,
  Population = pop_i,
  gamma_mean = round(mean(gamma_draws), 4),
  gamma_2.5 = round(quantile(gamma_draws, 0.025), 4),
  gamma_97.5 = round(quantile(gamma_draws, 0.975), 4),
  delta_mean = round(mean(delta_draws), 4),
  delta_2.5 = round(quantile(delta_draws, 0.025), 4),
  delta_97.5 = round(quantile(delta_draws, 0.975), 4),
  check.names = FALSE )
)))
)))

```


APPENDIX C
EXTRA TABLES AND FIGURES

C.1 Complete Tables for OCs of Rule 1, Rule 2, and Rule 3 when $k = 2$

C.1.1 Complete Tables for OCs of Rule 1 vs Rule 2: $n = 5, P^* = 0.95$

Table C.1: Inequality-check table for R_1 versus the CLT-based rule $R_2, n = 5, P^* = 0.95$.

δ	$\Delta_{\text{PCS},2}$	Ineq _{PCS} holds	$\Delta_{\text{PICS},2}$	Ineq _{PICS} holds	$\Delta_{\text{ESS},2}$	Ineq _{ESS} holds
0.0	1.11022×10^{-16}	TRUE	1.11022×10^{-16}	TRUE	2.22045×10^{-16}	TRUE
0.1	5.37013×10^{-3}	TRUE	-1.38943×10^{-2}	TRUE	-8.52416×10^{-3}	TRUE
0.2	6.54079×10^{-3}	TRUE	-4.39284×10^{-2}	TRUE	-3.73876×10^{-2}	TRUE
0.3	5.88048×10^{-3}	TRUE	-1.03187×10^{-1}	TRUE	-9.73065×10^{-2}	TRUE
0.4	4.61712×10^{-3}	TRUE	-2.13779×10^{-1}	TRUE	-2.09162×10^{-1}	TRUE
0.5	3.33294×10^{-3}	TRUE	-3.93101×10^{-1}	TRUE	-3.89768×10^{-1}	TRUE
0.6	2.26036×10^{-3}	TRUE	-5.07904×10^{-1}	TRUE	-5.05643×10^{-1}	TRUE
0.7	1.45492×10^{-3}	TRUE	-5.53744×10^{-1}	TRUE	-5.52289×10^{-1}	TRUE
0.8	8.92723×10^{-4}	TRUE	-5.56428×10^{-1}	TRUE	-5.55535×10^{-1}	TRUE
0.9	5.22471×10^{-4}	TRUE	-5.32231×10^{-1}	TRUE	-5.31709×10^{-1}	TRUE
1.0	2.90786×10^{-4}	TRUE	-4.91712×10^{-1}	TRUE	-4.91421×10^{-1}	TRUE
1.1	1.52713×10^{-4}	TRUE	-4.41962×10^{-1}	TRUE	-4.41810×10^{-1}	TRUE
1.2	7.44494×10^{-5}	TRUE	-3.87929×10^{-1}	TRUE	-3.87854×10^{-1}	TRUE
1.3	3.24859×10^{-5}	TRUE	-3.33156×10^{-1}	TRUE	-3.33123×10^{-1}	TRUE
1.4	1.14606×10^{-5}	TRUE	-2.80207×10^{-1}	TRUE	-2.80195×10^{-1}	TRUE
1.5	1.87246×10^{-6}	TRUE	-2.30892×10^{-1}	TRUE	-2.30890×10^{-1}	TRUE
1.6	-1.85259×10^{-6}	FALSE	-1.86409×10^{-1}	TRUE	-1.86411×10^{-1}	TRUE
1.7	-2.81183×10^{-6}	FALSE	-1.47438×10^{-1}	TRUE	-1.47441×10^{-1}	TRUE
1.8	-2.62441×10^{-6}	FALSE	-1.14220×10^{-1}	TRUE	-1.14223×10^{-1}	TRUE
1.9	-2.07499×10^{-6}	FALSE	-8.66499×10^{-2}	TRUE	-8.66520×10^{-2}	TRUE
2.0	-1.50444×10^{-6}	FALSE	-6.43528×10^{-2}	TRUE	-6.43543×10^{-2}	TRUE

C.1.2 Complete Tables for OCs of Rule 1 vs Rule 2: $n = 5, P^* = 0.99$

Table C.2: Inequality-check table for R_1 versus the CLT-based rule R_2 , $n = 5, P^* = 0.99$.

δ	$\Delta_{\text{PCS},2}$	Ineq _{PCS} holds	$\Delta_{\text{PICS},2}$	Ineq _{PICS} holds	$\Delta_{\text{ESS},2}$	Ineq _{ESS} holds
0.0	0.00000×10^0	TRUE	0.00000×10^0	TRUE	0.00000×10^0	TRUE
0.1	4.22066×10^{-4}	TRUE	-1.41677×10^{-3}	TRUE	-9.94703×10^{-4}	TRUE
0.2	4.35110×10^{-4}	TRUE	-4.97358×10^{-3}	TRUE	-4.53847×10^{-3}	TRUE
0.3	3.18376×10^{-4}	TRUE	-1.28044×10^{-2}	TRUE	-1.24861×10^{-2}	TRUE
0.4	1.90827×10^{-4}	TRUE	-2.87477×10^{-2}	TRUE	-2.85569×10^{-2}	TRUE
0.5	9.29142×10^{-5}	TRUE	-5.95286×10^{-2}	TRUE	-5.94357×10^{-2}	TRUE
0.6	3.03809×10^{-5}	TRUE	-1.16702×10^{-1}	TRUE	-1.16672×10^{-1}	TRUE
0.7	-3.66243×10^{-6}	FALSE	-2.19837×10^{-1}	TRUE	-2.19841×10^{-1}	TRUE
0.8	-1.86142×10^{-5}	FALSE	-3.97863×10^{-1}	TRUE	-3.97882×10^{-1}	TRUE
0.9	-2.24513×10^{-5}	FALSE	-5.39098×10^{-1}	TRUE	-5.39120×10^{-1}	TRUE
1.0	-2.07489×10^{-5}	FALSE	-6.03479×10^{-1}	TRUE	-6.03499×10^{-1}	TRUE
1.1	-1.69133×10^{-5}	FALSE	-6.19261×10^{-1}	TRUE	-6.19278×10^{-1}	TRUE
1.2	-1.27721×10^{-5}	FALSE	-6.04063×10^{-1}	TRUE	-6.04076×10^{-1}	TRUE
1.3	-9.15002×10^{-6}	FALSE	-5.69168×10^{-1}	TRUE	-5.69177×10^{-1}	TRUE
1.4	-6.30534×10^{-6}	FALSE	-5.22090×10^{-1}	TRUE	-5.22096×10^{-1}	TRUE
1.5	-4.21760×10^{-6}	FALSE	-4.68083×10^{-1}	TRUE	-4.68087×10^{-1}	TRUE
1.6	-2.75624×10^{-6}	FALSE	-4.10996×10^{-1}	TRUE	-4.10998×10^{-1}	TRUE
1.7	-1.76853×10^{-6}	FALSE	-3.53743×10^{-1}	TRUE	-3.53745×10^{-1}	TRUE
1.8	-1.11854×10^{-6}	FALSE	-2.98550×10^{-1}	TRUE	-2.98552×10^{-1}	TRUE
1.9	-6.99550×10^{-7}	FALSE	-2.47073×10^{-1}	TRUE	-2.47074×10^{-1}	TRUE
2.0	-4.33746×10^{-7}	FALSE	-2.00462×10^{-1}	TRUE	-2.00463×10^{-1}	TRUE

C.1.3 Complete Tables for OCs of Rule 1 vs Rule 3: $n = 5, P^* = 0.95$

Table C.3: Inequality-check table for R_1 versus the Gamma-based rule R_3 , $n = 5, P^* = 0.95$.

δ	$\Delta_{\text{PCS},3}$	IneqPCS holds	$\Delta_{\text{PICS},3}$	IneqPICS holds	$\Delta_{\text{ESS},3}$	IneqESS holds
0.0	4.32987×10^{-15}	TRUE	4.32987×10^{-15}	TRUE	8.65974×10^{-15}	TRUE
0.1	6.89238×10^{-3}	TRUE	-1.59251×10^{-2}	TRUE	-9.03268×10^{-3}	TRUE
0.2	9.06007×10^{-3}	TRUE	-4.83815×10^{-2}	TRUE	-3.93214×10^{-2}	TRUE
0.3	8.92513×10^{-3}	TRUE	-1.10217×10^{-1}	TRUE	-1.01292×10^{-1}	TRUE
0.4	7.80845×10^{-3}	TRUE	-2.23182×10^{-1}	TRUE	-2.15374×10^{-1}	TRUE
0.5	6.39845×10^{-3}	TRUE	-4.04231×10^{-1}	TRUE	-3.97833×10^{-1}	TRUE
0.6	5.02827×10^{-3}	TRUE	-5.19652×10^{-1}	TRUE	-5.14624×10^{-1}	TRUE
0.7	3.83771×10^{-3}	TRUE	-5.64633×10^{-1}	TRUE	-5.60795×10^{-1}	TRUE
0.8	2.86615×10^{-3}	TRUE	-5.64816×10^{-1}	TRUE	-5.61950×10^{-1}	TRUE
0.9	2.10478×10^{-3}	TRUE	-5.36630×10^{-1}	TRUE	-5.34526×10^{-1}	TRUE
1.0	1.52486×10^{-3}	TRUE	-4.91124×10^{-1}	TRUE	-4.89599×10^{-1}	TRUE
1.1	1.09244×10^{-3}	TRUE	-4.36144×10^{-1}	TRUE	-4.35052×10^{-1}	TRUE
1.2	7.75299×10^{-4}	TRUE	-3.77470×10^{-1}	TRUE	-3.76694×10^{-1}	TRUE
1.3	5.45787×10^{-4}	TRUE	-3.19332×10^{-1}	TRUE	-3.18786×10^{-1}	TRUE
1.4	3.81516×10^{-4}	TRUE	-2.64677×10^{-1}	TRUE	-2.64295×10^{-1}	TRUE
1.5	2.65031×10^{-4}	TRUE	-2.15350×10^{-1}	TRUE	-2.15085×10^{-1}	TRUE
1.6	1.83092×10^{-4}	TRUE	-1.72294×10^{-1}	TRUE	-1.72111×10^{-1}	TRUE
1.7	1.25855×10^{-4}	TRUE	-1.35751×10^{-1}	TRUE	-1.35625×10^{-1}	TRUE
1.8	8.61202×10^{-5}	TRUE	-1.05476×10^{-1}	TRUE	-1.05390×10^{-1}	TRUE
1.9	5.86870×10^{-5}	TRUE	-8.09156×10^{-2}	TRUE	-8.08569×10^{-2}	TRUE
2.0	3.98407×10^{-5}	TRUE	-6.13546×10^{-2}	TRUE	-6.13148×10^{-2}	TRUE

C.1.4 Complete Tables for OCs of Rule 1 vs Rule 3: $n = 5, P^* = 0.99$

Table C.4: Inequality-check table for R_1 versus the Gamma-based rule R_3 , $n = 5, P^* = 0.99$.

δ	$\Delta_{\text{PCS},3}$	Ineq _{PCS} holds	$\Delta_{\text{PICS},3}$	Ineq _{PICS} holds	$\Delta_{\text{ESS},3}$	Ineq _{ESS} holds
0.0	-1.11022×10^{-16}	TRUE	-1.11022×10^{-16}	TRUE	-2.22045×10^{-16}	TRUE
0.1	1.08571×10^{-3}	TRUE	-2.59545×10^{-3}	TRUE	-1.50974×10^{-3}	TRUE
0.2	1.40350×10^{-3}	TRUE	-8.02176×10^{-3}	TRUE	-6.61826×10^{-3}	TRUE
0.3	1.36006×10^{-3}	TRUE	-1.85903×10^{-2}	TRUE	-1.72302×10^{-2}	TRUE
0.4	1.17088×10^{-3}	TRUE	-3.82899×10^{-2}	TRUE	-3.71190×10^{-2}	TRUE
0.5	9.44456×10^{-4}	TRUE	-7.39323×10^{-2}	TRUE	-7.29878×10^{-2}	TRUE
0.6	7.30895×10^{-4}	TRUE	-1.37051×10^{-1}	TRUE	-1.36320×10^{-1}	TRUE
0.7	5.49558×10^{-4}	TRUE	-2.47047×10^{-1}	TRUE	-2.46497×10^{-1}	TRUE
0.8	4.04509×10^{-4}	TRUE	-4.32502×10^{-1}	TRUE	-4.32097×10^{-1}	TRUE
0.9	2.92892×10^{-4}	TRUE	-5.81205×10^{-1}	TRUE	-5.80913×10^{-1}	TRUE
1.0	2.09310×10^{-4}	TRUE	-6.52424×10^{-1}	TRUE	-6.52214×10^{-1}	TRUE
1.1	1.47979×10^{-4}	TRUE	-6.73675×10^{-1}	TRUE	-6.73527×10^{-1}	TRUE
1.2	1.03681×10^{-4}	TRUE	-6.61902×10^{-1}	TRUE	-6.61798×10^{-1}	TRUE
1.3	7.20877×10^{-5}	TRUE	-6.27921×10^{-1}	TRUE	-6.27848×10^{-1}	TRUE
1.4	4.97892×10^{-5}	TRUE	-5.79125×10^{-1}	TRUE	-5.79076×10^{-1}	TRUE
1.5	3.41883×10^{-5}	TRUE	-5.21062×10^{-1}	TRUE	-5.21028×10^{-1}	TRUE
1.6	2.33548×10^{-5}	TRUE	-4.58245×10^{-1}	TRUE	-4.58222×10^{-1}	TRUE
1.7	1.58807×10^{-5}	TRUE	-3.94459×10^{-1}	TRUE	-3.94443×10^{-1}	TRUE
1.8	1.07535×10^{-5}	TRUE	-3.32778×10^{-1}	TRUE	-3.32767×10^{-1}	TRUE
1.9	7.25419×10^{-6}	TRUE	-2.75496×10^{-1}	TRUE	-2.75489×10^{-1}	TRUE
2.0	4.87667×10^{-6}	TRUE	-2.24108×10^{-1}	TRUE	-2.24103×10^{-1}	TRUE

C.1.5 Complete Tables for OCs of Rule 1 vs Rule 2: $n = 25$, $P^* = 0.95$

Table C.5: Inequality-check table for R_1 versus the CLT-based rule R_2 , $n = 25$, $P^* = 0.95$.

δ	$\Delta_{\text{PCS},2}$	Ineq _{PCS} holds	$\Delta_{\text{PICS},2}$	Ineq _{PICS} holds	$\Delta_{\text{ESS},2}$	Ineq _{ESS} holds
0.0	1.11022×10^{-16}	TRUE	1.11022×10^{-16}	TRUE	2.22045×10^{-16}	TRUE
0.1	1.87320×10^{-2}	TRUE	-4.91275×10^{-1}	TRUE	-4.72543×10^{-1}	TRUE
0.2	9.00048×10^{-3}	TRUE	-7.92123×10^{-1}	TRUE	-7.83123×10^{-1}	TRUE
0.3	3.38229×10^{-3}	TRUE	-7.17689×10^{-1}	TRUE	-7.14307×10^{-1}	TRUE
0.4	1.10787×10^{-3}	TRUE	-5.90976×10^{-1}	TRUE	-5.89868×10^{-1}	TRUE
0.5	3.21521×10^{-4}	TRUE	-4.51069×10^{-1}	TRUE	-4.50748×10^{-1}	TRUE
0.6	8.28689×10^{-5}	TRUE	-3.16869×10^{-1}	TRUE	-3.16787×10^{-1}	TRUE
0.7	1.89648×10^{-5}	TRUE	-2.03264×10^{-1}	TRUE	-2.03245×10^{-1}	TRUE
0.8	3.85132×10^{-6}	TRUE	-1.18291×10^{-1}	TRUE	-1.18287×10^{-1}	TRUE
0.9	6.93594×10^{-7}	TRUE	-6.21311×10^{-2}	TRUE	-6.21304×10^{-2}	TRUE
1.0	1.10712×10^{-7}	TRUE	-2.93335×10^{-2}	TRUE	-2.93334×10^{-2}	TRUE
1.1	1.56557×10^{-8}	TRUE	-1.24087×10^{-2}	TRUE	-1.24087×10^{-2}	TRUE
1.2	1.96047×10^{-9}	TRUE	-4.69133×10^{-3}	TRUE	-4.69133×10^{-3}	TRUE
1.3	2.17324×10^{-10}	TRUE	-1.58199×10^{-3}	TRUE	-1.58199×10^{-3}	TRUE
1.4	2.13201×10^{-11}	TRUE	-4.75062×10^{-4}	TRUE	-4.75062×10^{-4}	TRUE
1.5	1.85052×10^{-12}	TRUE	-1.26874×10^{-4}	TRUE	-1.26874×10^{-4}	TRUE
1.6	1.42109×10^{-13}	TRUE	-3.01032×10^{-5}	TRUE	-3.01032×10^{-5}	TRUE
1.7	9.65894×10^{-15}	TRUE	-6.34005×10^{-6}	TRUE	-6.34005×10^{-6}	TRUE
1.8	5.55112×10^{-16}	TRUE	-1.18441×10^{-6}	TRUE	-1.18441×10^{-6}	TRUE
1.9	0.00000×10^0	TRUE	-1.96146×10^{-7}	TRUE	-1.96146×10^{-7}	TRUE
2.0	0.00000×10^0	TRUE	-2.87809×10^{-8}	TRUE	-2.87809×10^{-8}	TRUE

C.1.6 Complete Tables for OCs of Rule 1 vs Rule 2: $n = 25$, $P^* = 0.99$

Table C.6: Inequality-check table for R_1 versus the CLT-based rule R_2 , $n = 25$, $P^* = 0.99$.

δ	$\Delta_{\text{PCS},2}$	Ineq _{PCS} holds	$\Delta_{\text{PICS},2}$	Ineq _{PICS} holds	$\Delta_{\text{ESS},2}$	Ineq _{ESS} holds
0.0	0.00000×10^0	TRUE	0.00000×10^0	TRUE	0.00000×10^0	TRUE
0.1	2.86134×10^{-3}	TRUE	-9.75654×10^{-2}	TRUE	-9.47041×10^{-2}	TRUE
0.2	1.14148×10^{-3}	TRUE	-7.78854×10^{-1}	TRUE	-7.77712×10^{-1}	TRUE
0.3	3.47766×10^{-4}	TRUE	-8.83360×10^{-1}	TRUE	-8.83013×10^{-1}	TRUE
0.4	9.13508×10^{-5}	TRUE	-8.18016×10^{-1}	TRUE	-8.17925×10^{-1}	TRUE
0.5	2.11519×10^{-5}	TRUE	-7.11683×10^{-1}	TRUE	-7.11662×10^{-1}	TRUE
0.6	4.33731×10^{-6}	TRUE	-5.81217×10^{-1}	TRUE	-5.81213×10^{-1}	TRUE
0.7	7.88252×10^{-7}	TRUE	-4.40963×10^{-1}	TRUE	-4.40962×10^{-1}	TRUE
0.8	1.26947×10^{-7}	TRUE	-3.07806×10^{-1}	TRUE	-3.07806×10^{-1}	TRUE
0.9	1.81112×10^{-8}	TRUE	-1.96101×10^{-1}	TRUE	-1.96100×10^{-1}	TRUE
1.0	2.28812×10^{-9}	TRUE	-1.13296×10^{-1}	TRUE	-1.13296×10^{-1}	TRUE
1.1	2.55902×10^{-10}	TRUE	-5.90569×10^{-2}	TRUE	-5.90569×10^{-2}	TRUE
1.2	2.53282×10^{-11}	TRUE	-2.76639×10^{-2}	TRUE	-2.76639×10^{-2}	TRUE
1.3	2.21789×10^{-12}	TRUE	-1.16085×10^{-2}	TRUE	-1.16085×10^{-2}	TRUE
1.4	1.71863×10^{-13}	TRUE	-4.35286×10^{-3}	TRUE	-4.35286×10^{-3}	TRUE
1.5	1.17684×10^{-14}	TRUE	-1.45564×10^{-3}	TRUE	-1.45564×10^{-3}	TRUE
1.6	6.66134×10^{-16}	TRUE	-4.33441×10^{-4}	TRUE	-4.33441×10^{-4}	TRUE
1.7	0.00000×10^0	TRUE	-1.14774×10^{-4}	TRUE	-1.14774×10^{-4}	TRUE
1.8	0.00000×10^0	TRUE	-2.69989×10^{-5}	TRUE	-2.69989×10^{-5}	TRUE
1.9	0.00000×10^0	TRUE	-5.63721×10^{-6}	TRUE	-5.63721×10^{-6}	TRUE
2.0	0.00000×10^0	TRUE	-1.04398×10^{-6}	TRUE	-1.04398×10^{-6}	TRUE

C.1.7 Complete Tables for OCs of Rule 1 vs Rule 3: $n = 25$, $P^* = 0.95$

Table C.7: Inequality-check table for R_1 versus the Gamma-based rule R_3 , $n = 25$, $P^* = 0.95$.

δ	$\Delta_{\text{PCS},3}$	Ineq _{PCS} holds	$\Delta_{\text{PICS},3}$	Ineq _{PICS} holds	$\Delta_{\text{ESS},3}$	Ineq _{ESS} holds
0.0	-6.54587×10^{-13}	TRUE	-6.54587×10^{-13}	TRUE	-1.30917×10^{-12}	TRUE
0.1	1.93725×10^{-2}	TRUE	-4.92347×10^{-1}	TRUE	-4.72975×10^{-1}	TRUE
0.2	9.76576×10^{-3}	TRUE	-7.94113×10^{-1}	TRUE	-7.84347×10^{-1}	TRUE
0.3	3.97508×10^{-3}	TRUE	-7.19612×10^{-1}	TRUE	-7.15637×10^{-1}	TRUE
0.4	1.46519×10^{-3}	TRUE	-5.91493×10^{-1}	TRUE	-5.90028×10^{-1}	TRUE
0.5	5.00167×10^{-4}	TRUE	-4.49483×10^{-1}	TRUE	-4.48983×10^{-1}	TRUE
0.6	1.59450×10^{-4}	TRUE	-3.13733×10^{-1}	TRUE	-3.13573×10^{-1}	TRUE
0.7	4.77318×10^{-5}	TRUE	-1.99973×10^{-1}	TRUE	-1.99926×10^{-1}	TRUE
0.8	1.34802×10^{-5}	TRUE	-1.16066×10^{-1}	TRUE	-1.16053×10^{-1}	TRUE
0.9	3.60692×10^{-6}	TRUE	-6.13324×10^{-2}	TRUE	-6.13288×10^{-2}	TRUE
1.0	9.17958×10^{-7}	TRUE	-2.95650×10^{-2}	TRUE	-2.95641×10^{-2}	TRUE
1.1	2.23002×10^{-7}	TRUE	-1.30458×10^{-2}	TRUE	-1.30456×10^{-2}	TRUE
1.2	5.18819×10^{-8}	TRUE	-5.29270×10^{-3}	TRUE	-5.29264×10^{-3}	TRUE
1.3	1.15943×10^{-8}	TRUE	-1.98394×10^{-3}	TRUE	-1.98393×10^{-3}	TRUE
1.4	2.49567×10^{-9}	TRUE	-6.90612×10^{-4}	TRUE	-6.90609×10^{-4}	TRUE
1.5	5.18707×10^{-10}	TRUE	-2.24382×10^{-4}	TRUE	-2.24382×10^{-4}	TRUE
1.6	1.04338×10^{-10}	TRUE	-6.83765×10^{-5}	TRUE	-6.83764×10^{-5}	TRUE
1.7	2.03539×10^{-11}	TRUE	-1.96332×10^{-5}	TRUE	-1.96332×10^{-5}	TRUE
1.8	3.85803×10^{-12}	TRUE	-5.33474×10^{-6}	TRUE	-5.33473×10^{-6}	TRUE
1.9	7.11653×10^{-13}	TRUE	-1.37721×10^{-6}	TRUE	-1.37721×10^{-6}	TRUE
2.0	1.28009×10^{-13}	TRUE	-3.39038×10^{-7}	TRUE	-3.39037×10^{-7}	TRUE

C.1.8 Complete Tables for OCs of Rule 1 vs Rule 3: $n = 25$, $P^* = 0.99$

Table C.8: Inequality-check table for R_1 versus the Gamma-based rule R_3 , $n = 25$, $P^* = 0.99$.

δ	$\Delta_{\text{PCS},3}$	Ineq _{PCS} holds	$\Delta_{\text{PICS},3}$	Ineq _{PICS} holds	$\Delta_{\text{ESS},3}$	Ineq _{ESS} holds
0.0	-1.17684×10^{-14}	TRUE	-1.17684×10^{-14}	TRUE	-2.35367×10^{-14}	TRUE
0.1	3.13755×10^{-3}	TRUE	-9.85640×10^{-2}	TRUE	-9.54264×10^{-2}	TRUE
0.2	1.38257×10^{-3}	TRUE	-7.81960×10^{-1}	TRUE	-7.80577×10^{-1}	TRUE
0.3	4.88454×10^{-4}	TRUE	-8.89637×10^{-1}	TRUE	-8.89149×10^{-1}	TRUE
0.4	1.56864×10^{-4}	TRUE	-8.27707×10^{-1}	TRUE	-8.27550×10^{-1}	TRUE
0.5	4.70201×10^{-5}	TRUE	-7.23663×10^{-1}	TRUE	-7.23616×10^{-1}	TRUE
0.6	1.32779×10^{-5}	TRUE	-5.93318×10^{-1}	TRUE	-5.93305×10^{-1}	TRUE
0.7	3.55120×10^{-6}	TRUE	-4.51139×10^{-1}	TRUE	-4.51136×10^{-1}	TRUE
0.8	9.03316×10^{-7}	TRUE	-3.15198×10^{-1}	TRUE	-3.15197×10^{-1}	TRUE
0.9	2.19335×10^{-7}	TRUE	-2.01121×10^{-1}	TRUE	-2.01121×10^{-1}	TRUE
1.0	5.10044×10^{-8}	TRUE	-1.16859×10^{-1}	TRUE	-1.16859×10^{-1}	TRUE
1.1	1.13932×10^{-8}	TRUE	-6.18179×10^{-2}	TRUE	-6.18179×10^{-2}	TRUE
1.2	2.45134×10^{-9}	TRUE	-2.98301×10^{-2}	TRUE	-2.98301×10^{-2}	TRUE
1.3	5.09296×10^{-10}	TRUE	-1.31759×10^{-2}	TRUE	-1.31759×10^{-2}	TRUE
1.4	1.02408×10^{-10}	TRUE	-5.35050×10^{-3}	TRUE	-5.35050×10^{-3}	TRUE
1.5	1.99707×10^{-11}	TRUE	-2.00738×10^{-3}	TRUE	-2.00738×10^{-3}	TRUE
1.6	3.78408×10^{-12}	TRUE	-6.99348×10^{-4}	TRUE	-6.99348×10^{-4}	TRUE
1.7	6.97886×10^{-13}	TRUE	-2.27395×10^{-4}	TRUE	-2.27395×10^{-4}	TRUE
1.8	1.25455×10^{-13}	TRUE	-6.93439×10^{-5}	TRUE	-6.93439×10^{-5}	TRUE
1.9	2.19824×10^{-14}	TRUE	-1.99242×10^{-5}	TRUE	-1.99242×10^{-5}	TRUE
2.0	3.66374×10^{-15}	TRUE	-5.41711×10^{-6}	TRUE	-5.41711×10^{-6}	TRUE

C.1.9 Complete Tables for OCs of Rule 1 vs Rule 2: $n = 100$, $P^* = 0.95$

Table C.9: Inequality-check table for R_1 versus the CLT-based rule R_2 , $n = 100$, $P^* = 0.95$.

δ	$\Delta_{\text{PCS},2}$	Ineq _{PCS} holds	$\Delta_{\text{PICS},2}$	Ineq _{PICS} holds	$\Delta_{\text{ESS},2}$	Ineq _{ESS} holds
0.0	1.11022×10^{-16}	TRUE	1.11022×10^{-16}	TRUE	2.22045×10^{-16}	TRUE
0.1	9.33511×10^{-3}	TRUE	-8.25586×10^{-1}	TRUE	-8.16251×10^{-1}	TRUE
0.2	1.11014×10^{-3}	TRUE	-5.91203×10^{-1}	TRUE	-5.90093×10^{-1}	TRUE
0.3	8.28842×10^{-5}	TRUE	-3.16871×10^{-1}	TRUE	-3.16788×10^{-1}	TRUE
0.4	3.85143×10^{-6}	TRUE	-1.18291×10^{-1}	TRUE	-1.18287×10^{-1}	TRUE
0.5	1.10713×10^{-7}	TRUE	-2.93335×10^{-2}	TRUE	-2.93334×10^{-2}	TRUE
0.6	1.96047×10^{-9}	TRUE	-4.69133×10^{-3}	TRUE	-4.69133×10^{-3}	TRUE
0.7	2.13201×10^{-11}	TRUE	-4.75062×10^{-4}	TRUE	-4.75062×10^{-4}	TRUE
0.8	1.42109×10^{-13}	TRUE	-3.01032×10^{-5}	TRUE	-3.01032×10^{-5}	TRUE
0.9	5.55112×10^{-16}	TRUE	-1.18441×10^{-6}	TRUE	-1.18441×10^{-6}	TRUE
1.0	0.00000×10^0	TRUE	-2.87809×10^{-8}	TRUE	-2.87809×10^{-8}	TRUE
1.1	0.00000×10^0	TRUE	-4.30316×10^{-10}	TRUE	-4.30316×10^{-10}	TRUE
1.2	0.00000×10^0	TRUE	-3.94785×10^{-12}	TRUE	-3.94795×10^{-12}	TRUE
1.3	0.00000×10^0	TRUE	-2.21787×10^{-14}	TRUE	-2.22045×10^{-14}	TRUE
1.4	0.00000×10^0	TRUE	-7.61796×10^{-17}	TRUE	0.00000×10^0	TRUE
1.5	0.00000×10^0	TRUE	-1.59786×10^{-19}	TRUE	0.00000×10^0	TRUE
1.6	0.00000×10^0	TRUE	-2.04464×10^{-22}	TRUE	0.00000×10^0	TRUE
1.7	0.00000×10^0	TRUE	-1.59492×10^{-25}	TRUE	0.00000×10^0	TRUE
1.8	0.00000×10^0	TRUE	-7.57934×10^{-29}	TRUE	0.00000×10^0	TRUE
1.9	0.00000×10^0	TRUE	-2.19314×10^{-32}	TRUE	0.00000×10^0	TRUE
2.0	0.00000×10^0	TRUE	-3.86237×10^{-36}	TRUE	0.00000×10^0	TRUE

C.1.10 Complete Tables for OCs of Rule 1 vs Rule 2: $n = 100$, $P^* = 0.99$

Table C.10: Inequality-check table for R_1 versus the CLT-based rule R_2 , $n = 100$, $P^* = 0.99$.

δ	$\Delta_{\text{PCS},2}$	Ineq _{PCS} holds	$\Delta_{\text{PICS},2}$	Ineq _{PICS} holds	$\Delta_{\text{ESS},2}$	Ineq _{ESS} holds
0.0	0.00000×10^0	TRUE	0.00000×10^0	TRUE	0.00000×10^0	TRUE
0.1	1.20840×10^{-3}	TRUE	-9.46167×10^{-1}	TRUE	-9.44959×10^{-1}	TRUE
0.2	9.18048×10^{-5}	TRUE	-8.19151×10^{-1}	TRUE	-8.19059×10^{-1}	TRUE
0.3	4.34037×10^{-6}	TRUE	-5.81225×10^{-1}	TRUE	-5.81220×10^{-1}	TRUE
0.4	1.26968×10^{-7}	TRUE	-3.07806×10^{-1}	TRUE	-3.07806×10^{-1}	TRUE
0.5	2.28825×10^{-9}	TRUE	-1.13296×10^{-1}	TRUE	-1.13296×10^{-1}	TRUE
0.6	2.53291×10^{-11}	TRUE	-2.76639×10^{-2}	TRUE	-2.76639×10^{-2}	TRUE
0.7	1.71863×10^{-13}	TRUE	-4.35286×10^{-3}	TRUE	-4.35286×10^{-3}	TRUE
0.8	6.66134×10^{-16}	TRUE	-4.33441×10^{-4}	TRUE	-4.33441×10^{-4}	TRUE
0.9	0.00000×10^0	TRUE	-2.69989×10^{-5}	TRUE	-2.69989×10^{-5}	TRUE
1.0	0.00000×10^0	TRUE	-1.04398×10^{-6}	TRUE	-1.04398×10^{-6}	TRUE
1.1	0.00000×10^0	TRUE	-2.49275×10^{-8}	TRUE	-2.49275×10^{-8}	TRUE
1.2	0.00000×10^0	TRUE	-3.66182×10^{-10}	TRUE	-3.66182×10^{-10}	TRUE
1.3	0.00000×10^0	TRUE	-3.30042×10^{-12}	TRUE	-3.30047×10^{-12}	TRUE
1.4	0.00000×10^0	TRUE	-1.82145×10^{-14}	TRUE	-1.82077×10^{-14}	TRUE
1.5	0.00000×10^0	TRUE	-6.14565×10^{-17}	TRUE	0.00000×10^0	TRUE
1.6	0.00000×10^0	TRUE	-1.26619×10^{-19}	TRUE	0.00000×10^0	TRUE
1.7	0.00000×10^0	TRUE	-1.59147×10^{-22}	TRUE	0.00000×10^0	TRUE
1.8	0.00000×10^0	TRUE	-1.21935×10^{-25}	TRUE	0.00000×10^0	TRUE
1.9	0.00000×10^0	TRUE	-5.69142×10^{-29}	TRUE	0.00000×10^0	TRUE
2.0	0.00000×10^0	TRUE	-1.61751×10^{-32}	TRUE	0.00000×10^0	TRUE

C.1.11 Complete Tables for OCs of Rule 1 vs Rule 3: $n = 100$, $P^* = 0.95$

Table C.11: Inequality-check table for R_1 versus the Gamma-based rule R_3 , $n = 100$, $P^* = 0.95$, using the Gamma-quantile upper integration limit.

δ	$\Delta_{\text{PCS},3}$	Ineq _{PCS} holds	$\Delta_{\text{PICS},3}$	Ineq _{PICS} holds	$\Delta_{\text{ESS},3}$	Ineq _{ESS} holds
0.0	-1.09279×10^{-12}	TRUE	-1.09279×10^{-12}	TRUE	-2.18559×10^{-12}	TRUE
0.1	9.53438×10^{-3}	TRUE	-8.26073×10^{-1}	TRUE	-8.16539×10^{-1}	TRUE
0.2	1.20168×10^{-3}	TRUE	-5.91307×10^{-1}	TRUE	-5.90105×10^{-1}	TRUE
0.3	1.00963×10^{-4}	TRUE	-3.16072×10^{-1}	TRUE	-3.15971×10^{-1}	TRUE
0.4	5.75140×10^{-6}	TRUE	-1.17711×10^{-1}	TRUE	-1.17705×10^{-1}	TRUE
0.5	2.26827×10^{-7}	TRUE	-2.93859×10^{-2}	TRUE	-2.93857×10^{-2}	TRUE
0.6	6.33741×10^{-9}	TRUE	-4.84732×10^{-3}	TRUE	-4.84731×10^{-3}	TRUE
0.7	1.29341×10^{-10}	TRUE	-5.29360×10^{-4}	TRUE	-5.29360×10^{-4}	TRUE
0.8	2.92877×10^{-12}	TRUE	-3.87466×10^{-5}	TRUE	-3.87466×10^{-5}	TRUE
0.9	1.02240×10^{-12}	TRUE	-1.93630×10^{-6}	TRUE	-1.93630×10^{-6}	TRUE
1.0	1.00064×10^{-12}	TRUE	-6.75184×10^{-8}	TRUE	-6.75174×10^{-8}	TRUE
1.1	1.00031×10^{-12}	TRUE	-1.68073×10^{-9}	TRUE	-1.67973×10^{-9}	TRUE
1.2	1.00031×10^{-12}	TRUE	-3.03662×10^{-11}	TRUE	-2.93658×10^{-11}	TRUE
1.3	1.00031×10^{-12}	TRUE	-3.78129×10^{-13}	TRUE	6.22169×10^{-13}	TRUE
1.4	1.00031×10^{-12}	TRUE	-2.32804×10^{-15}	TRUE	9.97979×10^{-13}	TRUE
1.5	1.00031×10^{-12}	TRUE	-3.39031×10^{-18}	TRUE	1.00031×10^{-12}	TRUE
1.6	1.00031×10^{-12}	TRUE	-4.25315×10^{-22}	TRUE	1.00031×10^{-12}	TRUE
1.7	1.00031×10^{-12}	TRUE	-1.13842×10^{-27}	TRUE	1.00031×10^{-12}	TRUE
1.8	1.00031×10^{-12}	TRUE	-5.96186×10^{-36}	TRUE	1.00031×10^{-12}	TRUE
1.9	1.00031×10^{-12}	TRUE	-3.51504×10^{-49}	TRUE	1.00031×10^{-12}	TRUE
2.0	1.00031×10^{-12}	TRUE	-2.18213×10^{-74}	TRUE	1.00031×10^{-12}	TRUE

C.1.12 Complete Tables for OCs of Rule 1 vs Rule 3: $n = 100$, $P^* = 0.99$

Table C.12: Inequality-check table for R_1 versus the Gamma-based rule R_3 , $n = 100$, $P^* = 0.99$.

δ	$\Delta_{\text{PCS},3}$	Ineq _{PCS} holds	$\Delta_{\text{PICS},3}$	Ineq _{PICS} holds	$\Delta_{\text{ESS},3}$	Ineq _{ESS} holds
0.0	-5.33240×10^{-13}	TRUE	-5.33240×10^{-13}	TRUE	-1.06648×10^{-12}	TRUE
0.1	1.27291×10^{-3}	TRUE	-9.46989×10^{-1}	TRUE	-9.45716×10^{-1}	TRUE
0.2	1.08078×10^{-4}	TRUE	-8.21630×10^{-1}	TRUE	-8.21522×10^{-1}	TRUE
0.3	6.21652×10^{-6}	TRUE	-5.84259×10^{-1}	TRUE	-5.84253×10^{-1}	TRUE
0.4	2.47411×10^{-7}	TRUE	-3.09664×10^{-1}	TRUE	-3.09664×10^{-1}	TRUE
0.5	6.97051×10^{-9}	TRUE	-1.14195×10^{-1}	TRUE	-1.14195×10^{-1}	TRUE
0.6	1.42306×10^{-10}	TRUE	-2.82110×10^{-2}	TRUE	-2.82110×10^{-2}	TRUE
0.7	2.15428×10^{-12}	TRUE	-7.63190×10^{-16}	TRUE	2.15350×10^{-12}	TRUE
0.8	2.50910×10^{-14}	TRUE	-7.55554×10^{-24}	TRUE	2.50910×10^{-14}	TRUE
0.9	4.44089×10^{-16}	TRUE	-9.25503×10^{-37}	TRUE	4.44089×10^{-16}	TRUE
1.0	2.22045×10^{-16}	TRUE	9.30019×10^{-43}	TRUE	2.22045×10^{-16}	TRUE
1.1	2.22045×10^{-16}	TRUE	4.22228×10^{-47}	TRUE	2.22045×10^{-16}	TRUE
1.2	2.22045×10^{-16}	TRUE	1.91691×10^{-51}	TRUE	2.22045×10^{-16}	TRUE
1.3	2.22045×10^{-16}	TRUE	8.70277×10^{-56}	TRUE	2.22045×10^{-16}	TRUE
1.4	2.22045×10^{-16}	TRUE	3.95105×10^{-60}	TRUE	2.22045×10^{-16}	TRUE
1.5	2.22045×10^{-16}	TRUE	1.79377×10^{-64}	TRUE	2.22045×10^{-16}	TRUE
1.6	2.22045×10^{-16}	TRUE	8.14372×10^{-69}	TRUE	2.22045×10^{-16}	TRUE
1.7	2.22045×10^{-16}	TRUE	3.69724×10^{-73}	TRUE	2.22045×10^{-16}	TRUE
1.8	2.22045×10^{-16}	TRUE	1.67855×10^{-77}	TRUE	2.22045×10^{-16}	TRUE
1.9	2.22045×10^{-16}	TRUE	7.62059×10^{-82}	TRUE	2.22045×10^{-16}	TRUE
2.0	2.22045×10^{-16}	TRUE	3.45974×10^{-86}	TRUE	2.22045×10^{-16}	TRUE

C.2 Heat Maps for OCs of Rule 1, Rule 2, and Rule 3 when $k = 2$

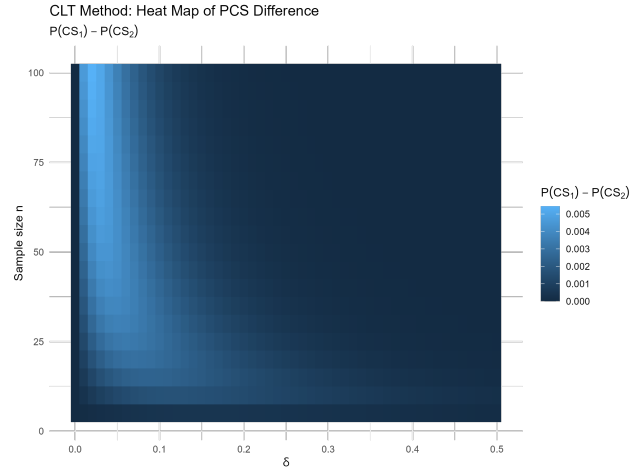


Figure C.1: Heat Map of $\Delta_{PCS,2}(\delta)$ for R_1 versus the CLT-based rule R_2 , with $P^* = 0.99$.

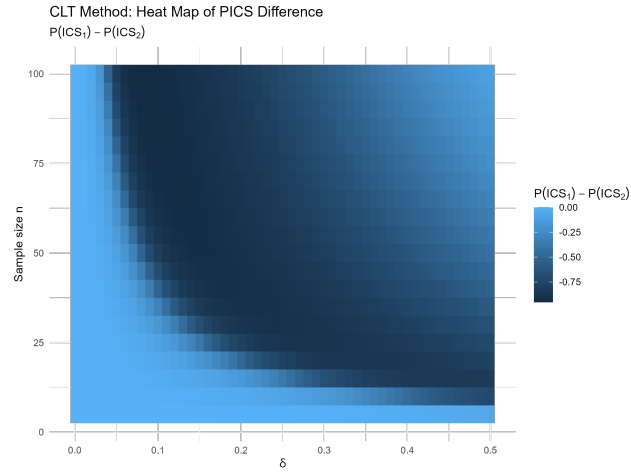


Figure C.2: Heat Map of $\Delta_{PICS,2}(\delta)$ for R_1 versus the CLT-based rule R_2 , with $P^* = 0.99$.

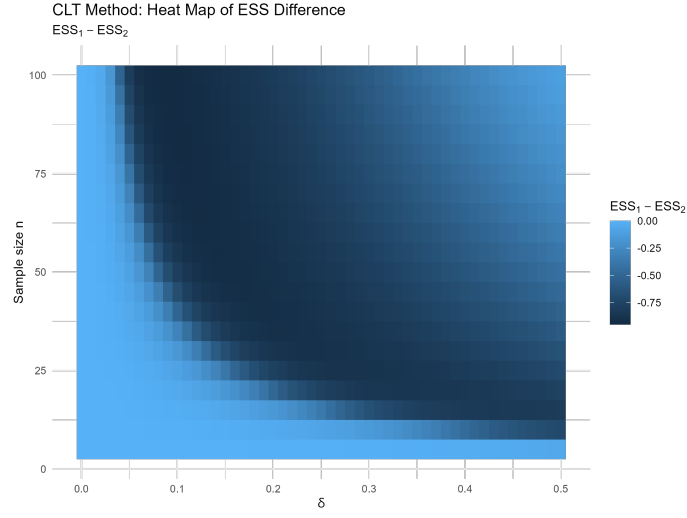


Figure C.3: Heat Map of $\Delta_{ESS,2}(\delta)$ for R_1 versus the CLT-based rule R_2 , with $P^* = 0.99$.

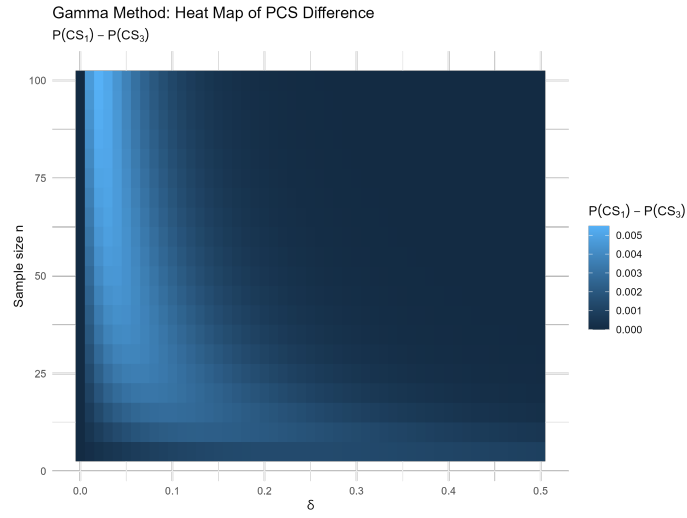


Figure C.4: Heat Map of $\Delta_{PCS,3}(\delta)$ for R_1 versus the Gamma-based rule R_3 , with $P^* = 0.99$.

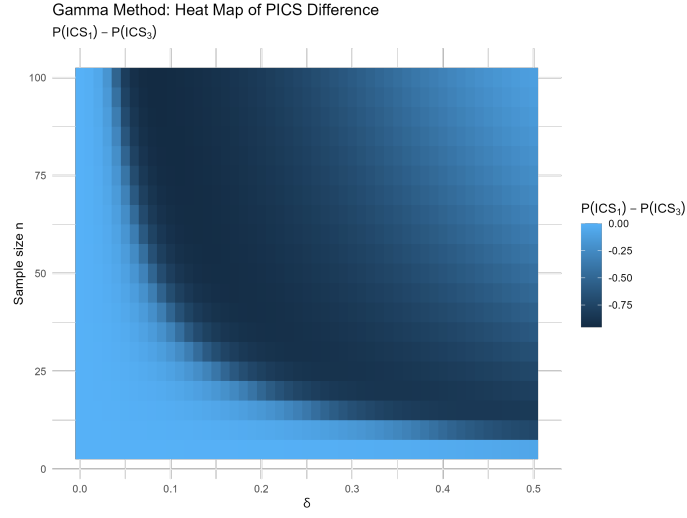


Figure C.5: Heat Map of $\Delta_{\text{PICS},3}(\delta)$ for R_1 versus the Gamma-based rule R_3 , with $P^* = 0.99$.

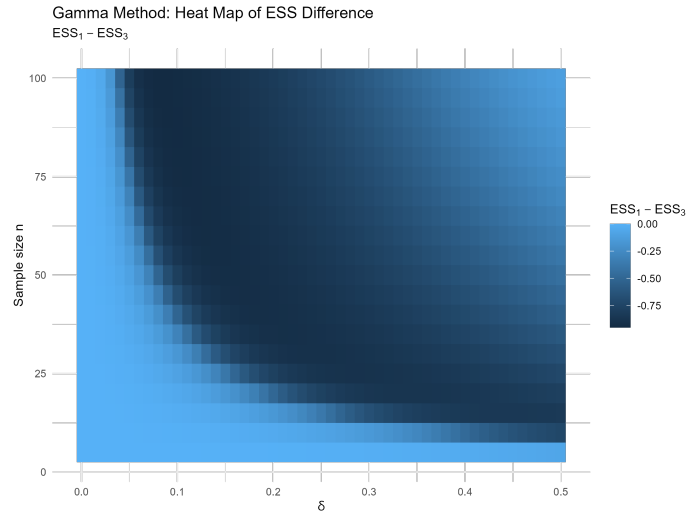


Figure C.6: Heat Map of $\Delta_{\text{ESS},3}(\delta)$ for R_1 versus the Gamma-based rule R_3 , with $P^* = 0.99$.

C.3 Additional Fully Simulated Weibull Study Trace Plots

C.3.1 Additional Random-Walk Trace Plots for the Fully Simulated Weibull Study

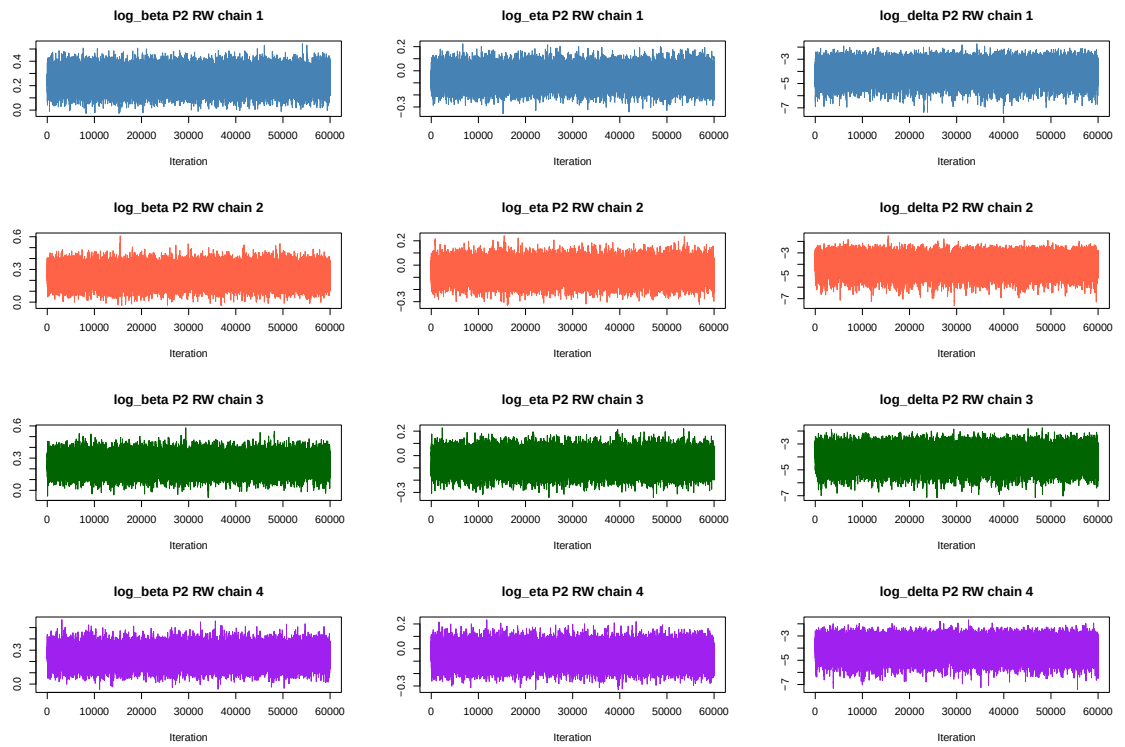


Figure C.7: Random-walk trace plots for Population P2

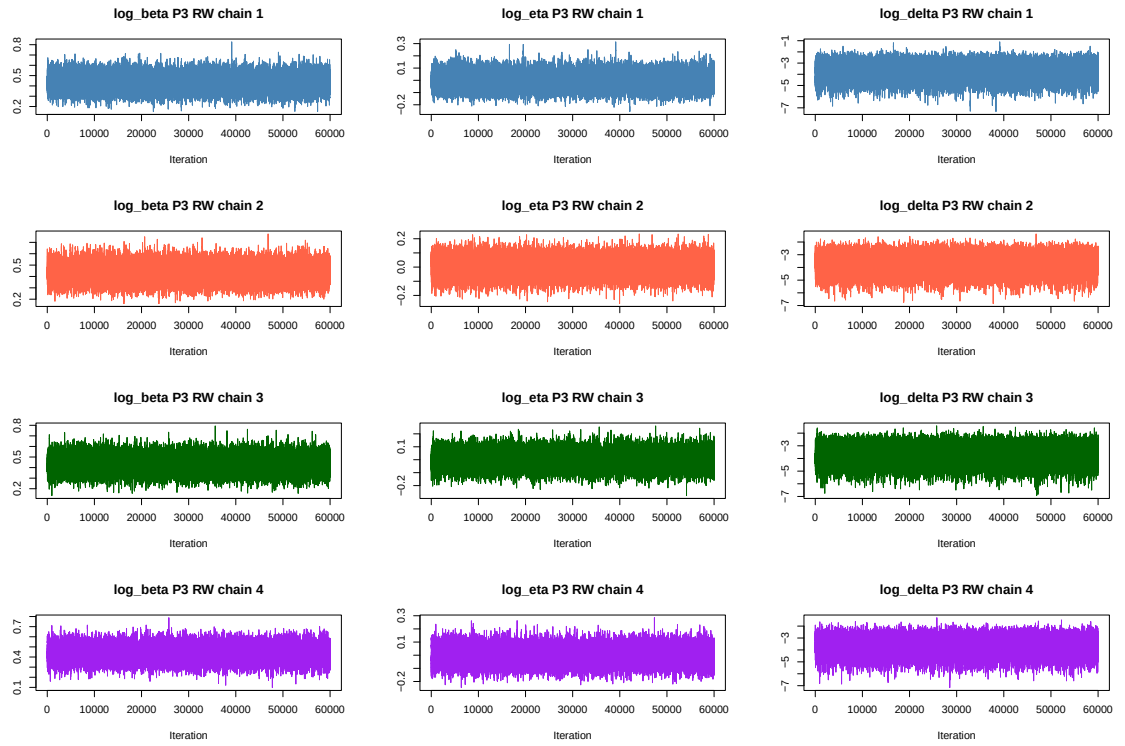


Figure C.8: Random-walk trace plots for Population P3

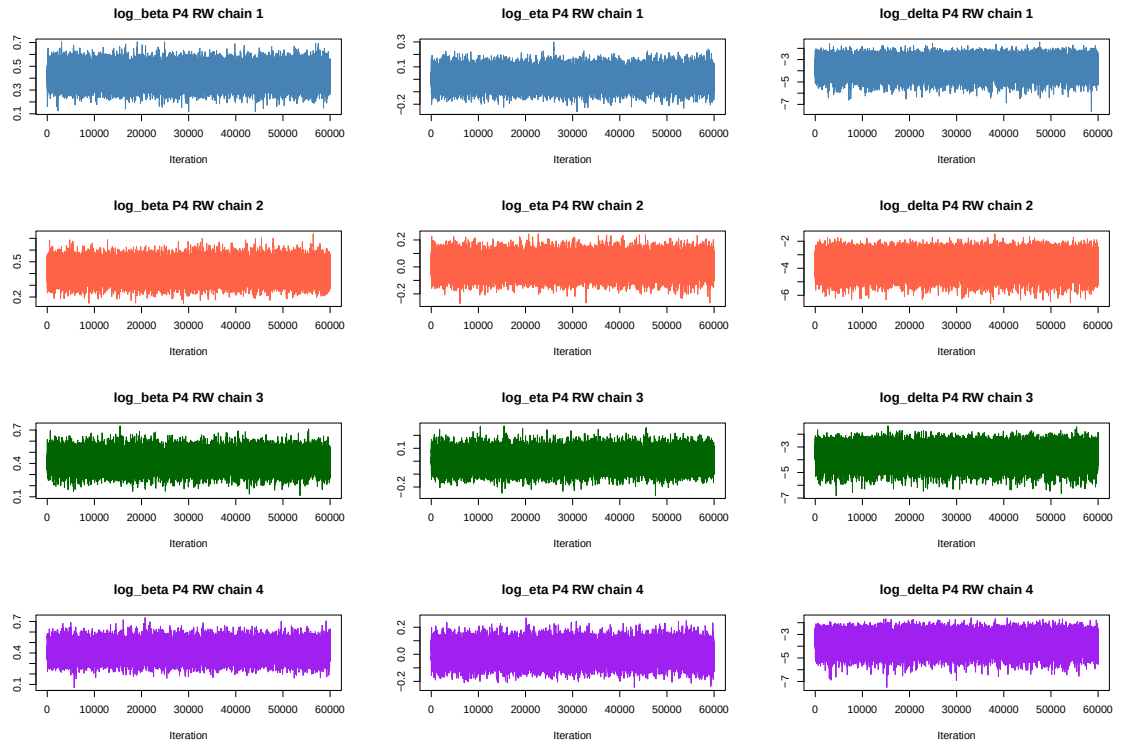


Figure C.9: Random-walk trace plots for Population P4

C.3.2 Additional Independence-Sampler Trace Plots for the Fully Simulated Weibull Study

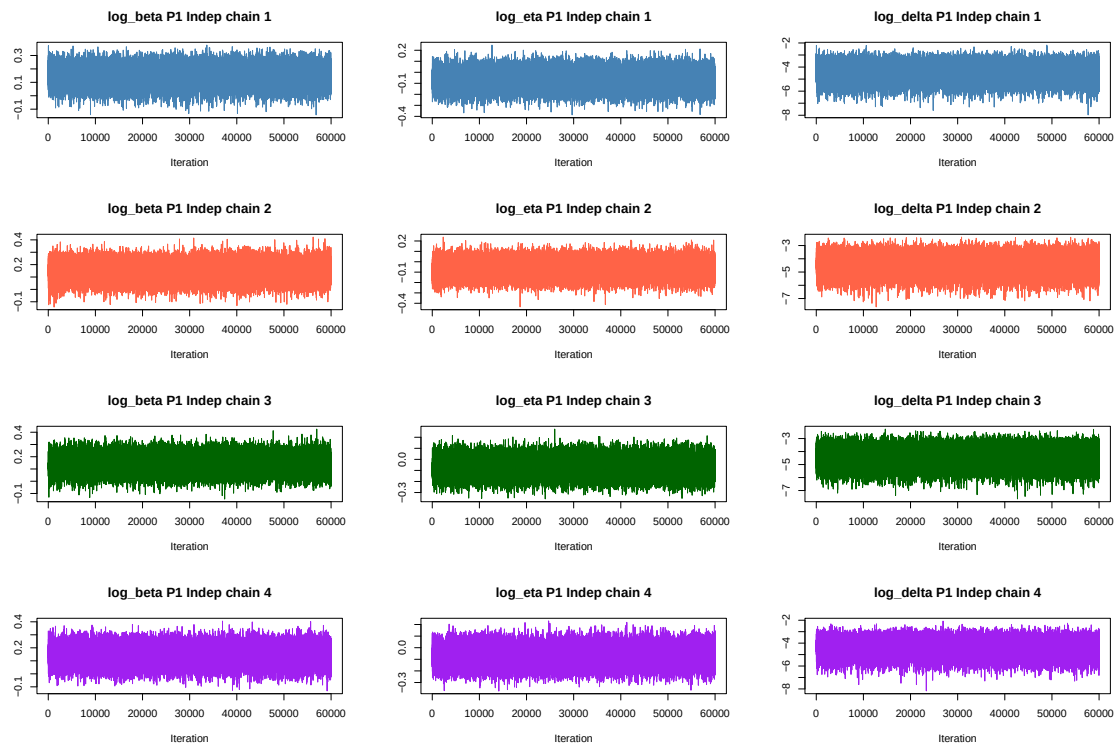


Figure C.10: Independence-sampler trace plots for Population P1

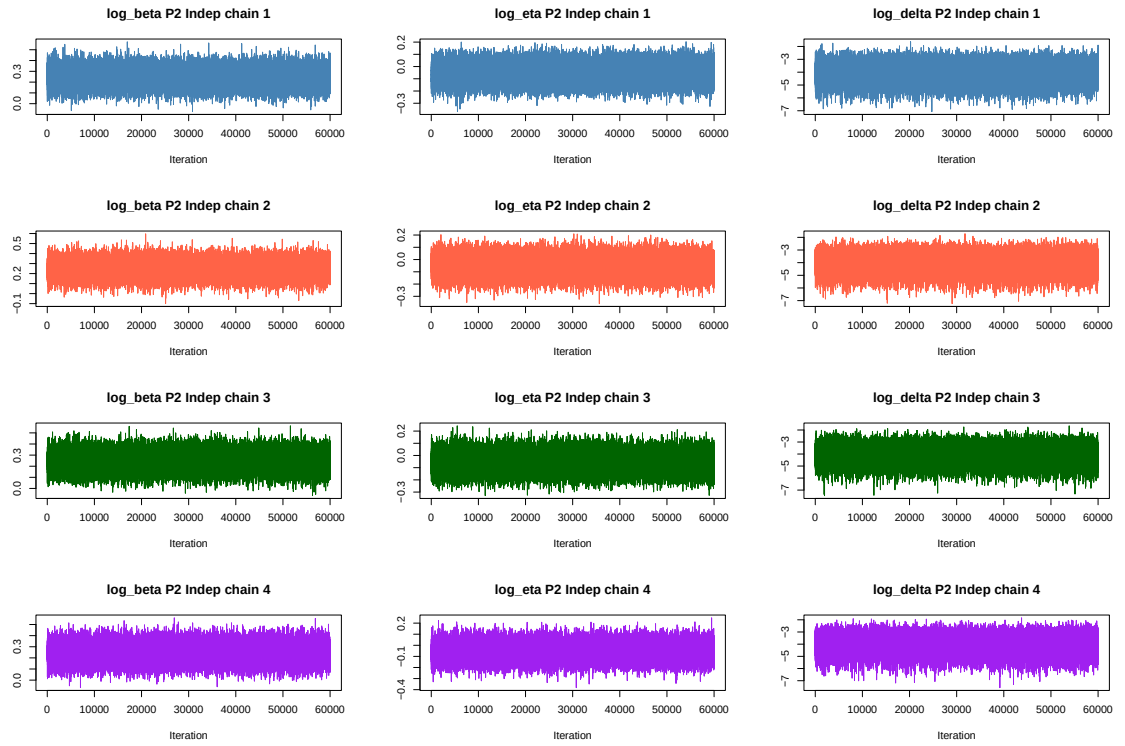


Figure C.11: Independence-sampler trace plots for Population P2

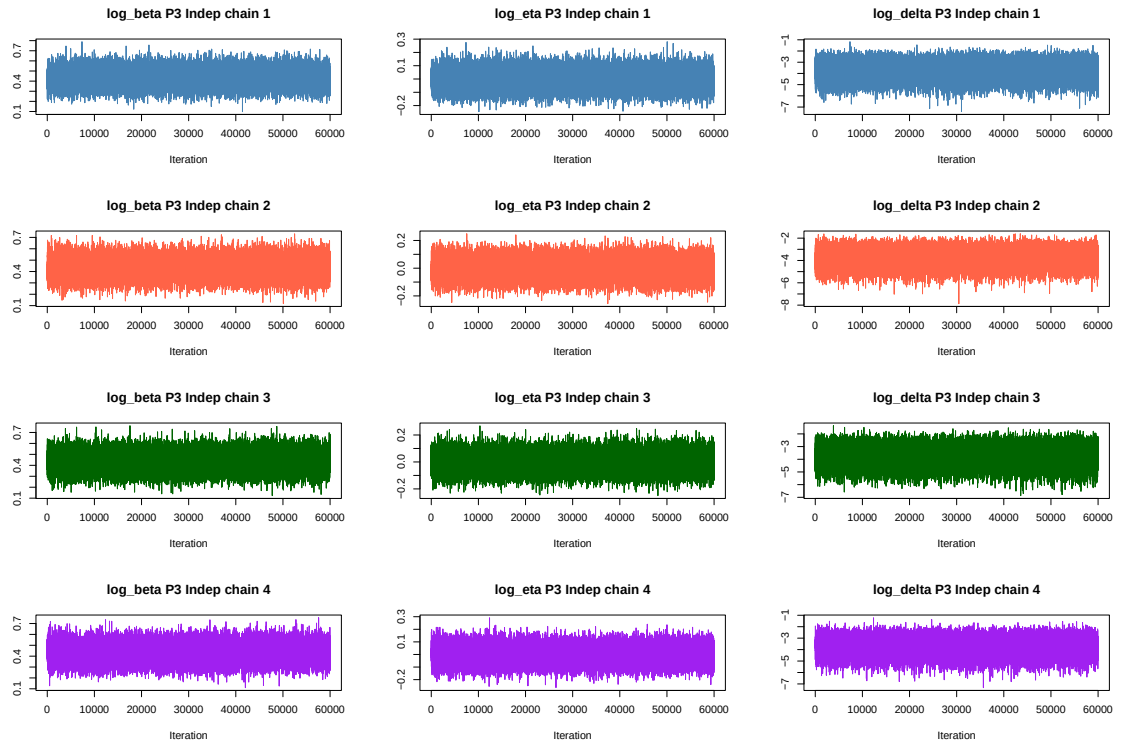


Figure C.12: Independence-sampler trace plots for Population P3

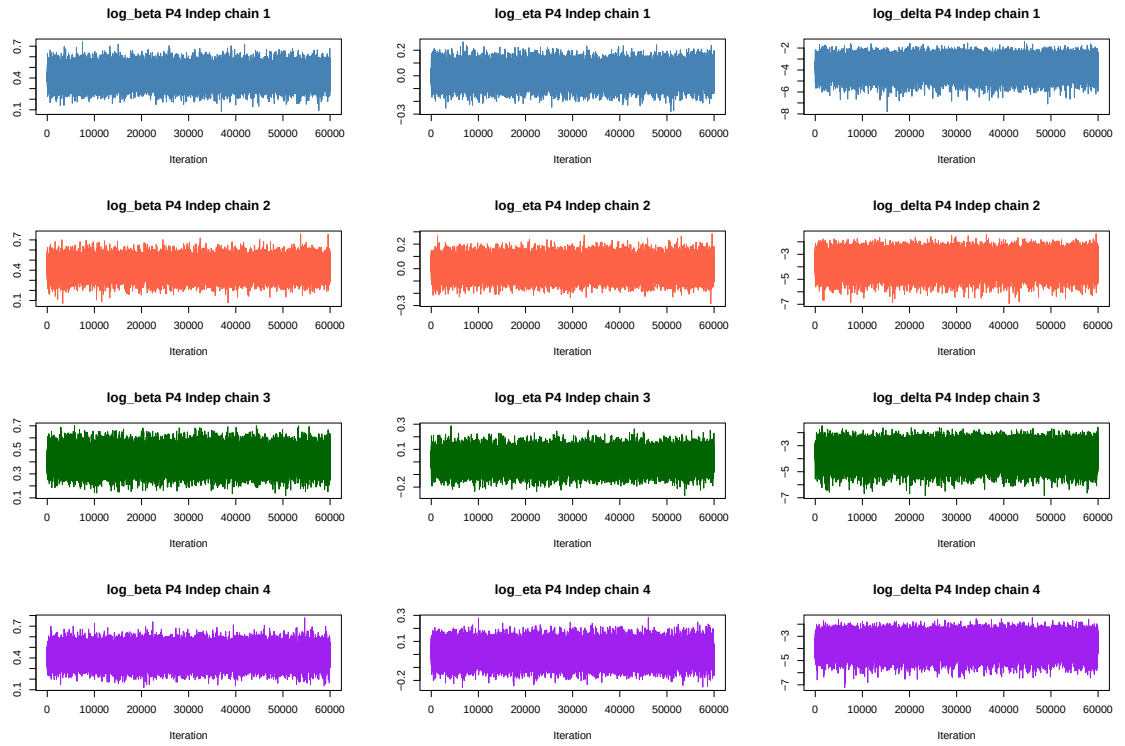


Figure C.13: Independence-sampler trace plots for Population P4

C.4 Additional Fully Simulated Study Autocorrelation Functions

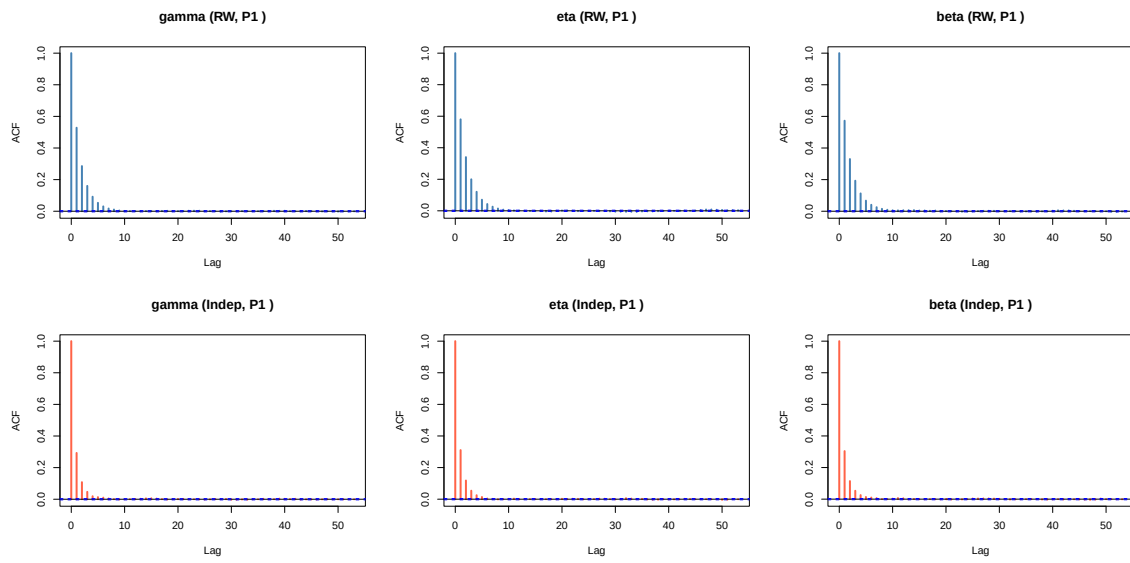


Figure C.14: Autocorrelation functions for γ , η , and β for Population P1

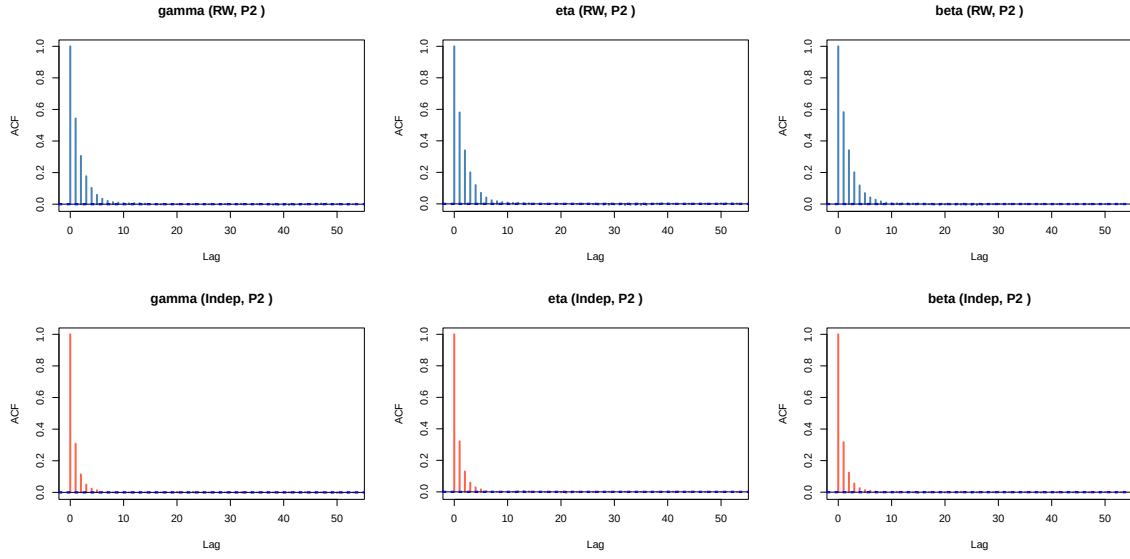


Figure C.15: Autocorrelation functions for γ , η , and β for Population P2

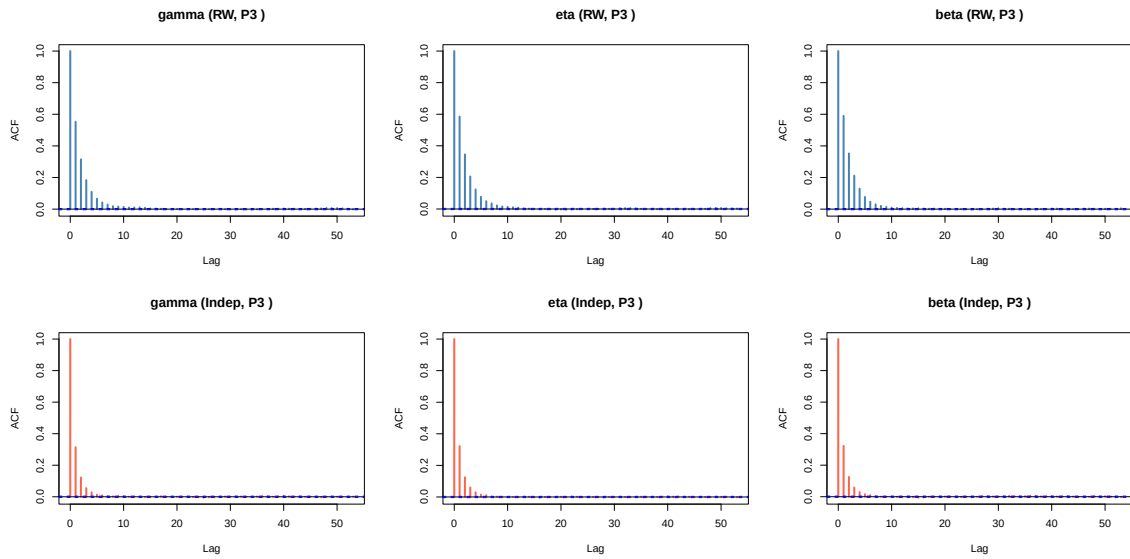


Figure C.16: Autocorrelation functions for γ , η , and β for Population P3

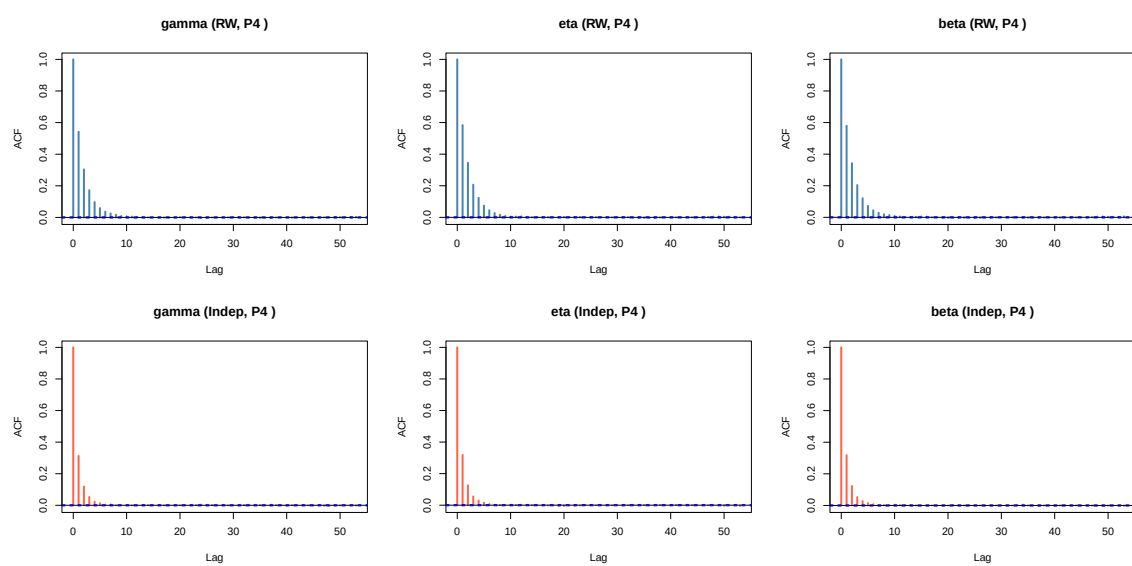


Figure C.17: Autocorrelation functions for γ , η , and β for Population P4

C.5 Additional Fully Simulated Study Posterior Predictive Check Plots

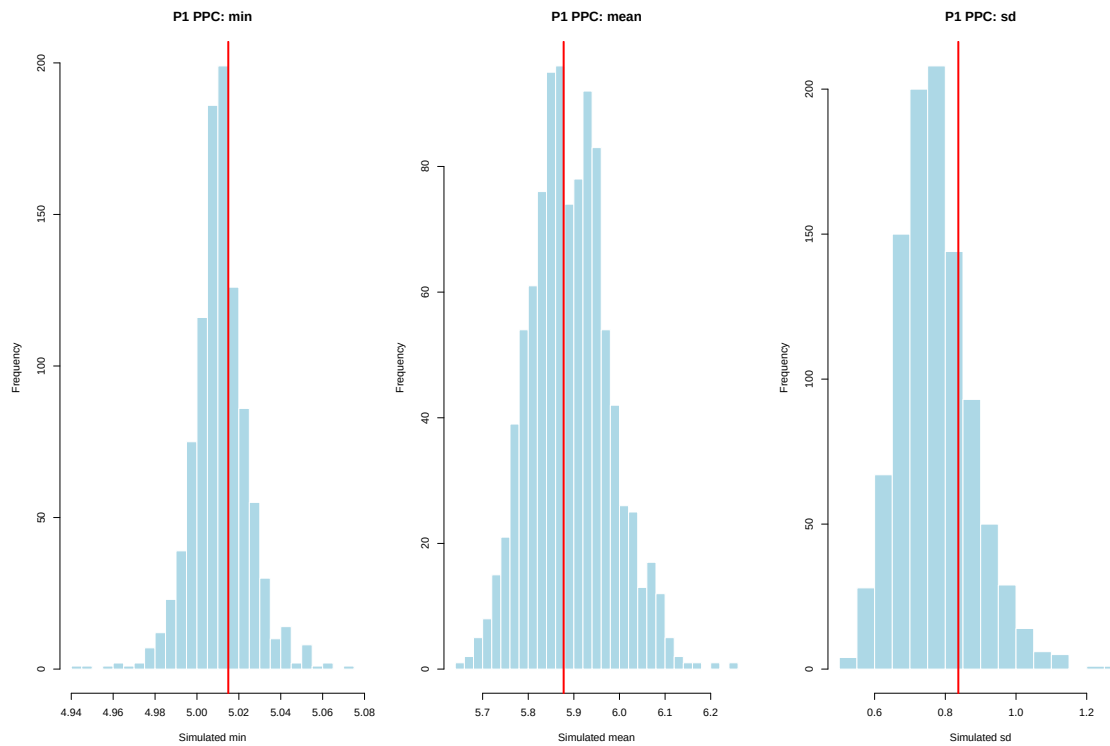


Figure C.18: Posterior predictive checks for Population P1

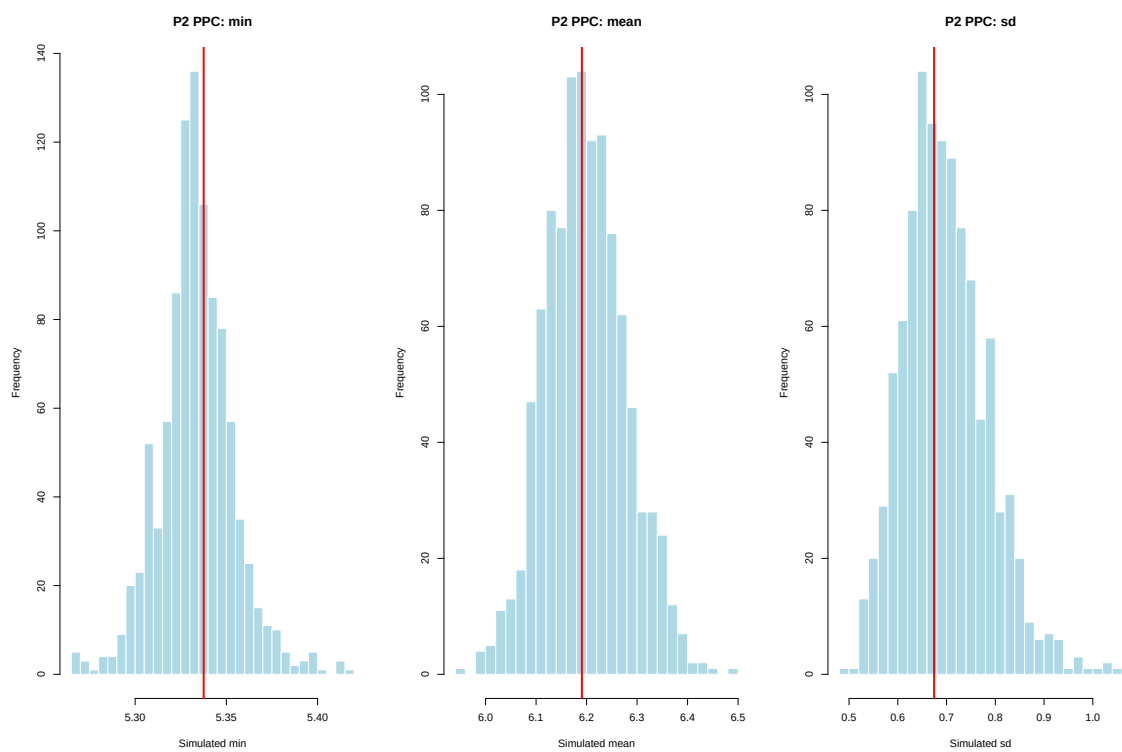


Figure C.19: Posterior predictive checks for Population P2

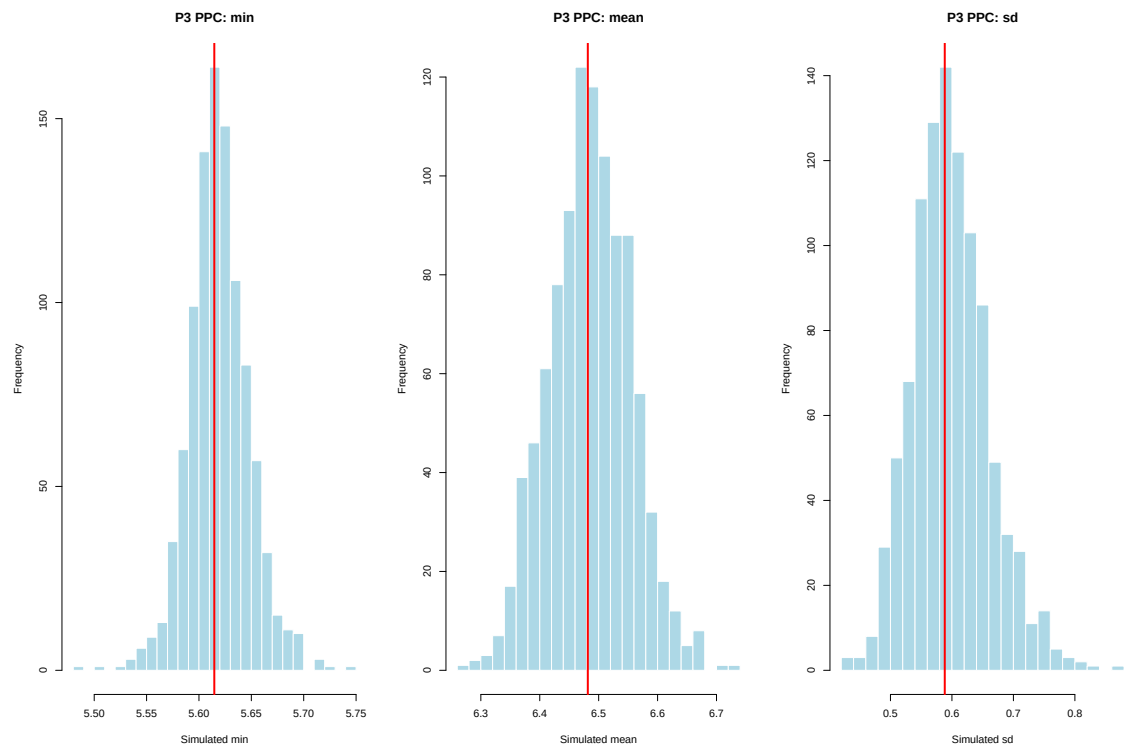


Figure C.20: Posterior predictive checks for Population P3

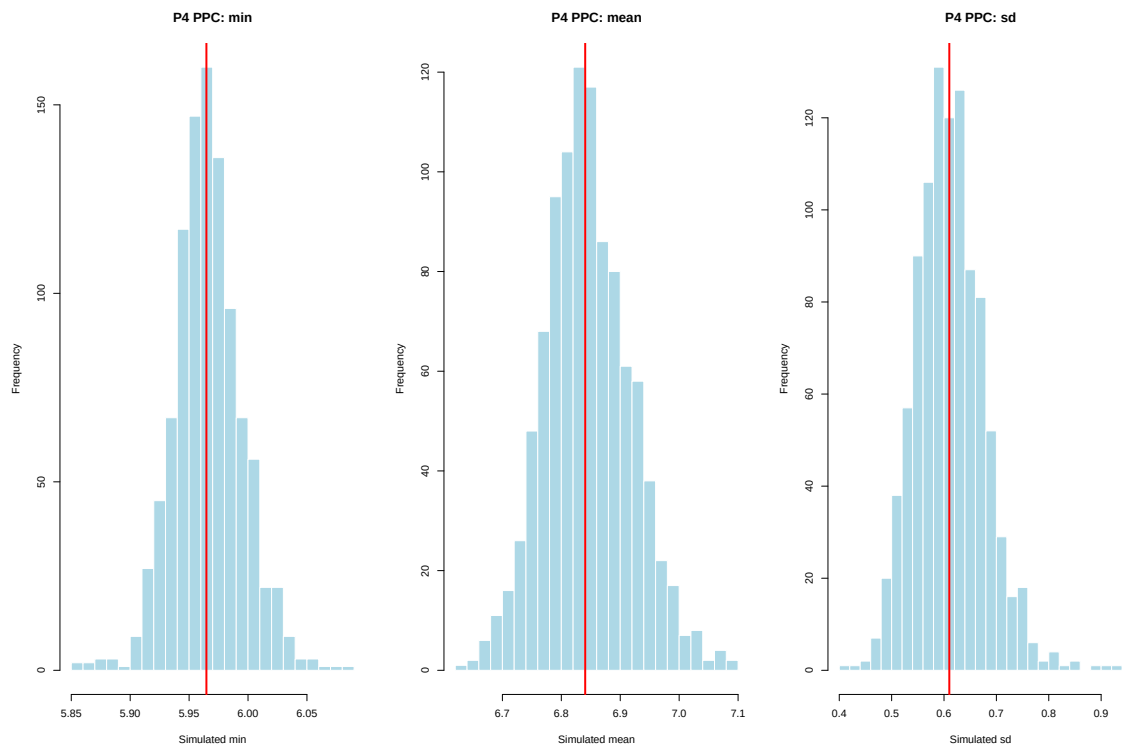


Figure C.21: Posterior predictive checks for Population P4

C.6 Additional Semi-Simulated Alloy Study Trace Plots

C.6.1 Additional Random-Walk Trace Plots for the Semi-Simulated Alloy Study

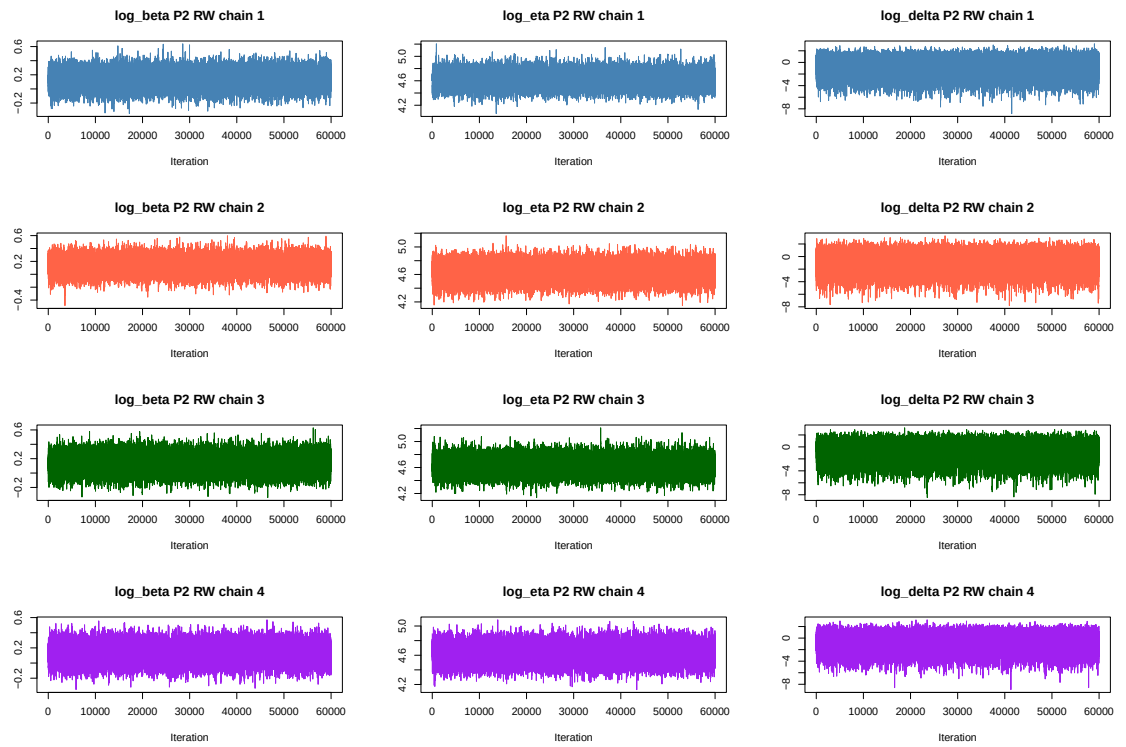


Figure C.22: Random-walk trace plots for Population P2

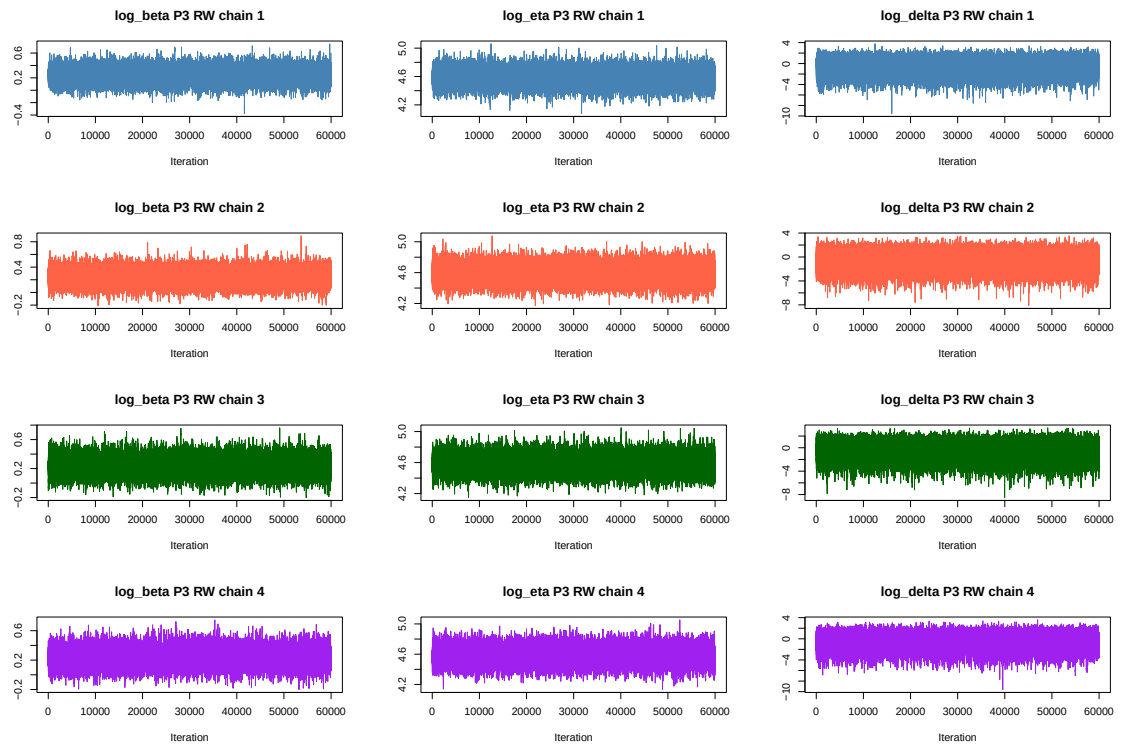


Figure C.23: Random-walk trace plots for Population P3

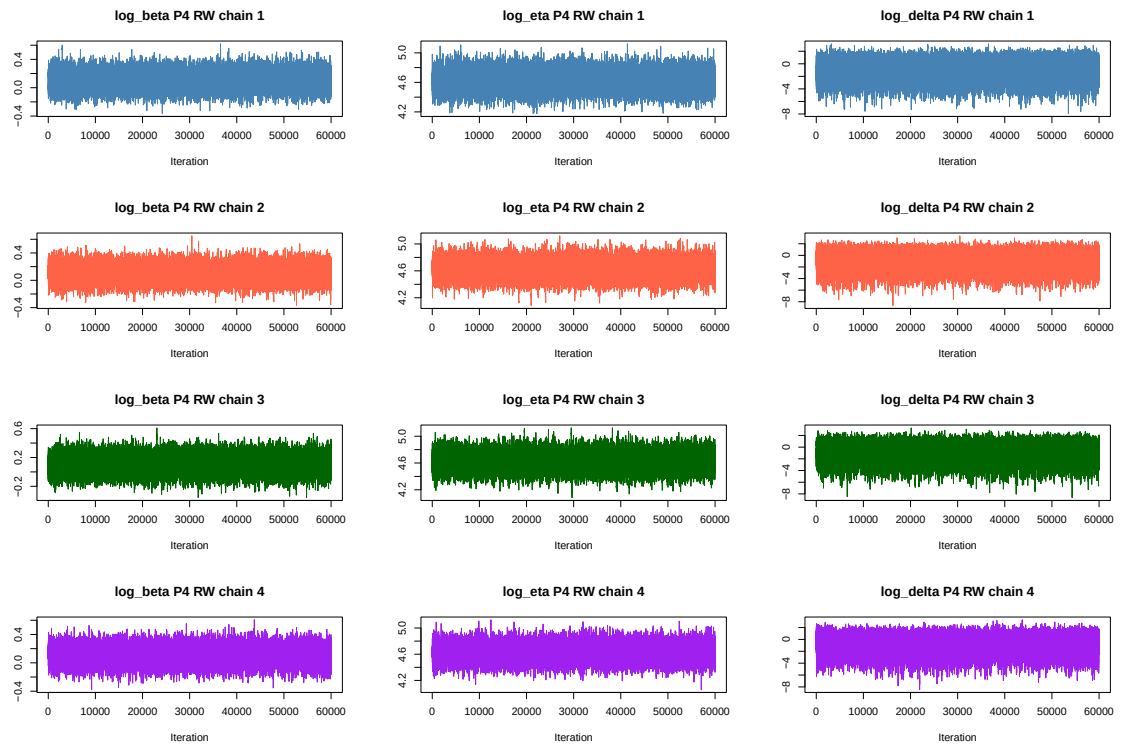


Figure C.24: Random-walk trace plots for Population P4

C.6.2 Additional Independence-Sampler Trace Plots for the Semi-Simulated Alloy Study

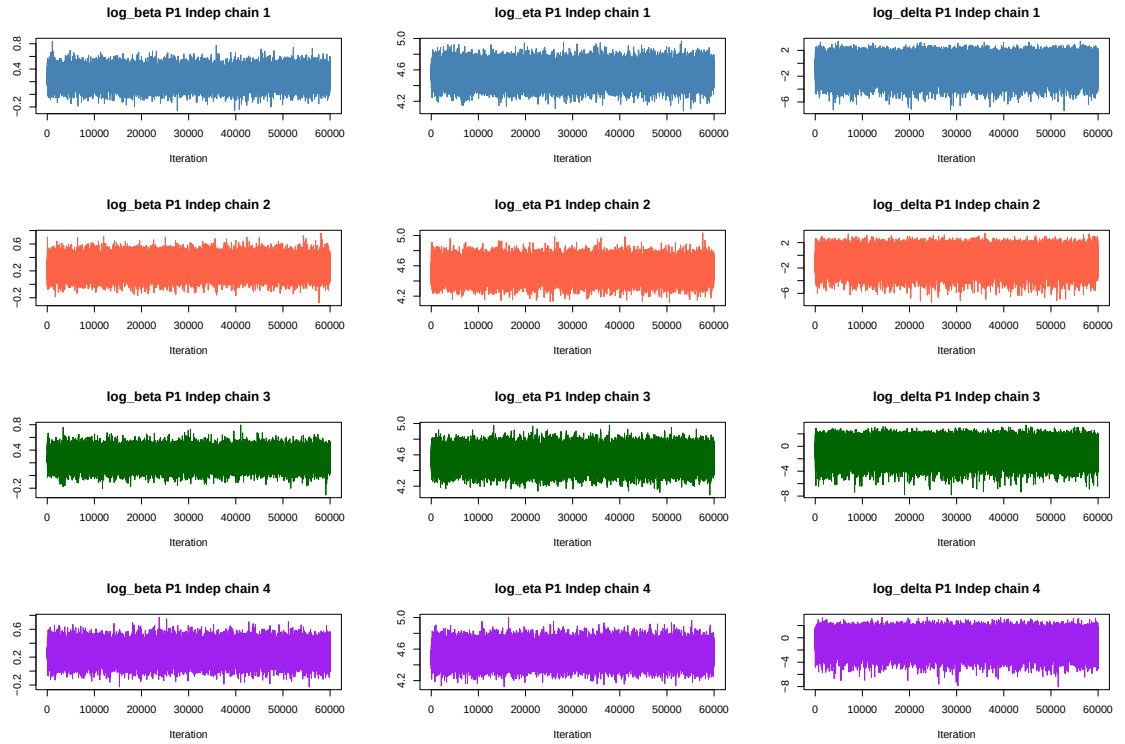


Figure C.25: Independence-sampler trace plots for Population P1

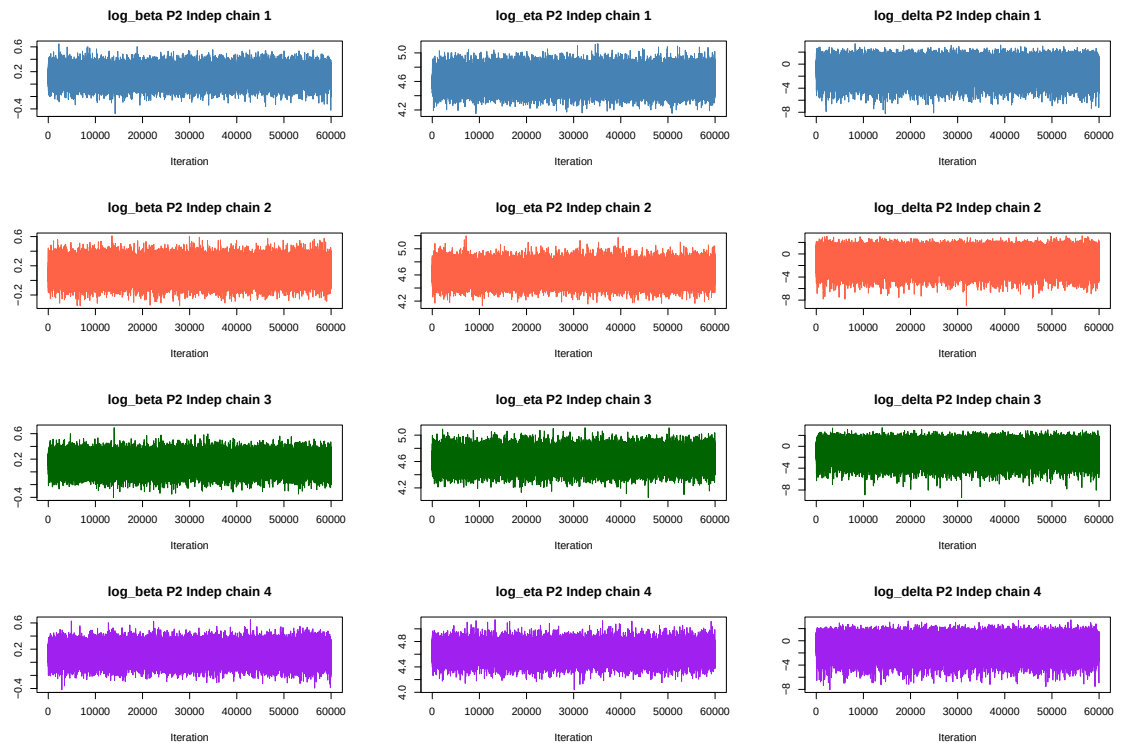


Figure C.26: Independence-sampler trace plots for Population P2

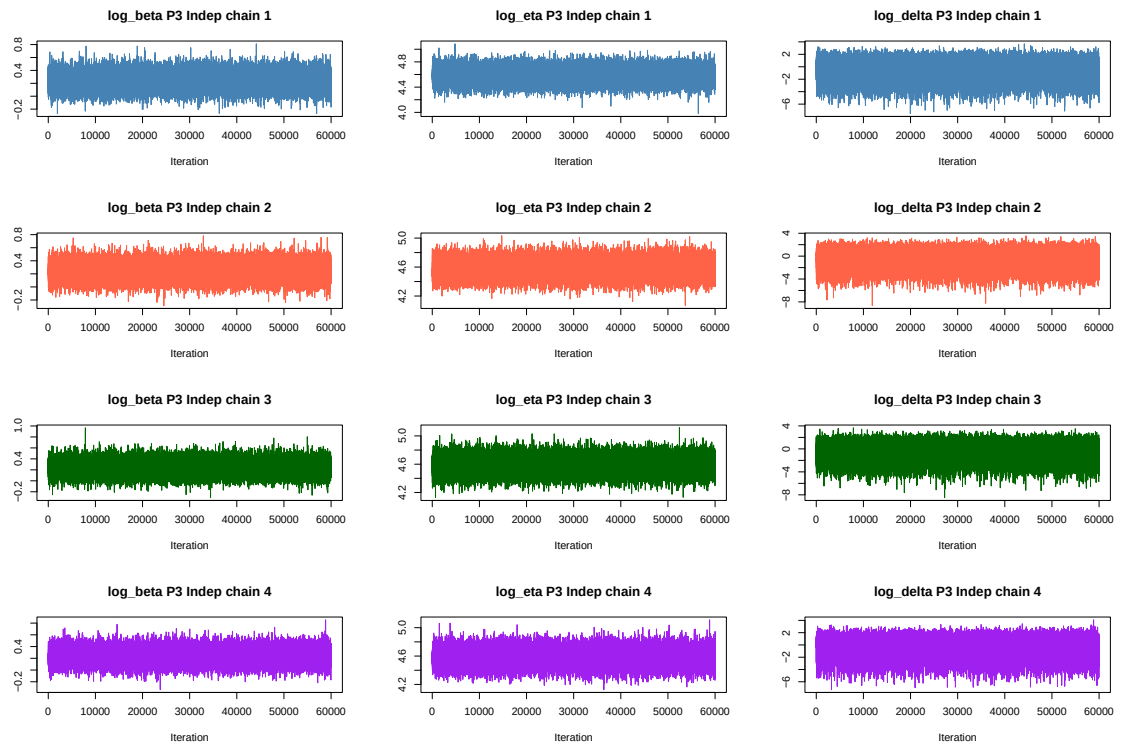


Figure C.27: Independence-sampler trace plots for Population P3

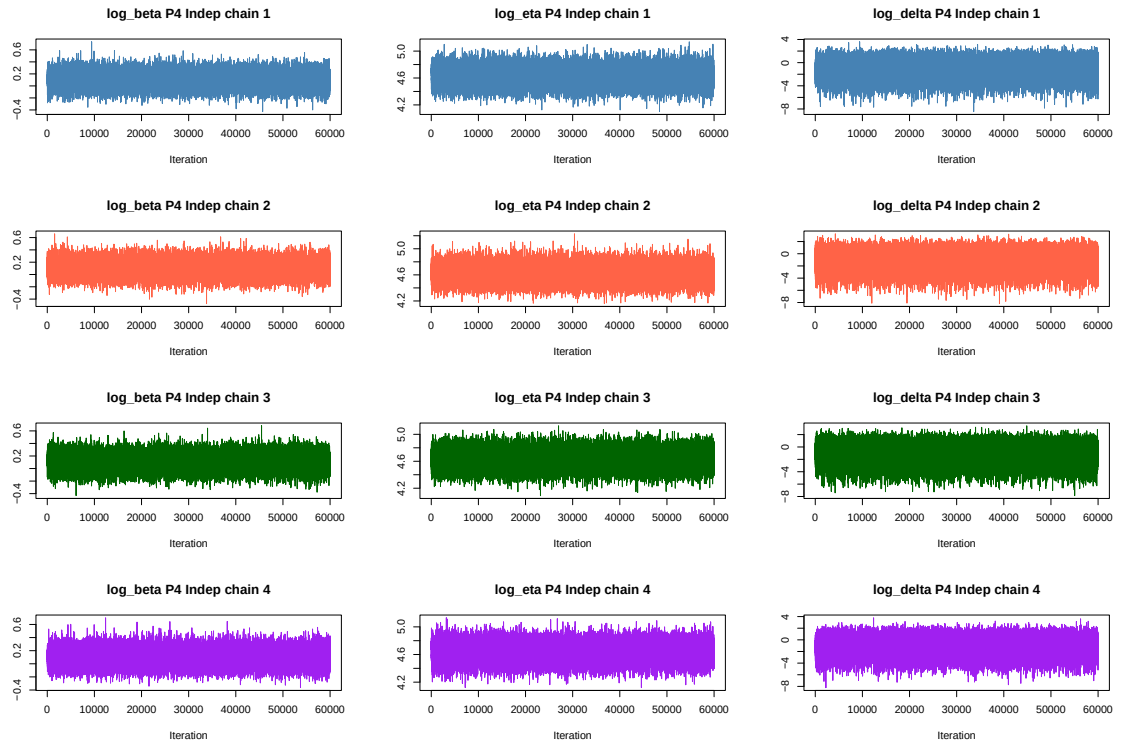


Figure C.28: Independence-sampler trace plots for Population P4

C.7 Additional Semi-Simulated Alloy Study Autocorrelation Functions

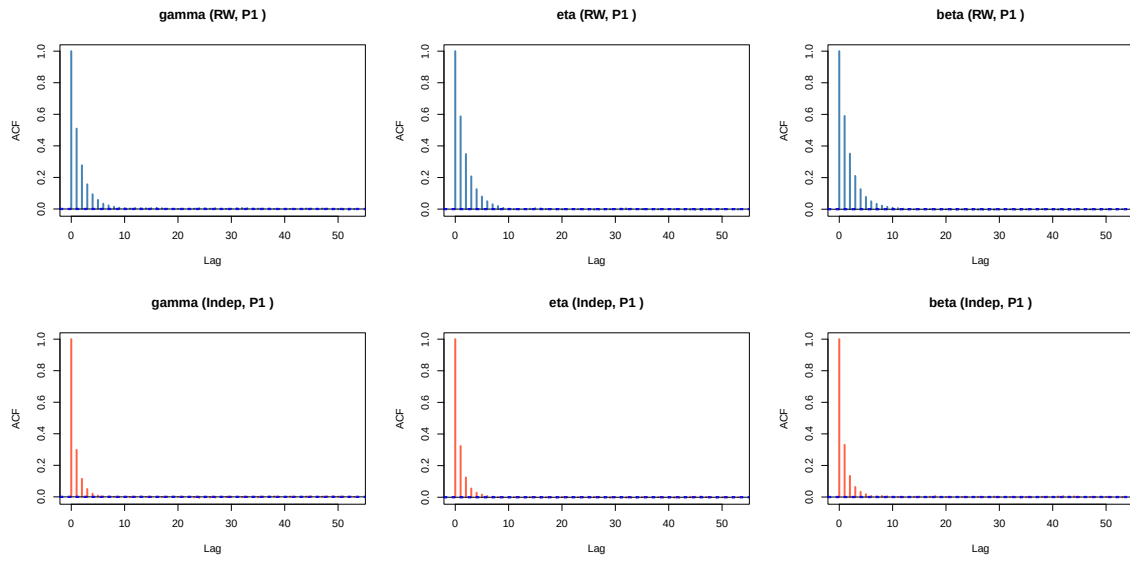


Figure C.29: Autocorrelation functions for γ , η , and β for Population P1

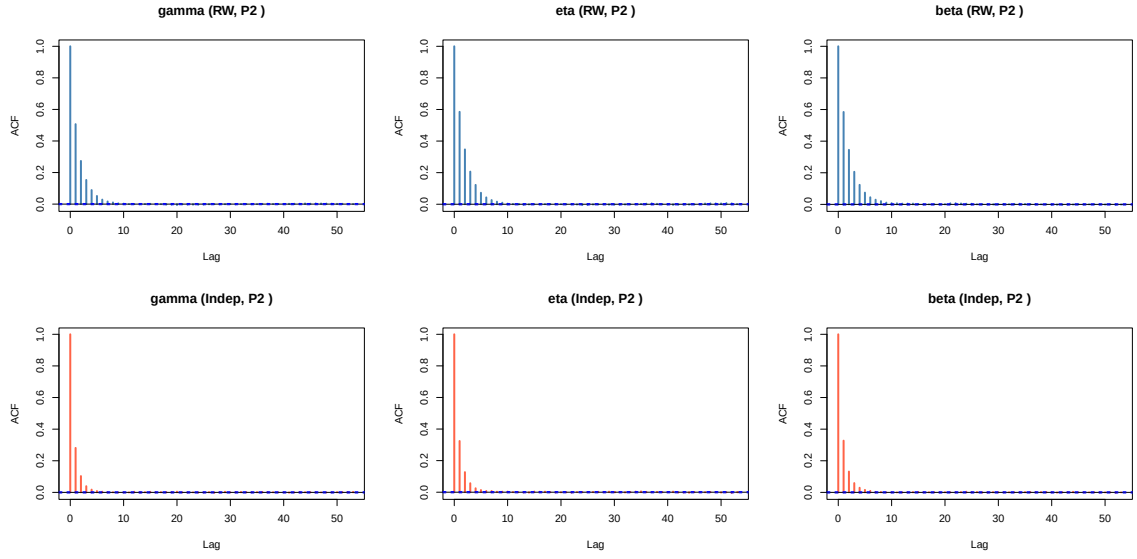


Figure C.30: Autocorrelation functions for γ , η , and β for Population P2

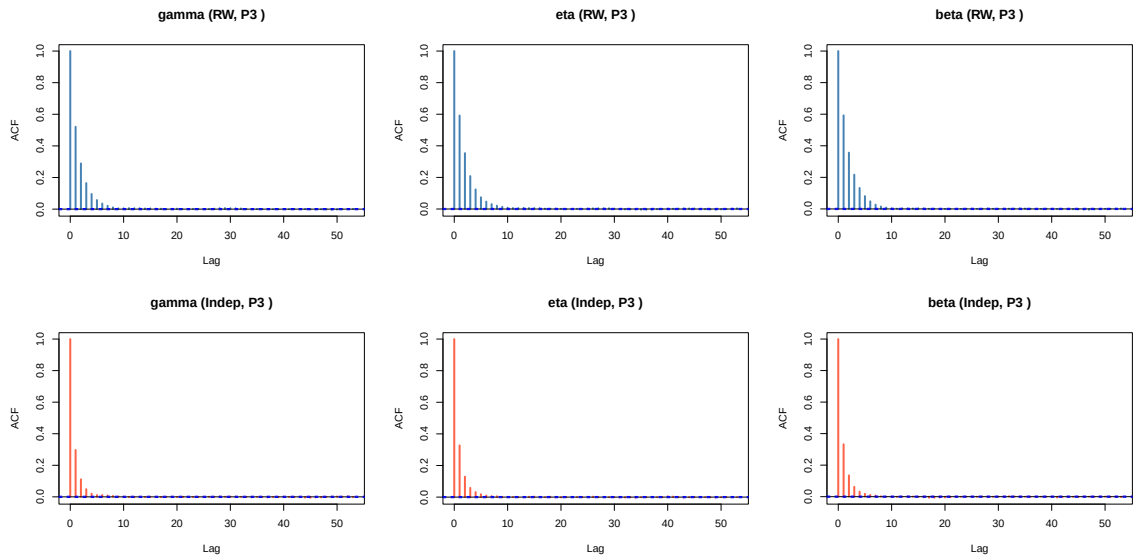


Figure C.31: Autocorrelation functions for γ , η , and β for Population P3

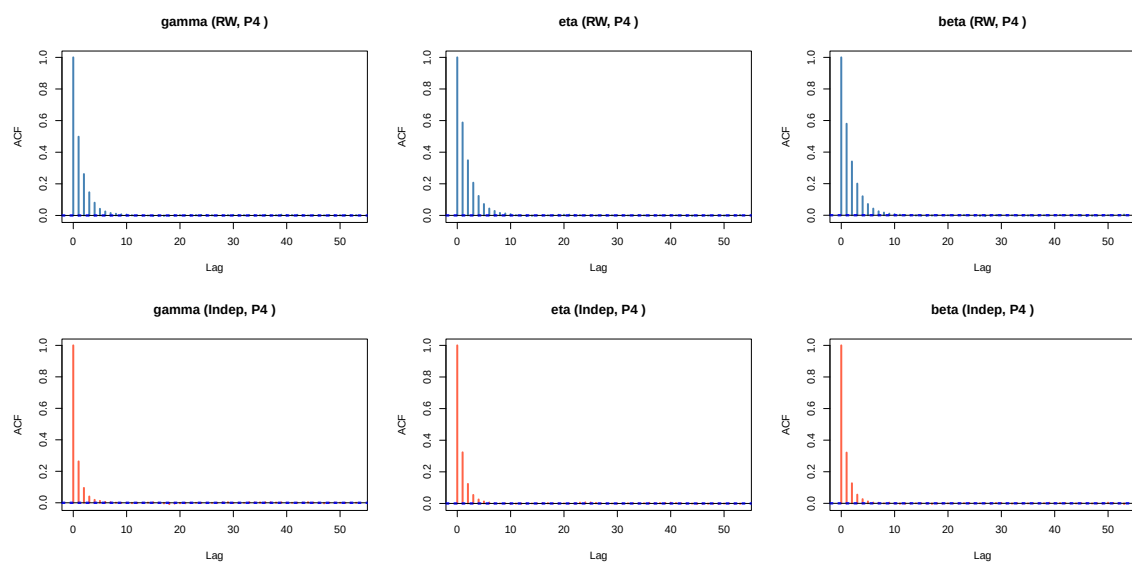


Figure C.32: Autocorrelation functions for γ , η , and β for Population P4

C.8 Additional Semi-Simulated Alloy Study Posterior Predictive Check Plots

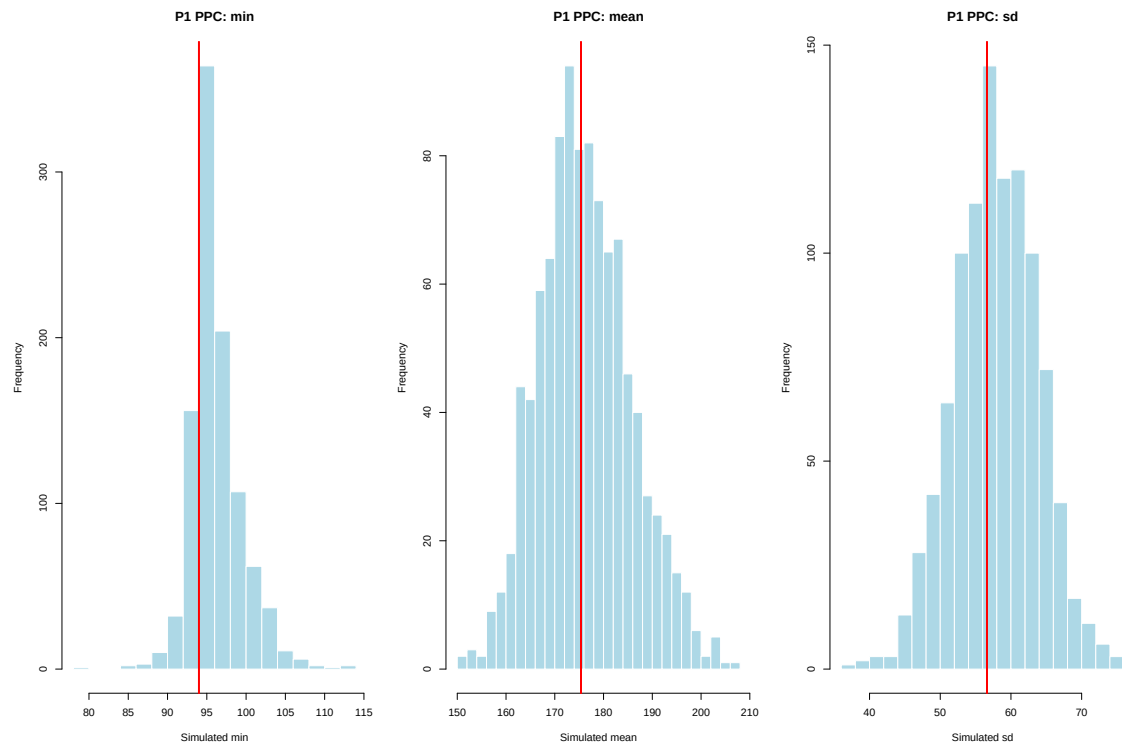


Figure C.33: Posterior predictive checks for Population P1

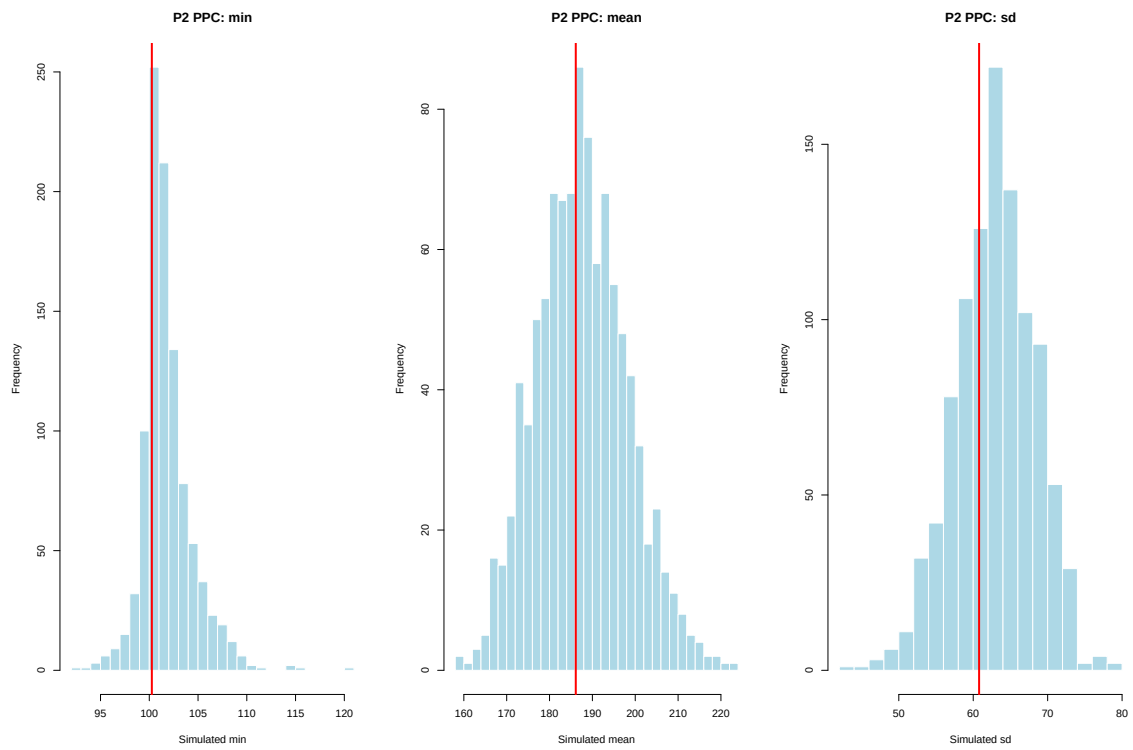


Figure C.34: Posterior predictive checks for Population P2

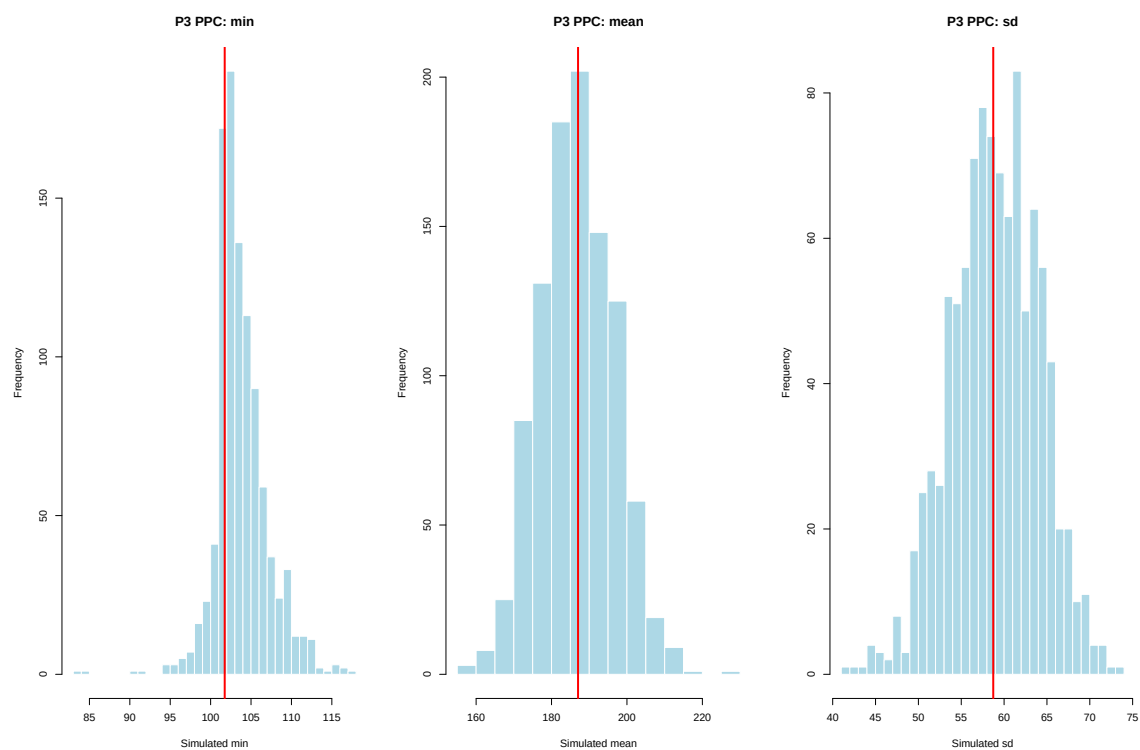


Figure C.35: Posterior predictive checks for Population P3

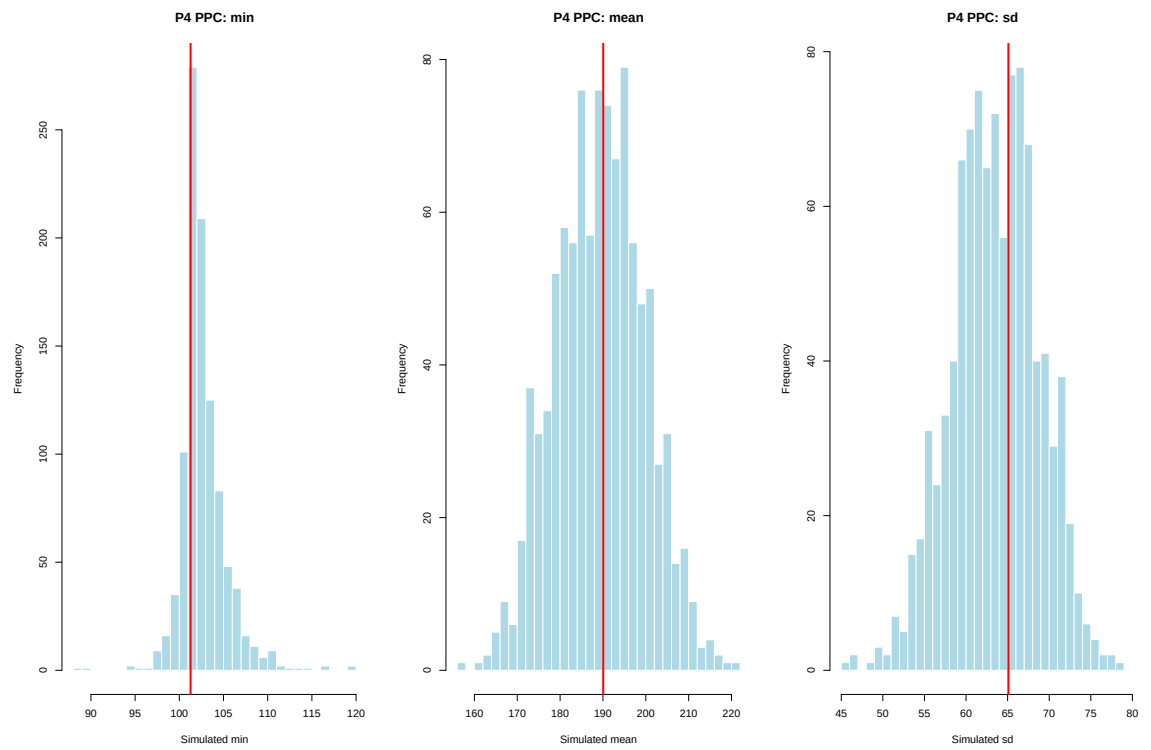


Figure C.36: Posterior predictive checks for Population P4.

REFERENCES

- [1] S. S. Gupta, "On some multiple decision (selection and ranking) rules," *Technometrics*, vol. 7, pp. 225–245, 1965.
- [2] S. S. Gupta and S. Panchapakesan, *Multiple Decision Procedures*. John Wiley & Sons, Inc, 1979.
- [3] J. D. Gibbons, I. Olkin, and M. Sobel, *Selecting and Ordering Populations: A New Statistical Methodology*. Society for Industrial and Applied Mathematics, 1999.
- [4] R. E. Bechhofer, "A single-sample multiple decision procedure for ranking means of normal populations with known variances," *The Annals of Mathematical Statistics*, vol. 25, pp. 16–39, 1954.
- [5] Y. Rinott, "On two-stage selection procedures and related probability-inequalities," *Communications in Statistics—Theory and Methods*, vol. 7, no. 8, pp. 799–811, 1978.
- [6] E. Paulson, "A sequential procedure for selecting the population with the largest mean from k normal populations," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 174–180, 1964.
- [7] Y. Kondo and J. V. Zidek, "Bayesian nonparametric subset selection procedures with weibull components," Tech. Rep. Technical Report 273, Department of Statistics, The University of British Columbia, may 2013. Technical report version.
- [8] S. S. Gupta and S. He, "Selecting the best weibull population based on type-i censored data: A bayesian approach," Tech. Rep. TR97-03C, Department of Statistics, Purdue University, 1997.
- [9] P. I. Frazier, "Tutorial: Optimization via simulation with bayesian statistics and dynamic programming," in *Proceedings of the 2012 Winter Simulation Conference*, 2012. Tutorial paper. Available as a conference tutorial PDF.
- [10] H. Singh and A. N. Gill, "Ranking and subset selection procedures for populations with censored data: A review," in *Analysis of Censored Data* (H. L. Koul and J. V. Deshpande, eds.), vol. 27 of *Institute of Mathematical Statistics Lecture Notes—Monograph Series*, Hayward, CA: Institute of Mathematical Statistics, 1995.
- [11] J. F. Lawless, *Statistical Models and Methods for Lifetime Data*. John Wiley & Sons, Inc., 2002.
- [12] W. Nelson, *Applied Life Data Analysis*. John Wiley & Sons, Inc., 2004.

- [13] W. Q. Meeker, L. A. Escobar, and F. G. Pascual, *Statistical Methods for Reliability Data*. John Wiley & Sons, Inc., 2 ed., 2022.
- [14] H. Rinne, *The Weibull Distribution: A Handbook*. Chapman & Hall/CRC, 2009.
- [15] G. C. McDonald and J. Hodaj, “Selection rules for exponential population threshold parameters,” *Applied Mathematics*, vol. 16, pp. 1–14, 2025.
- [16] G. C. McDonald and J. Hodaj, “Operating characteristics of subset selection rules for exponential population threshold parameters,” *Applied Mathematics*, vol. 16, pp. 441–460, 2025.
- [17] K. Lam, “A new procedure for selecting good populations,” *Biometrika*, vol. 73, pp. 201–206, 1986.
- [18] C. K. Ng, “Procedures for selecting good exponential populations,” *Communications in Statistics—Simulation and Computation*, vol. 42, pp. 1681–1692, 2013.
- [19] G. Casella and R. L. Berger, *Statistical Inference*. Duxbury, 2002.
- [20] J. Albert and J. Hu, *Probability and Bayesian Modeling*. CRC Press, Taylor & Francis Group, 2020.
- [21] R. E. Bechhofer, T. J. Santner, and D. M. Goldsman, *Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons*. John Wiley & Sons, Inc., 1995.
- [22] G. C. McDonald and J. Hodaj, “A note comparing two subset selection procedures for the threshold parameters of two exponential populations,” *Applied Mathematics*, vol. 16, pp. 877–883, 2025.
- [23] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions, Volume 1*. Wiley, 1994.
- [24] T. A. Hsu, “On some optimal selection procedures for weibull populations,” *Communications in Statistics—Theory and Methods*, vol. 11, pp. 2657–2668, 1982.
- [25] J. V. Kingston and J. K. Patel, “Selecting the best one of several weibull populations,” *Communications in statistics-Theory and Methods*, vol. 9(4), pp. 383–398, 1980.
- [26] R. Christensen, W. Johnson, A. Branscum, and T. E. Hanson, *Bayesian Ideas and Data Analysis*. CRC Press, Taylor & Francis Group, 2010.
- [27] W. M. Bolstad, *Understanding Computational Bayesian Statistics*. John Wiley & Sons, Inc., 2009.

- [28] M. S. Hamada, A. G. Wilson, S. C. Reese, and H. F. Martz, *Bayesian Reliability*. Springer, 2008.
- [29] H. F. Martz and R. A. Waller, *Bayesian Reliability Analysis*. John Wiley & Sons, Inc., 1982.
- [30] R. E. Kass and L. Wasserman, “The selection of prior distributions by formal rules,” *Journal of the American Statistical Association*, vol. 91(435), pp. 1343–1370, 1996.
- [31] D. Sun, “A note on noninformative priors for weibull distributions,” *Journal of Statistical Planning and Inference*, vol. 61, pp. 319–338, 1997.
- [32] D. I. De Souza, *Bayesian Weibull Estimation Using a Beta Prior for the Shape Parameter and an Inverse-Weibull Prior for the Scale Parameter*. Ph.d. dissertation, Wayne State University, Detroit, MI, 1987. Department of Mathematics.
- [33] M. Roos, T. G. Martins, L. Held, and H. Rue, “Sensitivity analysis for bayesian hierarchical models,” *Bayesian Analysis*, vol. 10, no. 2, pp. 321–349, 2015.
- [34] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [35] W. K. Hastings, “Monte carlo sampling methods using markov chains and their applications,” *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.
- [36] S. Chib and E. Greenberg, “Understanding the metropolis-hastings algorithm,” *The American Statistician*, vol. 49, no. 4, pp. 327–335, 1995.
- [37] A. Gelman and D. B. Rubin, “Inference from iterative simulation using multiple sequences,” *Statistical Science*, vol. 7, no. 4, pp. 457–472, 1992.
- [38] S. P. Brooks and A. Gelman, “General methods for monitoring convergence of iterative simulations,” *Journal of Computational and Graphical Statistics*, vol. 7, no. 4, pp. 434–455, 1998.
- [39] R. Jiang and D. N. P. Murthy, “A study of weibull shape parameter: Properties and significance,” *Reliability Engineering & System Safety*, vol. 96, no. 12, pp. 1619–1626, 2011.
- [40] G. O. Roberts, A. Gelman, and W. R. Gilks, “Weak convergence and optimal scaling of random walk metropolis algorithms,” *The Annals of Applied Probability*, vol. 7, no. 1, pp. 110–120, 1997.
- [41] G. O. Roberts and J. S. Rosenthal, “Optimal scaling for various metropolis-hastings algorithms,” *Statistical Science*, vol. 16, no. 4, pp. 351–367, 2001.

- [42] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis*. Chapman and Hall/CRC, 3 ed., 2013.
- [43] A. Gelman, X. L. Meng, and H. Stern, “Posterior predictive assessment of model fitness via realized discrepancies,” *Statistica Sinica*, vol. 6, no. 4, pp. 733–807, 1996.