Reliability Estimation in Series Systems: Maximum Likelihood Techniques for Right-Censored and Masked Failure Data

Alex Towell

Abstract

This paper investigates maximum likelihood techniques to estimate component reliability from masked failure data in series systems. A likelihood model accounts for right-censoring and candidate sets indicative of masked failure causes. Extensive simulation studies assess the accuracy and precision of maximum likelihood estimates under varying sample size, masking probability, and right-censoring time. The studies specifically examine the accuracy (bias) and precision of estimates, along with the coverage probability and width of BCa confidence intervals. Despite significant masking and censoring, the maximum likelihood estimator demonstrates good overall performance. The bootstrap yields reasonably well-calibrated confidence intervals even for small sample sizes. Together, the modeling framework and simulation studies provide rigorous validation of statistical learning from masked reliability data.

Contents

1 Introduction **2**

2 Series System Model **2**
2.1 Component Cause of Failure 5
2.2 System and Component Reliabilities 7

3 Likelihood Model for Masked Data **7**
3.1 Masked Component Cause of Failure 9
3.2 Right-Censored Data 11
3.3 Identifiability and Convergence Issues 12

4 Maximum Likelihood Estimation **12**

5 Bias-Corrected and Accelerated Bootstrap Confidence Intervals **13**

6 Series System with Weibull Components **14**
6.1 Likelihood Model 16
6.2 Weibull Series System: Homogeneous Shape Parameters 17

7 Simulation Study: Series System with Weibull Components **18**
7.1 Performance Metrics and Convergence Rate 19
7.2 Data Generating Process 19
7.3 Overview of Simulations 21
7.4 Scenario: Assessing the Impact of Right-Censoring 22
7.5 Scenario: Assessing the Impact of Masking Probability for Component Cause of Failure . . . 24
7.6 Scenario: Assessing the Impact of Sample Size 27

8 Future Work **29**

9 Conclusion **29**

A Series System with Weibull Component Lifetimes **30**
A.1 Log-likelihood Function 30
A.2 Score Function 30
A.3 Quantile Function 31
A.4 Maximum Likelihood Estimation 32

B Simulation **32**
B.1 Scenario Simulation 32
B.2 Bernoulli Candidate Set Model 34
B.3 Plot Generation for Sampling Distribution of MLE 34
B.4 Plot Generation for Coverage Probabilities 35

Acknowledgements **37**

1 Introduction

Quantifying the reliability of individual components in a series system [2] is challenging when only system-level failure data is observable, especially when this data is masked by right-censoring and ambiguity about the cause of failure. This paper develops and validates maximum likelihood techniques to estimate component reliability from right-censored lifetimes and candidate sets indicative of masked failure causes. Specific contributions include:

- Deriving a likelihood model that incorporates right-censoring and candidate sets to enable masked data to be used for parameter estimation.
- Conducting simulation studies for a well-designed series system with component lifetimes following a Weibull distribution. We assess the accuracy and precision of maximum likelihood estimates (MLE) under varying conditions related to sample size, masking probability, and right-censoring. We found that the MLE performs well in the presence of significant masking and censoring even for relatively small samples, but that the shape parameters is more difficult to estimate than the scale parameters.
- Evaluating the coverage probability (accuracy) and precision of the BCa confidence intervals constructed for the MLE. We found that the BCa confidence intervals are reasonably well-calibrated even for small sample sizes in the presence of significant masking and censoring, but that the confidence intervals for the shape parameters were slightly less well-calibrated than the confidence intervals for the scale parameters.

The simulation studies focus on three key aspects:

- The impact of right-censoring on component parameter estimates.
- How masking probability for the cause of failure affects the estimates.
- The role of sample size in mitigating challenges related to censoring and masking.

Together, the likelihood framework and simulation methodology enable rigorous validation of learning component reliability from limited masked system data. This expands the capability to quantify latent properties and perform robust statistical inference given significant data challenges.

2 Series System Model

Consider a system composed of m components arranged in a series configuration. Each component and system has two possible states, functioning or failed. We have n systems whose lifetimes are independent and identically distributed (i.i.d.). The lifetime of the i^{th} system is denoted by the random variable T_i and the lifetime of its j^{th} component is denoted by the random variable T_{ij}. We assume the component lifetimes in a single system are statistically independent and non-identically distributed. Here, lifetime (or lifespan) is defined as the elapsed time from when the new, functioning component (or system) is put into operation

until it fails for the first time. A series system fails when any component fails, thus the lifetime of the i^{th} system is given by the component with the shortest lifetime,

$$T_i = \min\{T_{i1}, T_{i2}, \dots, T_{im}\}.$$

There are three particularly important distribution functions in reliability analysis: the reliability function, the probability density function, and the hazard function. The reliability function, $R_{T_i}(t)$, is the probability that the i^{th} system has a lifetime greater than given duration t,

$$R_{T_i}(t) = \Pr\{T_i > t\} \tag{1}$$

The probability density function (pdf) of T_i is denoted by $f_{T_i}(t)$ and may be defined as

$$f_{T_i}(t) = -\frac{d}{dt} R_{T_i}(t).$$

Next, we introduce the hazard function. The probability that a failure occurs between t and Δt given that no failure occurs before time t is given by

$$\Pr\{T_i \leq t + \Delta t | T_i > t\} = \frac{\Pr\{t < T_i < t + \Delta t\}}{\Pr\{T_i > t\}}.$$

The failure rate is given by the dividing this equation by the length of the time interval, Δt:

$$\frac{\Pr\{t < T_i < t + \Delta t\}}{\Delta t} \frac{1}{\Pr\{T_i > t\}} = -\frac{R_{T_i}(t + \Delta t) - R_{T_i}(t)}{\Delta t} \frac{1}{R_{T_i}(t)}.$$

The hazard function $h_{T_i}(t)$ for T_i is the instantaneous failure rate at time t, which is given by

$$\begin{aligned} h_{T_i}(t) &= -\lim_{\Delta t \to 0} \frac{R_{T_i}(t + \Delta t) - R_{T_i}(t)}{\Delta t} \frac{1}{R_{T_i}(t)} \\ &= \left(-\frac{d}{dt} R_{T_i}(t)\right) \frac{1}{R_{T_i}(t)} = \frac{f_{T_i}(t)}{R_{T_i}(t)} \end{aligned} \tag{2}$$

The lifetime of the j^{th} component is assumed to follow a parametric distribution indexed by a parameter vector θ_j. The parameter vector of the overall system is defined as

$$\theta = (\theta_1, \dots, \theta_m),$$

where θ_j is the parameter vector of the j^{th} component.

When a random variable X is parameterized by a particular θ, we denote the reliability function by $R_X(t; \theta)$, and the same for the other distribution functions. As a special case, for the components in a series system, we subscript by their labels, e.g, the pdf of the j^{th} component is denoted by $f_j(t; \theta_j)$. Two continuous random variables X and Y have a joint pdf $f_{X,Y}(x, y)$. Given the joint pdf $f(x, y)$, the marginal pdf of X is given by

$$f_X(x) = \int_{\mathcal{Y}} f_{X,Y}(x, y) dy,$$

where $\mathcal{Y}$ is the support of Y. (If Y is discrete, replace the integration with a summation over $\mathcal{Y}$.)

The conditional pdf of Y given $X = x$, $f_{Y|X}(y|x)$, is defined as

$$f_{X|Y}(y|x) = \frac{f_{X,Y}(x, y)}{f_X(x)}.$$

We may generalize all of the above to more than two random variables, e.g., the joint pdf of $X_1, \dots, X_m$ is denoted by $f(x_1, \dots, x_m)$.

Next, we dive deeper into these concepts and provide mathematical derivations for the reliability function, pdf, and hazard function of the series system. We begin with the reliability function of the series system, as given by the following theorem.

Theorem 2.1. *The series system has a reliability function given by*

$$R_{T_i}(t;\theta) = \prod_{j=1}^{m} R_j(t;\theta_j). \tag{3}$$

Proof. The reliability function is defined as

$$R_{T_i}(t;\theta) = \Pr\{T_i > t\}$$

which may be rewritten as

$$R_{T_i}(t;\theta) = \Pr\{\min\{T_{i1}, \ldots, T_{im}\} > t\}.$$

For the minimum to be larger than t, every component must be larger than t,

$$R_{T_i}(t;\theta) = \Pr\{T_{i1} > t, \ldots, T_{im} > t\}.$$

Since the component lifetimes are independent, by the product rule the above may be rewritten as

$$R_{T_i}(t;\theta) = \Pr\{T_{i1} > t\} \times \cdots \times \Pr\{T_{im} > t\}.$$

By definition, $R_j(t;\theta) = \Pr\{T_{ij} > t\}$. Performing this substitution obtains the result

$$R_{T_i}(t;\theta) = \prod_{j=1}^{m} R_j(t;\theta_j).$$

□

Theorem 2.1 shows that the system's overall reliability is the product of the reliabilities of its individual components. This is an important relationship in all series systems and will be used in the subsequent derivations. Next, we turn our attention to the pdf of the system lifetime, described in the following theorem.

Theorem 2.2. *The series system has a pdf given by*

$$f_{T_i}(t;\theta) = \sum_{j=1}^{m} f_j(t;\theta_j) \prod_{\substack{k=1 \\ k\neq j}}^{m} R_k(t;\theta_j). \tag{4}$$

Proof. By definition, the pdf may be written as

$$f_{T_i}(t;\theta) = -\frac{d}{dt} \prod_{j=1}^{m} R_j(t;\theta_j).$$

By the product rule, this may be rewritten as

$$\begin{aligned} f_{T_i}(t;\theta) &= -\frac{d}{dt} R_1(t;\theta_1) \prod_{j=2}^{m} R_j(t;\theta_j) - R_1(t;\theta_1) \frac{d}{dt} \prod_{j=2}^{m} R_j(t;\theta_j) \\ &= f_1(t;\theta) \prod_{j=2}^{m} R_j(t;\theta_j) - R_1(t;\theta_1) \frac{d}{dt} \prod_{j=2}^{m} R_j(t;\theta_j). \end{aligned}$$

Recursively applying the product rule $m-1$ times results in

$$f_{T_i}(t;\theta) = \sum_{j=1}^{m-1} f_j(t;\theta_j) \prod_{\substack{k=1 \\ k\neq j}}^{m} R_k(t;\theta_k) - \prod_{j=1}^{m-1} R_j(t;\theta_j) \frac{d}{dt} R_m(t;\theta_m),$$

which simplifies to

$$f_{T_i}(t;\theta) = \sum_{j=1}^{m} f_j(t;\theta_j) \prod_{\substack{k=1 \\ k\neq j}}^{m} R_k(t;\theta_k).$$

□

Theorem 2.2 shows the pdf of the system lifetime is a function of the pdfs and reliabilities of its components. We continue with the hazard function of the system lifetime, defined in the next theorem.

Theorem 2.3. *The series system has a hazard function given by*

$$h_{T_i}(t;\theta) = \sum_{j=1}^{m} h_j(t;\theta_j)$$

Proof. By Equation (2), the i^{th} series system lifetime has a hazard function defined as

$$h_{T_i}(t;\theta) = \frac{f_{T_i}(t;\theta)}{R_{T_i}(t;\theta)}.$$

Plugging in expressions for these functions results in

$$h_{T_i}(t;\theta) = \frac{\sum_{j=1}^{m} f_j(t;\theta_j) \prod_{\substack{k=1 \\ k\neq j}}^{m} R_k(t;\theta_k)}{\prod_{j=1}^{m} R_j(t;\theta_j)},$$

which can be simplified to

$$h_{T_i}(t;\theta) = \sum_{j=1}^{m} \frac{f_j(t;\theta_j)}{R_j(t;\theta_j)} = \sum_{j=1}^{m} h_j(t;\theta_j).$$

□

Theorem 2.3 reveals that the system's hazard function is the sum of the hazard functions of its components. By definition, the hazard function is the ratio of the pdf to the reliability function,

$$h_{T_i}(t;\theta) = \frac{f_{T_i}(t;\theta)}{R_{T_i}(t;\theta)},$$

and we can rearrange this to get

$$\begin{aligned} f_{T_i}(t;\theta) &= h_{T_i}(t;\theta) R_{T_i}(t;\theta) \\ &= \left\{\sum_{j=1}^{m} h_j(t;\theta_j)\right\}\left\{\prod_{j=1}^{m} R_j(t;\theta_j)\right\}, \end{aligned} \tag{5}$$

which we sometimes find to be a more convenient form than Equation (4).

In this section, we derived the mathematical forms for the system's reliability, probability density, and hazard functions. Next, we build upon these concepts to derive distributions related to the component cause of failure.

2.1 Component Cause of Failure

Whenever a series system fails, precisely one of the components is the cause. We denote the component cause of failure of a series system by the random variable K_i, whose support is given by $\{1, \ldots, m\}$. For example, $K_i = j$ indicates that the component indexed by j failed first, i.e.,

$$T_{ij} < T_{ij'}$$

for every j' in the support of K_i except for j. Since we have series systems, K_i is unique.

The system lifetime and the component cause of failure has a joint distribution given by the following theorem.

Theorem 2.4. *The joint pdf of the component cause of failure K_i and series system lifetime T_i is given by*

$$f_{K_i,T_i}(j,t;\theta) = h_j(t;\theta_j) \prod_{l=1}^{m} R_l(t;\theta), \tag{6}$$

where $h_l(t;\theta_j)$ and $R_l(t;\theta_l)$ are respectively the hazard and reliability functions of the l^{th} component.

Proof. Consider a series system with 3 components. By the assumption that component lifetimes are mutually independent, the joint pdf of T_{i1}, T_{i2}, T_{i3} is given by

$$f(t_1, t_2, t_3; \theta) = \prod_{j=1}^{3} f_j(t_j; \theta_j),$$

where $f_j(t_j; \theta_j)$ is the pdf of the j^{th} component. The first component is the cause of failure at time t if $K_i = 1$ and $T_i = t$, which may be rephrased as the likelihood that $T_{i1} = t$, $T_{i2} > t$, and $T_{i3} > t$. Thus,

$$\begin{aligned} f_{K_i,T_i}(j, t; \theta) &= \int_t^\infty \int_t^\infty f_1(t; \theta_1) f_2(t_2; \theta_2) f_3(t_3; \theta_3) dt_3 dt_2 \\ &= \int_t^\infty f_1(t; \theta_1) f_2(t_2; \theta_2) R_3(t; \theta_3) dt_2 \\ &= f_1(t; \theta_1) R_2(t; \theta_2) R_3(t_1; \theta_3). \end{aligned}$$

By definition, $f_1(t; \theta_1) = h_1(t; \theta_1) R_1(t; \theta_1)$, and when we make this substitution into the above expression for $f_{K_i,T_i}(j, t; \theta)$, we obtain the result

$$f_{K_i,T_i}(j, t; \theta) = h_1(t; \theta_1) \prod_{l=1}^{m} R_l(t; \theta_l).$$

Generalizing this result completes the proof. □

Theorem 2.4 shows that the joint pdf of the component cause of failure and system lifetime is a function of the hazard functions and reliability functions of the components. This result will be used in the Section 3 to derive the likelihood function for the masked data.

The probability that the j^{th} component is the cause of failure is given by the following theorem.

Theorem 2.5. *The probability that the j^{th} component is the cause of failure is given by*

$$\Pr\{K_i = j\} = E_\theta \left[\frac{h_j(T_i; \theta_j)}{\sum_{l=1}^{m} h_l(T_i; \theta_l)} \right] \tag{7}$$

where K_i is the random variable denoting the component cause of failure of the i^{th} system and T_i is the random variable denoting the lifetime of the i^{th} system.

Proof. The probability the j^{th} component is the cause of failure is given by marginalizing the joint pdf of K_i and T_i over T_i,

$$\Pr\{K_i = j\} = \int_0^\infty f_{K_i,T_i}(j, t; \theta) dt.$$

By Theorem 2.4, this is equivalent to

$$\begin{aligned} \Pr\{K_i = j\} &= \int_0^\infty h_j(t; \theta_j) R_{T_i}(t; \theta) dt \\ &= \int_0^\infty \left(\frac{h_j(t; \theta_j)}{h_{T_i}(t; \theta)} \right) f_{T_i}(t; \theta) dt \\ &= E_\theta \left[\frac{h_j(T_i; \theta_j)}{\sum_{l=1}^{m} h_l(T_i; \theta_l)} \right]. \end{aligned}$$

□

If we know the system failure time, then we can simplify the above expression for the probability that the j^{th} component is the cause of failure. This is given by the following theorem.

Theorem 2.6. *The probability that the j^{th} component is the cause of system failure given that we know the system failure occured at time t_i is given by*

$$\Pr\{K_i = j | T_i = t_i\} = \frac{h_j(t_i;\theta_j)}{\sum_{l=1}^{m} h_l(t_i;\theta_l)}.$$

Proof. By the definition of conditional probability,

$$\begin{aligned}\Pr\{K_i = j | T_i = t_i\} &= \frac{f_{K_i,T_i}(j,t_i;\theta)}{f_{T_i}(t_i;\theta)} \\ &= \frac{h_j(t_i;\theta_j)R_{T_i}(t_i;\theta)}{f_{T_i}(t_i;\theta)}.\end{aligned}$$

Since $f_{T_i}(t_i;\theta) = h_{T_i}(t_i;\theta)R_{T_i}(t_i;\theta)$, we make this substitution and simplify to obtain

$$\Pr\{K_i = j | T_i = t_i\} = \frac{h_j(t_i;\theta_j)}{\sum_{l=1}^{m} h_l(t_i;\theta_l)}.$$

□

Theorems 2.5 and 2.6 are closely related and have similar forms. Theorem 2.6 can be seen as a special case of Theorem 2.5 because we can obtain Theorem 2.6 by setting $t_i = T_i$ in Theorem 2.5.

2.2 System and Component Reliabilities

The reliability of a system is described by its reliability function, which denotes the probability that the system is functioning at a given time, e.g., $R_{T_i}(t';\theta)$ denotes the probability that the i^{th} system is functioning at time t'. If we want a summary measure of the system's reliability, a common measure is the mean time to failure (MTTF), which is the expectation of the system lifetime,

$$\text{MTTF} = E_\theta[T_i], \tag{8}$$

which if certain assumptions are satisfied[1] is equivalent to the integration of the reliability function over its support. While the MTTF provides a summary measure of reliability, it is not a complete description. Depending on the failure characteristics, MTTF can be misleading. For example, a system that has a high likelihood of failing early in its life may still have a large MTTF if it is fat-tailed.[2]

The reliability of the components in the series system determines the reliability of the system. We denote the MTTF of the j^{th} component by MTTF_j and, according to Theorem 2.5, the probability that the j^{th} component is the cause of failure is given by $\Pr\{K_i = j\}$. In a **well-designed** series system, there are no components that are much "weaker" than any of the others, e.g., they have similar MTTFs and probabilities of being the component cause of failure. In this paper, we perform a sensivitivy analysis of the MLE for a well-designed series systems.

3 Likelihood Model for Masked Data

We aim to estimate an unknown parameter, θ, using *masked data*. We consider two types of masking: censoring of system failures and masking component causes of failure.

[1] T_i is non-negative and continuous, $R_{T_i}(t;\theta)$ is a well-defined, continuous, and differential function for $t > 0$, and $\int_0^\infty R_{T_i}(t;\theta)dt$ converges.

[2] A "fat-tailed" distribution refers to a probability distribution with tails that decay more slowly than those of the exponential family, such as the case with the Weibull when its shape parameter is greater than 1. This means that extreme values are more likely to occur, and the distribution is more prone to "black swan" events or rare occurrences. In the context of reliability, a fat-tailed distribution might imply a higher likelihood of unusually long lifetimes, which can skew measures like the MTTF. [12]

Censoring We generally encounter two types of censoring: the system failure is observed to occur within some time interval, or the system failure is not observed but we know that it was functioning at least until some point in time. The latter is known as *right-censoring*, which is the type of censoring we consider in this paper.

Component Cause of Failure Masking In the case of masking the component cause of failure, we may not know the precise component cause of failure, but we may have some indication. A common example is when a diagnostician is able to isolate the cause of failure to a subset of the components. We call this subset the *candidate set*.

Masked Data In this paper, each system is put into operation and observed until either it fails or its failure is right-censored after some duration τ, so we do not directly observe the system lifetime but rather we observe the right-censored lifetime, S_i, which is given by

$$S_i = \min\{\tau, T_i\}. \tag{9}$$

We also observe an event indicator, δ_i, which is given by

$$\delta_i = 1_{T_i<\tau}, \tag{10}$$

where $1_{\text{condition}}$ is an indicator function that denotes 1 if the condition is true and 0 otherwise. Here, $\delta_i = 1$ indicates the i^{th} system's failure was observed and $\delta_i = 0$ indicates it was right-censored.[3] If a system failure event is observed ($\delta_i = 1$), then we also observe a candidate set that contains the component cause of failure. We denote the candidate set for the i^{th} system by $\mathcal{C}_i$, which is a subset of $\{1, \dots, m\}$.

In summary, the observed data is assumed to be i.i.d. and is given by $D = \{D_1, \dots, D_n\}$, where each D_i contains the following elements:

- S_i is the right-censored system lifetime of the i^{th} system.
- δ_i is the event indicator for the i^{th} system.
- $\mathcal{C}_i$ is the set of candidate component causes of failure for the i^{th} system.

The masked data generation process is illustrated in Figure 1.

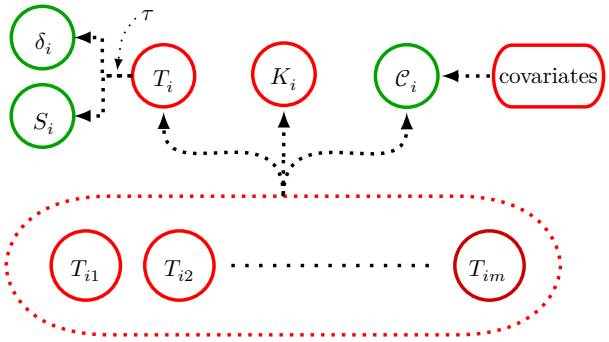

Figure 1: This figure showcases a dependency graph of the generative model for $D_i = (S_i, \delta_i, \mathcal{C}_i)$. The elements in green are observed in the sample, while the elements in red are unobserved (latent). We see that $\mathcal{C}_i$ is related to both the unobserved component lifetimes $T_{i1}, \dots, T_{im}$ and other unknown and unobserved covariates, like ambient temperature or the particular diagnostician who generated the candidate set. These two complications for $\mathcal{C}_i$ are why we seek a way to construct a reduced likelihood function in later sections that is not a function of the distribution of $\mathcal{C}_i$.

[3] In some likelihood models, there may be more than two possible values for δ_i, but in this paper, we only consider the case where δ_i is binary. Future work could consider the case where δ_i is categorical by including more types of censoring events and more types of component cause of failure masking.

An example of masked data D with a right-censoring time $\tau = 5$ can be seen in Table 1 for a series system with 3 components.

Table 1: Right-censored lifetime data with masked component cause of failure.

System	Right-censored lifetime (S_i)	Event indicator (δ_i)	Candidate set ($\mathcal{C}_i$)
1	1.1	1	$\{1, 2\}$
2	1.3	1	$\{2\}$
4	2.6	1	$\{2, 3\}$
5	3.7	1	$\{1, 2, 3\}$
6	5	0	$\emptyset$
3	5	0	$\emptyset$

In our model, we assume the data is governed by a pdf, which is determined by a specific parameter, represented as θ within the parameter space Ω. The joint pdf of the data D can be represented as follows:

$$f(D;\theta) = \prod_{i=1}^{n} f(D_i;\theta) = \prod_{i=1}^{n} f(s_i, \delta_i, c_i;\theta),$$

where s_i is the observed system lifetime, δ_i is the observed event indicator, and c_i is the observed candidate set of the i^{th} system.

This joint pdf tells us how likely we are to observe the particular data, D, given the parameter θ. When we keep the data constant and allow the parameter θ to vary, we obtain what is called the likelihood function L, defined as

$$L(\theta) = \prod_{i=1}^{n} L_i(\theta)$$

where

$$L_i(\theta) = f(D_i;\theta)$$

is the likelihood contribution of the i^{th} system.

For each type of data, right-censored data and masked component cause of failure data, we will derive the *likelihood contribution* L_i, which refers to the part of the likelihood function that this particular piece of data contributes to. We present the following theorem for the likelihood contribution model.

Theorem 3.1. *The likelihood contribution of the i-th system is given by*

$$L_i(\theta) \propto \prod_{j=1}^{m} R_j(s_i;\theta_j)\left(\sum_{j\in c_i} h_j(s_i;\theta_j)\right)^{\delta_i} \tag{11}$$

where $\delta_i = 0$ indicates the i^{th} system is right-censored at time s_i and $\delta_i = 1$ indicates the i^{th} system is observed to have failed at time s_i with a component cause of failure potentially masked by the candidate set c_i.

In the following subsections, we prove this result for each type of masked data, right-censored system lifetime data ($\delta_i = 0$) and masking of the component cause of failure ($\delta_i = 1$).

3.1 Masked Component Cause of Failure

Suppose a diagnostician is unable to identify the precise component cause of the failure, e.g., due to cost considerations he or she replaced multiple components at once, successfully repairing the system but failing to precisely identity the failed component. In this case, the cause of failure is said to be *masked*.

The unobserved component lifetimes may have many covariates, like ambient operating temperature, but the only covariate we observe in our masked data model are the system's lifetime and additional masked data in the form of a candidate set that is somehow correlated with the unobserved component lifetimes.

The key goal of our analysis is to estimate the parameter θ, which maximize the likelihood of the observed data, and to estimate the precision and accuracy of this estimate using the Bootstrap method. To achieve this, we first need to assess the joint distribution of the system's continuous lifetime, T_i, and the discrete candidate set, $\mathcal{C}_i$, which can be written as

$$f_{T_i,\mathcal{C}_i}(t_i, c_i; \theta) = f_{T_i}(t_i; \theta) \Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i\},$$

where $f_{T_i}(t_i; \theta)$ is the pdf of T_i and $\Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i\}$ is the conditional pmf of $\mathcal{C}_i$ given $T_i = t_i$.

We assume the pdf $f_{T_i}(t_i; \theta)$ is known, but we do not have knowledge of $\Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i\}$, i.e., the data generating process for candidate sets is unknown. However, it is critical that the masked data, $\mathcal{C}_i$, is correlated with the i^{th} system. This way, the conditional distribution of $\mathcal{C}_i$ given $T_i = t_i$ may provide information about θ, despite our statistical interest being primarily in the series system rather than the candidate sets.

To make this problem tractable, we assume a set of conditions that make it unnecessary to estimate the generative processes for candidate sets. The most important way in which $\mathcal{C}_i$ is correlated with the i^{th} system is given by assuming the following condition.

Condition 1. The candidate set $\mathcal{C}_i$ contains the index of the the failed component, i.e.,

$$\Pr_\theta\{K_i \in \mathcal{C}_i\} = 1$$

where K_i is the random variable for the failed component index of the i^{th} system.

Assuming Condition 1, $\mathcal{C}_i$ must contain the index of the failed component, but we can say little else about what other component indices may appear in $\mathcal{C}_i$.

In order to derive the joint distribution of $\mathcal{C}_i$ and T_i assuming Condition 1, we take the following approach. We notice that $\mathcal{C}_i$ and K_i are statistically dependent. We denote the conditional pmf of $\mathcal{C}_i$ given $T_i = t_i$ and $K_i = j$ as

$$\Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\}.$$

Even though K_i is not observable in our masked data model, we can still consider the joint distribution of T_i, K_i, and $\mathcal{C}_i$. By Theorem 2.4, the joint pdf of T_i and K_i is given by

$$f_{T_i,K_i}(t_i, j; \theta) = h_j(t_i; \theta_j) \prod_{l=1}^{m} R_l(t_i; \theta_l),$$

where $h_j(t_i; \theta_j)$ and $R_j(s_i; \theta_j)$ are respectively the hazard and reliability functions of the j^{th} component. Thus, the joint pdf of T_i, K_i, and $\mathcal{C}_i$ may be written as

$$\begin{aligned} f_{T_i,K_i,\mathcal{C}_i}(t_i, j, c_i; \theta) &= f_{T_i,K_i}(t_i, k; \theta) \Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\} \\ &= h_j(t_i; \theta_j) \prod_{l=1}^{m} R_l(t_i; \theta_l) \Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\}. \end{aligned} \tag{12}$$

We are going to need the joint pdf of T_i and $\mathcal{C}_i$, which may be obtained by summing over the support $\{1, \ldots, m\}$ of K_i in Equation (12),

$$f_{T_i,\mathcal{C}_i}(t_i, c_i; \theta) = \prod_{l=1}^{m} R_l(t_i; \theta_l) \sum_{j=1}^{m} \left\{ h_j(t_i; \theta_j) \Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\} \right\}.$$

By Condition 1, $\Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\} = 0$ when $K_i = j$ and $j \notin c_i$, and so we may rewrite the joint pdf of T_i and $\mathcal{C}_i$ as

$$f_{T_i,\mathcal{C}_i}(t_i, c_i; \theta) = \prod_{l=1}^{m} R_l(t_i; \theta_l) \sum_{j \in c_i} \left\{ h_j(t_i; \theta_j) \Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\} \right\}. \tag{13}$$

When we try to find an MLE of θ (see Section 4), we solve the simultaneous equations of the MLE and choose a solution $\hat{\theta}$ that is a maximum for the likelihood function. When we do this, we find that $\hat{\theta}$ depends on the unknown conditional pmf $\Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\}$. So, we are motivated to seek out more conditions (that approximately hold in realistic situations) whose MLEs are independent of the pmf $\Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\}$.

Condition 2. Given an observed system failure time $T_i = t_i$ and candidate set c_i, the probability of the candidate set is the same when we condition on any component cause of failure in the candidate set. That is,

$$\Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j'\} = \Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\}$$

for all $j, j' \in c_i$.

According to [8], in many industrial problems, masking generally occurred due to time constraints and the expense of failure analysis. In this setting, Condition 2 generally holds.

Assuming Conditions 1 and 2, $\Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\}$ may be factored out of the summation in Equation (13), and thus the joint pdf of T_i and $\mathcal{C}_i$ may be rewritten as

$$f_{T_i,\mathcal{C}_i}(t_i, c_i; \theta) = \Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j'\} \prod_{l=1}^{m} R_l(t_i; \theta_l) \sum_{j \in c_i} h_j(t_i; \theta_j)$$

where $j' \in c_i$.

If $\Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j'\}$ is a function of θ, the MLEs are still dependent on the unknown $\Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j'\}$. This is a more tractable problem, but we are primarily interested in the situation where we do not need to know (nor estimate) $\Pr_\theta\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j'\}$ to find an MLE of θ. The last condition we assume achieves this result.

Condition 3. The masking probabilities conditioned on failure time T_i and component cause of failure K_i are not functions of θ. In this case, the conditional probability of $\mathcal{C}_i$ given $T_i = t_i$ and $K_i = j'$ is denoted by

$$\beta_i = \Pr\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j'\}$$

where β_i is not a function of θ.

When Conditions 1, 2, and 3 are satisfied, the joint pdf of T_i and $\mathcal{C}_i$ is given by

$$f_{T_i,\mathcal{C}_i}(t_i, c_i; \theta) = \beta_i \prod_{l=1}^{m} R_l(t_i; \theta_l) \sum_{j \in c_i} h_j(t_i; \theta_j).$$

When we fix the sample and allow θ to vary, we obtain the contribution to the likelihood L from the i^{th} observation when the system lifetime is exactly known (i.e., $\delta_i = 1$) but the component cause of failure is masked by a candidate set c_i:

$$L_i(\theta) \propto \prod_{l=1}^{m} R_l(t_i; \theta_l) \sum_{j \in c_i} h_j(t_i; \theta_j), \tag{14}$$

where we dropped the factor β_i since it is not a function of θ.[4]

To summarize this result, assuming Conditions 1, 2, and 3, if we observe an exact system failure time for the i-th system ($\delta_i = 1$), but the component that failed is masked by a candidate set c_i, then its likelihood contribution is given by Equation (14).

3.2 Right-Censored Data

As described in Section 3, we observe realizations of $(S_i, \delta_i, \mathcal{C}_i)$ where $S_i = \min\{T_i, \tau\}$ is the right-censored system lifetime, $\delta_i = 1_{T_i < \tau}$ is the event indicator, and $\mathcal{C}_i$ is the candidate set.

In the previous section, we discussed the likelihood contribution from an observation of a masked component cause of failure, i.e., $\delta_i = 1$. We now derive the likelihood contribution of a *right-censored* observation, $\delta_i = 0$, in our masked data model.

[4]When doing maximum likelihood estimation, we are interested in the parameter values that maximize the likelihood function. Since β_i is not a function of θ, it does not affect the location of the maximum of the likelihood function, and so we can drop it from the likelihood function.

Theorem 3.2. *The likelihood contribution of a right-censored observation ($\delta_i = 0$) is given by*

$$L_i(\theta) = \prod_{l=1}^{m} R_l(s_i; \theta_l). \tag{15}$$

Proof. When right-censoring occurs, then $S_i = \tau$, and we only know that $T_i > \tau$, and so we integrate over all possible values that it may have obtained,

$$L_i(\theta) = \Pr_\theta\{T_i > s_i\}.$$

By definition, this is just the survival or reliability function of the series system evaluated at s_i,

$$L_i(\theta) = R_{T_i}(s_i; \theta) = \prod_{l=1}^{m} R_l(t_i; \theta_l).$$

□

When we combine the two likelihood contributions, we obtain the likelihood contribution for the i^{th} system shown in Theorem 3.1,

$$L_i(\theta) \propto \begin{cases} \prod_{l=1}^{m} R_l(s_i; \theta_l) & \text{if } \delta_i = 0 \\ \prod_{l=1}^{m} R_l(t_i; \theta_l) \sum_{j \in c_i} h_j(s_i; \theta_j) & \text{if } \delta_i = 1. \end{cases}$$

We use this result in Section 4 to derive the maximum likelihood estimator (MLE) of θ.

3.3 Identifiability and Convergence Issues

In our likelihood model, masking and right-censoring can lead to issues related to identifiability and flat likelihood regions. Identifiability refers to the unique mapping of the model parameters to the likelihood function, and lack of identifiability can lead to multiple sets of parameters that explain the data equally well, making inference about the true parameters challenging [11], while flat likelihood regions can complicate convergence [13].

In our simulation study, we address these challenges in a pragmatic way. Specifically, failure to converge to a solution within a maximum of 125 iterations is interpreted as evidence of the aforementioned issues, leading to the discarding of the sample, with the process then repeated with a new synthetic sample. Note, however, that in Section 5 where we discuss the bias-corrected and accelerated (BCa) bootstrap method for constructing confidence intervals, we do not discard any resamples.

This strategy helps ensure the robustness of the results, while acknowledging the inherent complexities of likelihood-based estimation in models characterized by masking and right-censoring.

4 Maximum Likelihood Estimation

In our analysis, we use maximum likelihood estimation (MLE) to estimate the series system parameter θ from the masked data [3, 5]. The MLE finds parameter values that maximize the likelihood of the observed data under the assumed model. A maximum likelihood estimate, $\widehat{\theta}$, is a solution of

$$L(\widehat{\theta}) = \max_{\theta \in \Omega} L(\theta), \tag{16}$$

where $L(\theta)$ is the likelihood function of the observed data. For computational efficiency and analytical simplicity, we work with the log-likelihood function, denoted as $\ell(\theta)$, instead of the likelihood function [5].

Theorem 4.1. *The log-likelihood function, $\ell(\theta)$, for our masked data model is the sum of the log-likelihoods for each observation,*

$$\ell(\theta) = \sum_{i=1}^{n} \ell_i(\theta), \tag{17}$$

where $\ell_i(\theta)$ is the log-likelihood contribution for the i^{th} observation:

$$\ell_i(\theta) = \sum_{j=1}^{m} \log R_j(s_i;\theta_j) + \delta_i \log\Bigg(\sum_{j\in c_i} h_j(s_i;\theta_j)\Bigg). \tag{18}$$

Proof. The log-likelihood function is the logarithm of the likelihood function,

$$\ell(\theta) = \log L(\theta) = \log \prod_{i=1}^{n} L_i(\theta) = \sum_{i=1}^{n} \log L_i(\theta).$$

Substituting $L_i(\theta)$ from Equation (11), we consider these two cases of δ_i seperately to obtain the result in Theorem 4.1. **Case 1**: If the i-th system is right-censored ($\delta_i = 0$),

$$\ell_i(\theta) = \log \prod_{l=1}^{m} R_l(s_i;\theta_l) = \sum_{l=1}^{m} \log R_l(s_i;\theta_l).$$

Case 2: If the i-th system's component cause of failure is masked but the failure time is known ($\delta_i = 1$),

$$\ell_i(\theta) = \sum_{l=1}^{m} \log R_l(t_i;\theta_l) + \log\beta_i + \log\Bigg(\sum_{j\in c_i} h_j(s_i;\theta_j)\Bigg).$$

By Condition 3, we may discard the $\log\beta_i$ term since it does not depend on θ, giving us the result

$$\ell_i(\theta) = \sum_{l=1}^{m} \log R_l(s_i;\theta_l) + \log\Bigg(\sum_{j\in c_i} h_j(s_i;\theta_j)\Bigg).$$

Combining these two cases gives us the result in Theorem 4.1. □

The MLE, $\hat{\theta}$, is often found by solving a system of equations derived from setting the derivative of the log-likelihood function to zero [3], i.e.,

$$\frac{\partial}{\partial\theta_j}\ell(\theta) = 0, \tag{19}$$

for each component θ_j of the parameter θ. When there's no closed-form solution, we resort to numerical methods like the Newton-Raphson method.

Assuming some regularity conditions, such as the likelihood function being identifiable, the MLE has many desirable asymptotic properties that underpin statistical inference, namely that it is an asymptotically unbiased estimator of the parameter θ and it is normally distributed with a variance given by the inverse of the Fisher Information Matrix (FIM) [5]. However, for smaller samples, these asymptotic properties may not yield accurate approximations. We propose to use the bootstrap method to offer an empirical approach for estimating the sampling distribution of the MLE, in particular for computing confidence intervals.

5 Bias-Corrected and Accelerated Bootstrap Confidence Intervals

We utilize the non-parametric bootstrap to approximate the sampling distribution of the MLE. In the non-parametric bootstrap, we resample from the observed data with replacement to generate a bootstrap sample. The MLE is then computed for the bootstrap sample. This process is repeated B times, giving us B bootstrap replicates of the MLE. The sampling distribution of the MLE is then approximated by the empirical distribution of the bootstrap replicates of the MLE.

The method we use to generate confidence intervals is known as Bias-Corrected and Accelerated Bootstrap Confidence Intervals (BCa), which applies two corrections to the standard bootstrap method:

- Bias correction: This adjusts for bias in the bootstrap distribution itself. This bias is measured as the difference between the mean of the bootstrap distribution and the observed statistic. It works by transforming the percentiles of the bootstrap distribution to correct for these issues.

  This may be a useful transformation in our case since we are dealing with small samples and we have two potential sources of bias: right-censoring and masking component cause of failure. They seem to have opposing effects on the MLE, but the relationship is difficult to quantify.

- Acceleration: This adjusts for the rate of change of the statistic as a function of the true, unknown parameter. This correction is important when the shape of the statistic's distribution changes with the true parameter.

  Since we have a number of different shape parameters, $k_1, \ldots, k_m$, we may expect the shape of the distribution of the MLE to change as a function of the true parameter, making this correction potentially useful.

Since we are primarly interested in generating confidence intervals for small samples for a potentially biased MLE, the BCa method may be a good choice for our analysis. For more details on BCa, see [6].

In our simulation study, we will assess the performance of the bootstrapped BCa confidence intervals by computing the coverage probability of the confidence intervals. A well-calibrated 95% confidence interval contains the true value around 95% of the time. If the confidence interval is too narrow, it will have a coverage probability less than 95%, which conveys a sort of false confidence in the precision of the MLE. If the confidence interval is too wide, it will have a coverage probability greater than 95%, which conveys a lack of confidence in the precision of the MLE. We want confidence intervals to be as narrow as possible while still having a coverage probability close to the nominal level, 95%.

While the bootstrap method provides a robust and flexible tool for statistical estimation, its effectiveness can be influenced by several factors [7]. Firstly, instances of non-convergence in our bootstrap samples were observed. Such cases can occur when the estimation method, like the MLE used in our analysis, fails to converge due to the specifics of the resampled data [5]. This issue can potentially introduce bias or reduce the effective sample size of our bootstrap distribution.

Secondly, the bootstrap's accuracy can be compromised with small sample sizes, as the method relies on the law of large numbers to approximate the true sampling distribution. For small datasets, the bootstrap samples might not adequately represent the true variability in the data, leading to inaccurate results [7].

Thirdly, our data involves right censoring and a masking of the component cause of failure when a system failure is observed. These aspects can cause certain data points or trends to be underrepresented or not represented at all in our data, introducing bias in the bootstrap distribution [10].

Despite these challenges, we found the bootstrap method useful in approximating the sampling distribution of the MLE, taking care in interpreting the results, particularly as it relates to coverage probabilities.

6 Series System with Weibull Components

The Weibull distribution, introduced by Waloddi Weibull in 1937, has been instrumental in reliability analysis due to its ability to model a wide range of failure behaviors. Reflecting on its utility, Weibull modestly noted that it "[...] may sometimes render good service." [1]. In the context of our study, we utilize the Weibull to model a system as originating from Weibull components in a series configuration, producing a specific form of the likelihood model described in Section 3, which deals with challenges such as right censoring and masked component cause of failure.

The j^{th} component of the i^{th} has a lifetime distribution given by

$$T_{ij} \sim \text{Weibull}(k_j, \lambda_j) \qquad \text{for } i = 1, \ldots, n \text{ and } j = 1, \ldots, m,$$

where $\lambda_j > 0$ is the scale parameter and $k_j > 0$ is the shape parameter. The j^{th} component has a reliability function, pdf, and hazard function given respectively by

$$R_j(t; \lambda_j, k_j) = \exp\left\{-\left(\frac{t}{\lambda_j}\right)^{k_j}\right\}, \tag{20}$$

$$f_j(t; \lambda_j, k_j) = \frac{k_j}{\lambda_j}\left(\frac{t}{\lambda_j}\right)^{k_j - 1} \exp\left\{-\left(\frac{t}{\lambda_j}\right)^{k_j}\right\}, \tag{21}$$

$$h_j(t; \lambda_j, k_j) = \frac{k_j}{\lambda_j}\left(\frac{t}{\lambda_j}\right)^{k_j - 1}. \tag{22}$$

The shape parameter of the Weibull distribtion is of particular importance:

- $k_j < 1$ indicates infant mortality. An example of how this might arise is a result of defective components being weeded out early, and the remaining components surviving for a much longer time.
- $k_j = 1$ indicates random failures (independent of age). An example of how this might arise is a result of random shocks to the system, but otherwise the system is age-independent.[5]
- $k_j > 1$ indicates wear-out failures. An example of how this might arise is a result of components wearing as they age

We show that the lifetime of the series system composed of m Weibull components has a reliability, hazard, and probability density functions given by the following theorem.

Theorem 6.1. *The lifetime of a series system composed of m Weibull components has a reliability function, hazard function, and pdf respectively given by*

$$R_{T_i}(t;\theta) = \exp\left\{-\sum_{j=1}^{m}\left(\frac{t}{\lambda_j}\right)^{k_j}\right\}, \tag{23}$$

$$h_{T_i}(t;\theta) = \sum_{j=1}^{m}\frac{k_j}{\lambda_j}\left(\frac{t}{\lambda_j}\right)^{k_j-1}, \tag{24}$$

$$f_{T_i}(t;\theta) = \left\{\sum_{j=1}^{m}\frac{k_j}{\lambda_j}\left(\frac{t}{\lambda_j}\right)^{k_j-1}\right\}\exp\left\{-\sum_{j=1}^{m}(\frac{t}{\lambda_j})^{k_j}\right\}. \tag{25}$$

Proof. The proof for the reliability function follows from Theorem 2.1,

$$R_{T_i}(t;\theta) = \prod_{j=1}^{m} R_j(t;\lambda_j,k_j).$$

Plugging in the Weibull component reliability functions obtains the result

$$R_{T_i}(t;\theta) = \prod_{j=1}^{m}\exp\left\{-\left(\frac{t}{\lambda_j}\right)^{k_j}\right\} = \exp\left\{-\sum_{j=1}^{m}\left(\frac{t}{\lambda_j}\right)^{k_j}\right\}.$$

The proof for the hazard function follows from Theorem 2.3,

$$h_{T_i}(t;\theta) = \sum_{j=1}^{m} h_j(t;\theta_j) = \sum_{j=1}^{m}\frac{k_j}{\lambda_j}\left(\frac{t}{\lambda_j}\right)^{k_j-1}$$

The proof for the pdf follows from Theorem 2.2. By definition,

$$f_{T_i}(t;\theta) = h_{T_i}(t;\theta)R_{T_i}(t;\theta).$$

Plugging in the failure rate and reliability functions given respectively by Equations (23) and (24) completes the proof. □

In Section 2.2, we discussed the concept of reliability, with the MTTF being a common measure of reliability. In the case of Weibull components, the MTTF of the j^{th} component is given by

$$\text{MTTF}_j = \lambda_j\Gamma\left(1+\frac{1}{k_j}\right), \tag{26}$$

where Γ is the gamma function.

We mentioned that the MTTF can sometimes be a poor measure of reliability, e.g., the MTTF and the probability of failing early can be large. The Weibull is a good example of this phenomenon. If $k_j > 1$, the j^{th} component is fat-tailed and it can exhibit both a large MTTF and a high probability of failing early. The probability of component failure given by Equation (7) is a particularly useful measure of component reliability relative to the other components in the system.

[5]The exponential distribution is a special case of the Weibull distribution when $k_j = 1$.

6.1 Likelihood Model

In Section 3, we discussed two separate kinds of likelihood contributions, masked component cause of failure data (with exact system failure times) and right-censored data. The likelihood contribution of the i^{th} system is given by the following theorem.

Theorem 6.2. *Let δ_i be an indicator variable that is 1 if the i^{th} system fails and 0 (right-censored) otherwise. Then the likelihood contribution of the i^{th} system is given by*

$$L_i(\theta) \propto \begin{cases} \exp\left\{-\sum_{j=1}^{m}\left(\frac{t_i}{\lambda_j}\right)^{k_j}\right\} \sum_{j\in c_i} \frac{k_j}{\lambda_j}\left(\frac{t_i}{\lambda_j}\right)^{k_j-1} & \text{if } \delta_i = 1, \\ \exp\left\{-\sum_{j=1}^{m}\left(\frac{t_i}{\lambda_j}\right)^{k_j}\right\} & \text{if } \delta_i = 0. \end{cases} \tag{27}$$

Proof. By Theorem 3.1, the likelihood contribution of the i-th system is given by

$$L_i(\theta) \propto \begin{cases} R_{T_i}(s_i;\theta) & \text{if } \delta_i = 0 \\ R_{T_i}(s_i;\theta)\sum_{j\in c_i} h_j(s_i;\theta_j) & \text{if } \delta_i = 1. \end{cases}$$

By Equation (23), the system reliability function R_{T_i} is given by

$$R_{T_i}(t_i;\theta) = \exp\left\{-\sum_{j=1}^{m}\left(\frac{t_i}{\lambda_j}\right)^{k_j}\right\}.$$

and by Equation (22), the Weibull component hazard function h_j is given by

$$h_j(t_i;\theta_j) = \frac{k_j}{\lambda_j}\left(\frac{t_i}{\lambda_j}\right)^{k_j-1}.$$

Plugging these into the likelihood contribution function obtains the result. □

Taking the log of the likelihood contribution function obtains the following result.

Corollary 6.1. *The log-likelihood contribution of the i-th system is given by*

$$\ell_i(\theta) = -\sum_{j=1}^{m}\left(\frac{t_i}{\lambda_j}\right)^{k_j} + \delta_i \log\left(\sum_{j\in c_i}\frac{k_j}{\lambda_j}\left(\frac{t_i}{\lambda_j}\right)^{k_j-1}\right) \tag{28}$$

where we drop any terms that do not depend on θ since they do not affect the MLE.

See Appendix A.1 for the R code that implements the log-likelihood function for the series system with Weibull components.

We find an MLE by solving (19), i.e., a point $\hat{\theta} = (\hat{k}_1, \hat{\lambda}_1, \ldots, \hat{k}_m, \hat{\lambda}_m)$ satisfying $\nabla_\theta \ell(\hat{\theta}) = 0$, where ∇_θ is the gradient of the log-likelihood function (score) with respect to θ. To solve this system of equations, we use the Newton-Raphson method, which requires the score and the Hessian of the log-likelihood function. We analytically derive the score since it is useful to have for the Newton-Raphson method, but we do not do the same for the Hessian of the log-likelihood for the following reasons:

- The gradient is easy to derive, and it is useful to have for computing gradients efficiently and accurately, which will be useful for numerically approximating the Hessian.
- The Hessian is tedious and error prone to derive, and Newton-like methods often do not require the Hessian to be explicitly computed.

The following theorem derives the score function.

Theorem 6.3. *The gradient of the log-likelihood contribution of the i-th system is given by*

$$\nabla \ell_i(\theta) = \left(\frac{\partial \ell_i(\theta)}{\partial k_1}, \frac{\partial \ell_i(\theta)}{\partial \lambda_1}, \ldots, \frac{\partial \ell_i(\theta)}{\partial k_m}, \frac{\partial \ell_i(\theta)}{\partial \lambda_m} \right)', \tag{29}$$

where

$$\frac{\partial \ell_i(\theta)}{\partial k_r} = -\left(\frac{t_i}{\lambda_r}\right)^{k_r} \log\left(\frac{t_i}{\lambda_r}\right) + \frac{\frac{1}{t_i}\left(\frac{t_i}{\lambda_r}\right)^{k_r}\left(1 + k_r \log\left(\frac{t_i}{\lambda_r}\right)\right)}{\sum_{j \in c_i} \frac{k_j}{\lambda_j}\left(\frac{t_i}{\lambda_j}\right)^{k_j - 1}} 1_{\delta_i = 1 \wedge r \in c_i} \tag{30}$$

and

$$\frac{\partial \ell_i(\theta)}{\partial \lambda_r} = \frac{k_r}{\lambda_r}\left(\frac{t_i}{\lambda_r}\right)^{k_r} - \frac{\left(\frac{k_r}{\lambda_r}\right)^2\left(\frac{t_i}{\lambda_r}\right)^{k_r - 1}}{\sum_{j \in c_i} \frac{k_j}{\lambda_j}\left(\frac{t_i}{\lambda_j}\right)^{k_j - 1}} 1_{\delta_i = 1 \wedge r \in c_i} \tag{31}$$

The result follows from taking the partial derivatives of the log-likelihood contribution of the i-th system given by Equation (27). It is a tedious calculation so the proof has been omitted, but the result has been verified by using a very precise numerical approximation of the gradient.

By the linearity of differentiation, the gradient of a sum of functions is the sum of their gradients, and so the score function conditioned on the entire sample is given by

$$\nabla \ell(\theta) = \sum_{i=1}^{n} \nabla \ell_i(\theta). \tag{32}$$

6.2 Weibull Series System: Homogeneous Shape Parameters

In a series system, the system is only as reliable as its weakest link (weakest component). In a well-designed series system, there is no single component that is much weaker than the others. In the case of components with Weibull lifetimes, this implies the shape parameters are homogenous and the scale parameters are homogenous. The shape parameters are particularly important since they determine the failure behavior of the components.

When the shape parameters are homogenous, the lifetime of the series system with components that are Weibull distributed is also Weibull distributed, as shown in the following theorem.

Theorem 6.4. *If the shape parameters of the components are homogenous, then the lifetime series system follows a Weibull distribution with a shape parameter k given by the identical shape parameters of the components and a scale parameter λ given by*

$$\lambda = \left(\sum_{j=1}^{m} \lambda_j^{-k} \right)^{-1/k}, \tag{33}$$

where λ_j is the scale parameter of the j^{th} component.

Proof. Given m Weibull lifetimes $T_{i1}, \ldots, T_{im}$ with the same shape parameter k and scale parameters $\lambda_1, \ldots, \lambda_m$, the reliability function of the series system is given by

$$R_{T_i}(t; \theta) = \exp\left\{ -\sum_{j=1}^{m} \left(\frac{t}{\lambda_j}\right)^k \right\}.$$

To show that the series system lifetime is Weibull, we need to find a single scale parameter λ such that

$$R_{T_i}(t; \theta) = \exp\left\{ -\left(\frac{t}{\lambda}\right)^k \right\},$$

which has the solution

$$\lambda = \frac{1}{\left(\frac{1}{\lambda_1^k} + \ldots + \frac{1}{\lambda_m^k}\right)^{\frac{1}{k}}}.$$

□

Theorem 6.5. *If a series system has Weibull components with homogeneous shape parameters, the component cause of failure is conditionally independent of the system failure time:*

$$\Pr\{K_i = j | T_i = t_i\} = \Pr\{K_i = j\} = \frac{\lambda_j^{-k}}{\sum_{l=1}^{m} \lambda_l^{-k}}.$$

Proof. By Theorem 2.6, the conditional probability of the j^{th} component being the cause of failure given the system failure time is given by

$$\begin{aligned}\Pr\{K_i = j | T_i = t\} &= \frac{f_{K_i,T_i}(j,t;\theta)}{f_{T_i}(t;\theta)} = \frac{h_j(t;k,\lambda_j)R_{T_i}(t;\theta)}{h_{T_i}(t;\theta_j)R_{T_i}(t;\theta)} \\ &= \frac{h_j(t;k,\lambda_j)}{\sum_{l=1}^{m} h_l(t;k,\lambda_l)} = \frac{\frac{k}{\lambda_j}\left(\frac{t}{\lambda_j}\right)^{k-1}}{\sum_{l=1}^{m} \frac{k}{\lambda_l}\left(\frac{t}{\lambda_l}\right)^{k-1}} = \frac{\left(\frac{1}{\lambda_j}\right)^{k}}{\sum_{l=1}^{m}\left(\frac{1}{\lambda_l}\right)^{k}}.\end{aligned}$$

□

According to the bias-variance trade-off, we expect the MLE of the homogenous model, which has $m+1$ parameters (m being he number of components in the series system), to be more biased but have less variance than the MLE of the full model, which has $2m$ parameters.

7 Simulation Study: Series System with Weibull Components

In this simulation study, we assess the sensitivity of the MLE and BCa confidence intervals to various simulation scenarios for the likelihood model defined in Section 6. We begin by specifying the parameters of the series system that will be the central object of our simulation study. We consider the data in [9], in which they study the reliability of a series system with three components. They fit Weibull components in a series configuration to the data, resulting in an MLE with shape and scale estimates given by the first three components in Table 2. To make the model slightly more complex, we add two more components to this series system, with shape and scale parameters given by the last two components in Table 2. We will refer to this system as the **base** system.

In Section 2.2, we defined a well-designed series system as one that consists of components with similar reliabilities, where we define reliability in three ways: the reliability function, MTTF, and probability that a specific component will be the cause of failure (which is a measure of relative reliability of the components). We will use these three measures of reliability to assess the base system. To a first approximation, the base system defined in Table 2 satisfies this definition of being a well-designed system. We see that there are no components that are significantly less reliable than any of the others, component 1 being the most reliable and component 3 being the least reliable.

The reliability function, unlike the other two measures of reliability, is not a summary statistic of reliability, but is rather a function of time. Since most of our simulations has the right-censoring time set to the 82.5% quantile of the series system, which we denote here by $\tau_{0.825}$, we can compare the reliability functions of the components at this time. We see that the reliability of the components at this right-censoring time are similar, with component 1 being the most reliable and component 3 being the least reliable. These results are consistent with the previous analysis based on the MTTF and probability of component cause of failure being similar.

The shape parameters for each component is larger than 1, which means each component has a failure characteristic that is more wear-out than infant mortality. The system as a whole is therefore more likely to fail due to wear-out failures than infant mortality. This too is consistent with a well-designed system.

Homogenous Shape Parameters The base system is a well-designed system, and so it is likely that the likelihood model that assumes homogeneous shape parameters described in Section 6.2 would provide a good fit to any data generated from this system. We performed a preliminary investigation into this by simulating data from the base system, and deviations from the base system that make the system less well-designed, and fitting the homogeneous model to the data. We found that the MLE of the homogeneous model was

Table 2: Weibull Components in Series Configuration

	Shape (k_j)	Scale (λ_j)	MTTF$_j$	Pr$\{K_i = j\}$	$R_j(\tau_{0.825}; k_j, \lambda_j)$
Component 1	1.2576	994.3661	924.869	0.169	0.744
Component 2	1.1635	908.9458	862.157	0.207	0.698
Component 3	1.1308	840.1141	803.564	0.234	0.667
Component 4	1.1802	940.1342	888.237	0.196	0.711
Component 5	1.2034	923.1631	867.748	0.195	0.711
Series System	NA	NA	222.884	NA	0.175

very close to the true parameter values for slight deviations from the base system, but the MLE was biased for larger deviations from the base system. This is consistent with the bias-variance trade-off, where the MLE of the homogeneous model is more biased but has less variance than the MLE of the full model. We do not explore this furhter in this simulation study, but it is an interesting avenue for future research.

7.1 Performance Metrics and Convergence Rate

In this section, we describe the measures we use to assess the performance of the MLE and the BCa confidence intervals in our simulation study. We assess two important properties of the MLE for each simulation scenario:

- **Accuracy (Bias)**: How close is the expected value of the MLE to the true parameter values? If the expected value of the MLE is close to the true parameter values, the accuracy is high. We measure this in our simulation study by plotting the mean of the MLE, which gives us a sense of the bias of the MLE when compared to the true parameter values.
- **Precision**: How much does the MLE vary from sample to sample? If the MLE varies little from sample to sample, the precision is high. We measure this in our simulation study by plotting the quantiles of the MLE distribution, which gives us a sense of the variability of the MLE.

In addition, we assess the following properties of the BCa confidence intervals described in Section 5:

- **Accuracy (Coverage Probability)**: How often do the computed confidence intervals contain the true parameter values? If the confidence intervals contain the true parameter values around 95% of the time, the confidence intervals are said to be well-calibrated.
- **Precision**: How wide are the confidence intervals? If the confidence intervals vary little from sample to sample, the precision is high. We measure this in our simulation study by plotting the IQR of the confidence intervals.

If the confidence intervals are both accurate and precise, then we can have high confidence that it is close to the true parameter values.

Convergence Rate In Section 3.3, we discussed convergence issues with the MLE. We discard any simulation scenario where the MLE did not converge to a local maximum of the log-likelihood function within 125 iterations. In Figure 7.1, we plot the convergence rate of the MLE for each simulation scenario. The convergence rate is defined as the proportion of simulations where the MLE converged to a local maximum.

The MLE converged to a local maximum in most simulation scenarios with very high probability. However, when we have significant masking ($p \geq 0.55$), significant right censoring ($q = 0.6$), or very small samples ($n = 50$), the MLE may not converge to a local maximum. We take this as evidence that the MLE is not a robust or reliable estimator in these extreme cases, due to the lack of information in the data, and under these conditions, the MLE should be taken with a grain of salt.

7.2 Data Generating Process

In this section, we describe the data generating process for our simulation studies. It consists of three parts: the series system, the candidate set model, and the right-censoring model.

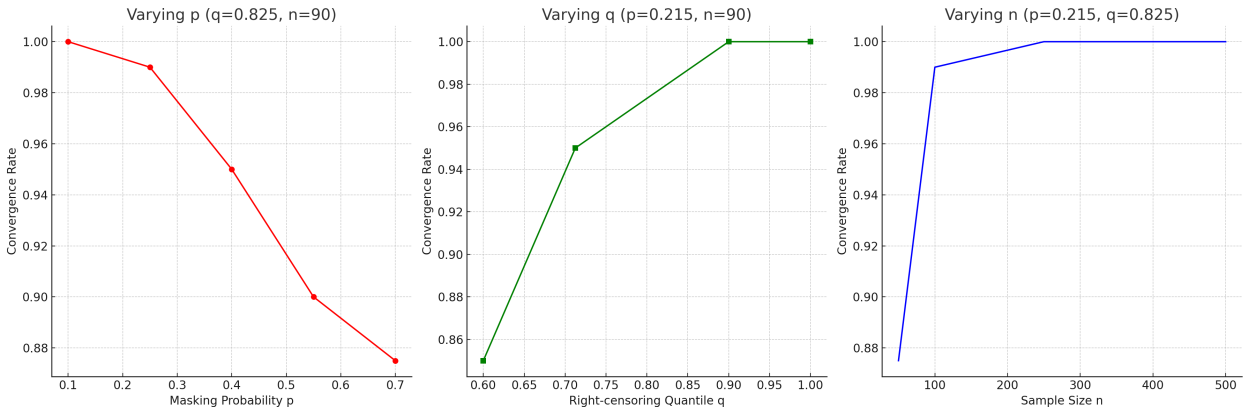

Figure 2: Simulation Scenarios vs Convergence Rate

Series System Lifetime We generate data from a Weibull series system with m components. As described in Section 6, the j^{th} component of the i^{th} system has a lifetime distribution given by

$$T_{ij} \sim \text{Weibull}(k_j, \lambda_j)$$

and the lifetime of the series system composed of m Weibull components is defined as

$$T_i = \min\{T_{i1}, \ldots, T_{im}\}.$$

To generate a data set, we first generate the m component failure times, by efficiently sampling from their respective distributions, and we then set the failure time t_i of the system to the minimum of the component failure times.

Right-Censoring Model We employ a simple right-censoring model, where the right-censoring time τ is fixed at some known value, e.g., an experiment is run for a fixed amount of time τ, and all systems that have not failed by the end of the experiment are right-censored. The censoring time S_i of the i^{th} system is thus given by

$$S_i = \min\{T_i, \tau\}.$$

So, after we generate the system failure time T_i, we generate the censoring time S_i by taking the minimum of T_i and τ. In our simulation study, we paramaterize the right-censoring time τ by the quantile $q = 0.825$ of the series system,

$$\tau = F_{T_i}^{-1}(q).$$

This means that 82.5% of the series systems are expected to fail before time τ and 17.5% of the series are expected to be right-censored. To solve for the 82.5% quantile of the series system, we define the function g as

$$g(\tau) = F_{T_i}(\tau; \theta) - q$$

and find its root using the Newton-Raphson method. See Appendix A.3 for the R code that implements this procedure.

Masking Model for Component Cause of Failure We must generate data that satisfies the masking conditions described in Section 3.1. There are many ways to satisfying the masking conditions. We choose the simplest method, which we call the *Bernoulli candidate set model*. In this model, we satisfy Conditions 1, 2, and 3 by generating a candidate set c_i for each system i as follows:

- If the j^{th} component fails, it is deterministically placed in the candidate set. This satisfies Condition 1, $\Pr\{K_i \in \mathcal{C}_i\} = 1$.

- For each of the $m-1$ components that did not fail, we generate a Bernoulli random variable X_j with probability p of being 1, where p is a fixed probability. If $X_j = 1$, the j^{th} component is placed in the candidate set, which satisfies Condition 3, since the Bernoulli random variables are not a function of θ.

- Condition 2 may be the least intuitive of the three conditions. It states that

$$\Pr\{\mathcal{C}_i = c_i | K_i = j, T_i = t_i\} = \Pr\{\mathcal{C}_i = c_i | K_i = j', T_i = t_i\}$$

for all $j, j' \in c_i$. In words, this means that the probability of the candidate set $\mathcal{C}_i$ being equal to some set c_i is the same when conditioned on any component in c_i being the cause of failure and when condition on the given system failure time t_i. This is satisfied by the Bernoulli masking model since, first, it is independent of the system failure time t_i, and second, the probability that a non-failed component is in the candidate set is fixed at p for all components, and p in this case does not vary with the component cause of failure

For example, suppose we have a system with $m = 3$ components, and the first component fails:

$$\Pr\{\mathcal{C}_i = c_i | K_i = 1, T_i = t_i\} = \begin{cases} (1-p)^2 & \text{if } c_i = \{1\} \\ p(1-p) & \text{if } c_i = \{1,2\} \\ (1-p)p & \text{if } c_i = \{1,3\} \\ p^2 & \text{if } c_i = \{1,2,3\}, \end{cases} \tag{34}$$

where a non-failed component is in the candidate set with probability p and otherwise it is not placed in the candidate set with probability $1-p$. Now, let's change the component cause of failure to the second component:

$$\Pr\{\mathcal{C}_i = c_i | K_i = 2, T_i = t_i\} = \begin{cases} (1-p)^2 & \text{if } c_i = \{2\} \\ p(1-p) & \text{if } c_i = \{1,2\} \\ (1-p)p & \text{if } c_i = \{2,3\} \\ p^2 & \text{if } c_i = \{1,2,3\} \end{cases} \tag{35}$$

The same pattern holds for the third component. We see that the probability of the candidate set being equal to some set c_i is the same for all components in c_i and for all system failure times t_i.

There are many more ways to satisfy the masking conditions, but we choose the Bernoulli candidate set model because it is simple to understand and allows us to easily vary the masking probability p for our simulation study.

See Appendix B.2 for the R code that implements this model.

7.3 Overview of Simulations

We define a simulation scenario to be some combination of n (sample size), p (masking probability in our Bernoulli candidate set model), and q (right-censoring quantile). We are interested in choosing a small number of scenarios that are representative of real-world scenarios and that are interesting to analyze. For how we run a simulation scenario, see Appendix B.1, but here is an outline of the process:

1. **Parameter Initialization**: Fix a combination of simulation parameters to some value, and vary the remaining parameters. For example, if we want to assess how the sampling distribution of the MLE changes with respect to sample size, we might choose some particular values for p and q and then vary the sample size n over the desired range.

2. **Data Generation**: Simulate $R \geq 300$ datasets from the Data Generating Process (DGP) described in Section 7.2.

3. **MLE Computation**: Compute an MLE for each of the R datasets.

4. **Statistical Evaluation**: For each of these R MLEs, compute some function of the MLE, like the BCa confidence intervals. This will give us R statistics as a Monte-carlo estimate of the sampling distribution of the statistic.

5. **Distribution Property Estimation**: Use the R statistics to estimate some property of the sampling distribution of the statistic, e.g., the mean of the MLE or the coverage probability of the BCa confidence intervals, with respect to the parameter we are varying in the scenario, e.g., assess how the coverage probability of the BCa confidence intervals changes with respect to sample size.

In this study, we are focusing on three distinct scenarios. Section 7.4 explores how varying the right-censoring affects the estimator with the masking and sample size fixed. Section 7.5 explores how varying the masking affects the estimator with the right-censoring and sample size fixed. Section 7.6 explores how varying the sample affects the estimator with the right-censoring and masking fixed. Each scenario aims to provide insights into how these parameters influence the behavior of MLEs, which is crucial for understanding their performance in real-world applications.

7.4 Scenario: Assessing the Impact of Right-Censoring

In this scenario, we use the well-designed series system described in Table 2, and we vary the right-censoring quantile (q) from 60% to 100% (no right-censoring), with a component cause of failure masking probability of $p = 21.5\%$ and sample size $n = 100$.

In Figure 3, we show the effect of right-censoring on the MLEs for the shape and scale parameters. The top four plots only show the effect on the MLEs for the shape and scale parameters of components 1 and 4, since the rest were essentially identical, and the bottom two plots show the coverage probabilities for all parameters.

7.4.1 Background

When a right-censoring event occurs, in order to increase the likelihood of the data, the MLE is nudged in a direction that increases the probability of a right-censoring event at time τ, which is given by $R_{T_i}(t;\theta)$, representing a source of bias in the estimate.

To increase $R_{T_i}(\tau)$, we move in the direction (gradient) of these partial derivatives. The partial derivatives of $R_{T_i}(\tau)$ are given by

$$\frac{\partial R_{T_i}(\tau)}{\partial \lambda_j} = R_{T_i}(\tau;\theta)\left(\frac{\tau}{\lambda_j}\right)^{k_j}\frac{k_j}{\lambda_j},$$
$$\frac{\partial R_{T_i}(\tau)}{\partial k_j} = R_{T_i}(\tau;\theta)\left(\frac{\tau}{\lambda_j}\right)^{k_j}\left(\log\lambda_j - \log\tau\right),$$

for $j = 1, \ldots, m$. We see that these partial derivatives are related to the score of a right-censored likelihood contribution in Theorem 6.3. Let us analyze the implications these partial derivatives have on the MLE:

- **Effect of Increasing Right-Censoring Quantile**: As the right-censoring quantile q increases (τ increases), $R_{T_i}(\tau;\theta)$ decreases, reducing the impact of right-censoring on the MLE. This behavior is evident in Figure 3.

- **Positive Bias in Scale Parameters**: The partial derivatives with respect to the scale parameters are always positive. This means that right-censoring introduces a positive bias in the scale parameter estimates, making right-censoring events more likely. The extent of this bias is related to the amount of right-censoring $(1 - q)$, as seen in Figure 3.

- **Conditional Bias in Shape Parameters**: The partial derivative with respect to the shape parameter of the j^{th} component, k_j, is non-negative if $\lambda_j \geq \tau$ and otherwise negative. In our well-designed series system, the scale parameters are large compared to most of the right-censoring times for $\tau(q)$, so the

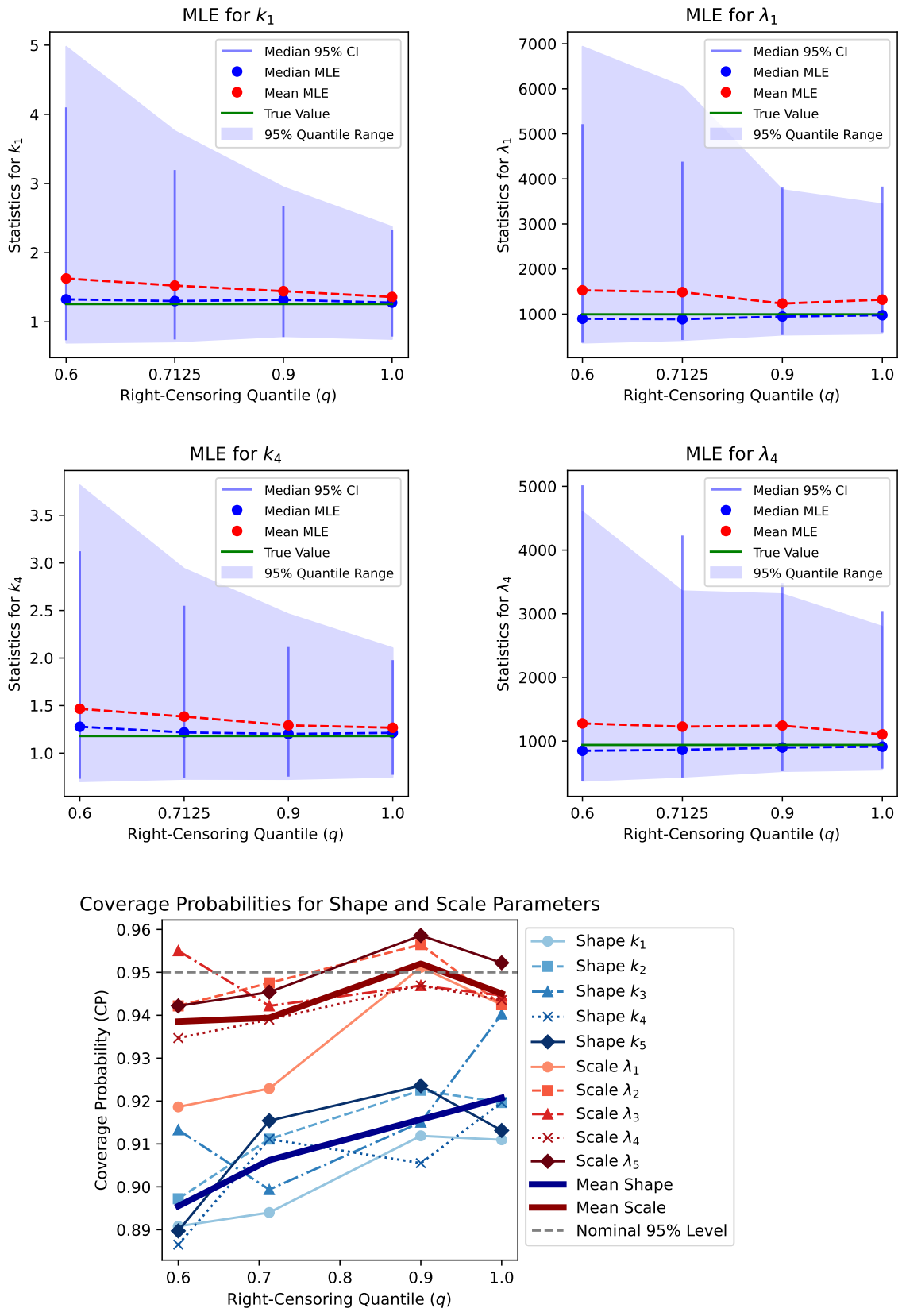

Figure 3: Right-Censoring Quantile vs MLE ($p = 0.215, n = 100$)

MLE nudges the shape parameter estimates in a positive direction to increase the probability of a right-censoring event $R_{T_i}(\tau)$ at time τ. We see this in Figure 3, where the shape parameter estimates are positively biased for most of the quantiles q.

7.4.2 Key Observations

Coverage Probability (CP) The confidence intervals are generally well-calibrated, obtaining coverages above 90% for most of the parameters across the entire range of right-censoring quantiles, and they are converging to the nominal 95% level as the right-censoring quantile increases. This suggests that the bootstrapped CIs will contain the true value of the parameters with the specified confidence level with high probability. The CIs are neither too wide nor too narrow.

However, the scale parameters are better calibrated than the shape parameters. The scale parameters are consistently around the nominal 95% level for all right-censoring quantiles, but the shape parameters are consistently less well-calibrated, suggesting that the shape parameters are more difficult to estimate than the scale parameters.

Dispersion of MLEs The shaded regions representing the 95% probability range of the MLEs get narrower as the right-censoring quantile increases. This is an indicator of the increased precision in the estimates as more data is available due to decreased censoring.

IQR of Bootstrapped CIs The IQR (vertical blue bars) decreaes in length as the right-censoring quantile increases. This suggests that the bootstrapped CIs are getting more consistent and focused around a narrower range as the right-censoring quantile increaes, while maintaining a good coverage probability. As right-censoring events become less likely, the bootstrapped CIs are more likely to be closer to each other and the true value of the parameters.

For small right-censoring quantiles (small right-censoring times), they are quite large, but to maintain well-calibrated CIs, this was necessary. The estimator is quite sensitive to the data, and so the bootstrapped CIs are quite wide to account for this sensitivity when the sample contains insufficient information due to censoring.

Bias of MLEs The red dashed line indicating the mean of MLEs initially is quite biased for the shape parameters, but quickly diminshes to neglible levels as the right-censoring quantile increases. The bias for the shape parameters never reach zero, but this is potentially due to masking.

The bias for the scale parameters is quite small and remains stable across different right-censoring quantiles, suggesting that the scale MLEs are reasonably unbiased. It could be the case that making or other factors are counteracting the bias due to right-censoring.

7.4.3 Summary

In this scenario, we find that right-censoring significantly influences the MLEs of shape and scale parameters. As the degree of right-censoring decreases, the precision of these estimates improves, and the bias in the shape parameters diminishes, although it never fully disappears, possibly due to masking effects. Additionally, Bootstrapped (BCa) confidence intervals are generally well-calibrated, particularly for scale parameters, and become more focused as right-censoring decreases. These insights set the stage for the subsequent scenario on masking probability, where we will explore how masking adds another layer of complexity to parameter estimation and how these effects might be mitigated.

7.5 Scenario: Assessing the Impact of Masking Probability for Component Cause of Failure

In this scenario, we use the well-designed series system described in Table 2. We fix the sample size to $n = 90$ (reasonable sample size) and we fix the right-censoring quantile to $q = 0.825$, and we vary the masking probability from p from 0.1 (very slight masking the component cause of failure) to 0.85 (extreme masking of the component cause of failure).

In Figure 4, we show the effect of the masking probability p on the MLE for the shape and scale parameters. The top four plots only show the effect on the MLEs for the the shape and scale parameters of components 1 and 4, since the rest were essentially identical, and the bottom two plots show the coverage probabilities for all parameters.

7.5.1 Background

Masking introduces a layer of complexity that is different from right-censoring. While right-censoring deals with the uncertainty in the timing of failure, masking adds ambiguity in identifying which component actually failed. In the Bernoulli masking model, the failed component is guaranteed to be in the candidate set and each non-failed component is included with a probability p. This has the following implications on the MLE:

- **Ambiguity**: A higher p makes larger candidate sets more probable, making it less clear for the MLE which parameters should be adjusted to make the data more likely. This is particularly problematic for components that are not the cause of failure, since the MLE will adjust their parameters to make them more likely to be the cause of failure, which is not necessarily correct.
- **Bias**: In an ideal scenario, knowing which component failed would allow the MLE to make that component's failure more likely and a right-censoring effect would be applied to the non-failed components. However, a larger p introduces uncertainty, making the MLE adjust parameters for non-failed components to be more likely to fail at the specified time. For a well-designed series system, this may be an too problematic, but if the system deviates from that design, the effect of masking is likely to be much more pronoucned source of bias.
- **Precision**: As p increases, the likelihood function becomes less informative, reducing the precision of the MLE estimates.

7.5.2 Key Observations

Coverage Probability (CP) For the scale parameters, the 95% CI is well-calibrated for Bernulli masking probabilities up to $p = 0.725$, which is really quite significant, obtaining coverages over 90%, but drops precipitously after that point. For the shape parameters, the 95% CI is well-calibrated for masking probabilities only up to $p = 0.4$, which is still large, obtaining coverages generally over 90, but begins to drop slowly after that point.

The BCa confidence intervals are well-calibrated for most realistic masking probabilities, constructing CIs that are neither too wide nor too narrow, but when the masking is severe and the sample size is small, one should take the CIs with a grain of salt.

Dispersion of MLEs The shaded regions representing the 95% quantile of the MLEs become wider as the masking probability increases. This is an indicator of the decreased precision in the estimates when provided with more ambiguous data about the component cause of failure. However, even for fairly significant Bernoulli masking, $p \leq 0.55$, the 95% quantiles are narrow and the CP is well-calibrated, indicating that the MLEs are still precise and accurate.

IQR of Bootstrapped CIs The IQR (vertical blue bars) show that the bootstrapped BCa CIs are becoming more spread out as the masking probability increases. They are also asymmetric, with the lower bound being more spread out than the upper-bound, but this is consistent with the actual behavior of the dispersion of the MLEs, which exhibits the same pattern. The width of the CIs consistently increase as the masking probability increases, which we intuitively expected given the increased uncertainty about the component cause of failure. After a Bernoulli masking probability of $p \approx 0.5$, the width of the CIs rapidly increase, which is apparently necessary for the CPs to remain well-calibrated.

Bias of MLEs The red dashed line indicating the mean of the MLEs remains stable across different masking probabilities, only showing a significant positive bias when the masking probability p becomes quite significant.

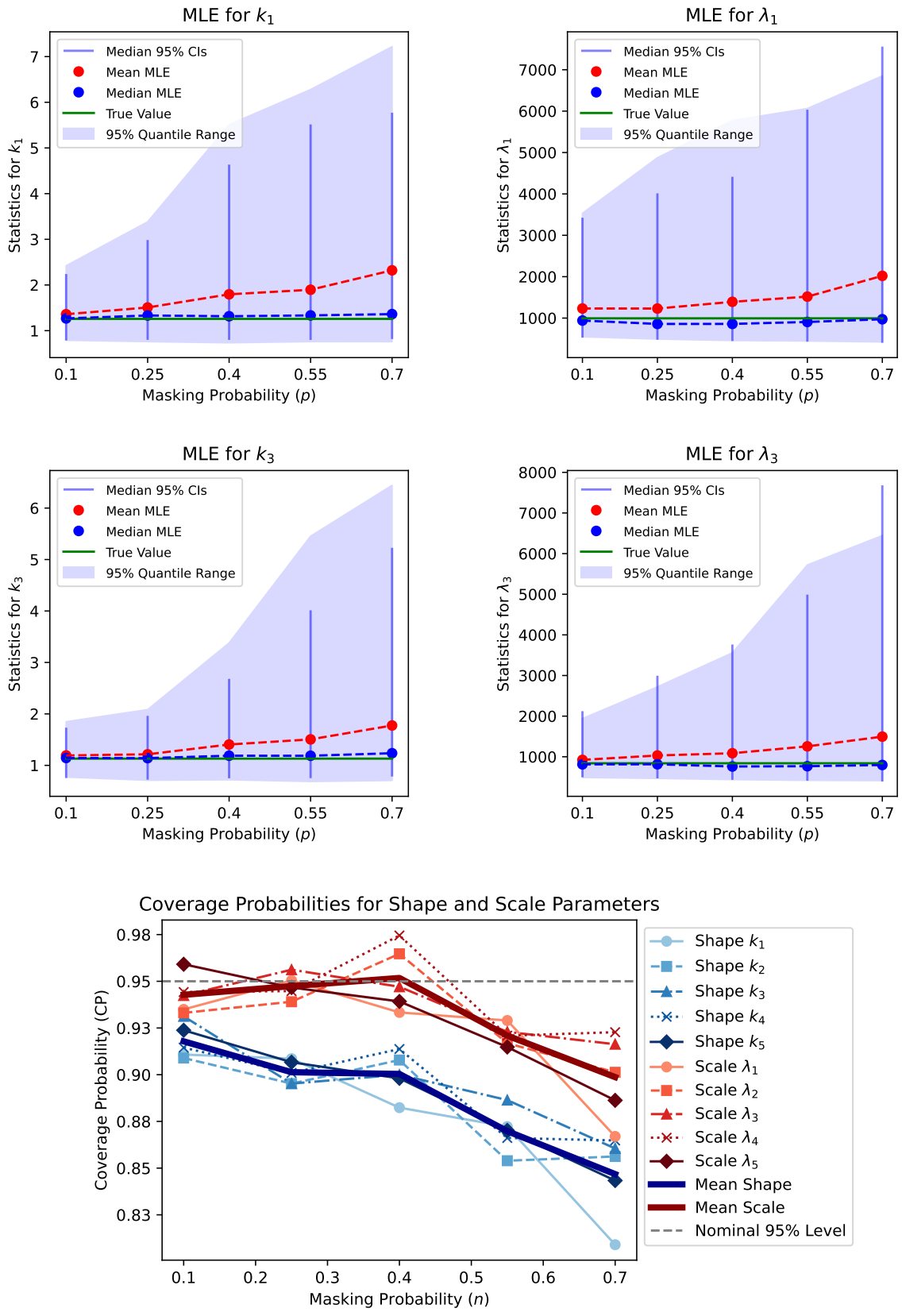

Figure 4: Component Cause of Failure Masking (p) vs MLE

7.5.3 Summary

In this scenario, we examine the influence of masking probability p on the MLEs for shape and scale parameters, keeping the sample size and right-censoring constant. As masking probability increases, the precision of the MLEs decreases, and the coverage probability of the 95% BCa confidence intervals begins to drop. However, even at fairly significant Bernoulli masking probabilities $p \leq 0.55$, the MLEs remain precise and well-calibrated. These observations highlight the challenges of parameter estimation under varying degrees of masking and set the stage for the subsequent scenario on sample size, which shows how increasing the sample size can mitigate the effects of both right-censoring and masking.

7.6 Scenario: Assessing the Impact of Sample Size

In this scenario, we use the well-designed series system described in Table 2. We fix the masking probability to $p = 0.215$ (moderate masking), we fix the right-censoring quantile to $q = 0.825$ (moderate censoring), and we vary the sample size n from 50 (small sample size) to 1000 (very large sample size).

In Figure 5, we show the effect of the sample size n on the MLEs for the shape and scale parameters. The top four plots only show the effect on the MLEs for the shape and scale parameters of components 1 and 4, since the rest were essentially identical, and the bottom two plots show the coverage probabilities for all parameters.

7.6.1 Key Observations

Coverage Probability (CP) The confidence intervals are generally well-calibrated, obtaining coverages above 90% for most of the parameters across the entire sample size range, and they are converging to the nominal 95% level as the sample size increases. This suggests that the bootstrapped CIs will contain the true value of the parameters with the specified confidence level with high probability. The CIs are neither too wide nor too narrow.

However, as in the previous scenario where we varied the right-censoring amount, the scale parameters are better calibrated than the shape parameters. The scale parameters are consistently around the nominal 95% level for all sample sizes, but the shape parameters are consistently less well-calibrated, which reinforces the earlier claim that the shape parameters are more difficult to estimate than the scale parameters.

Dispersion of MLEs The shaded regions representing the 95% probability range of the MLEs get narrower as the sample size increases. This is an indicator of the increased precision in the estimates when provided with more data. This is consistent with the asymptotic properties of the MLE when the regularity conditions are satisfied, e.g., converges in probability to the true value of the parameter as n goes to infinity.

IQR of Bootstrapped CIs The IQR (vertical blue bars) reduces with an increase in sample size. This suggests that the bootstrapped CIs are getting more consistent and focused around a narrower range with larger samples while maintaining a good coverage probability. As we get more data, the bootstrapped CIs are more likely to be closer to each other and the true value of the scale parameter.

For small sample sizes, they are quite large, but to maintain well-calibrated CIs, this was necessary. The estimator is quite sensitive to the data, and so the bootstrapped CIs are quite wide to account for this sensitivity when the sample size is small and not necessarily representative of the true distribution.

Bias of MLEs The red dashed line indicating the mean of MLEs initially has a large positive bias for the shape parameters, but diminshes to neglible levels as the sample size increases. In the previous right-censoring scenario, the bias in the shape parameters never reached zero, but we see that in this scenario, at around a sample size of 250, the bias approaches zero, suggesting that there is enough information in the sample to overcome the bias from the right-censoring ($q = 0.825$) and masking ($p = 0.215$) effects.

The bias for the scale parameters is quite small and remains stable across different sample sizes, suggesting that the scale MLEs are reasonably unbiased, which was also the case in the previous scenario.

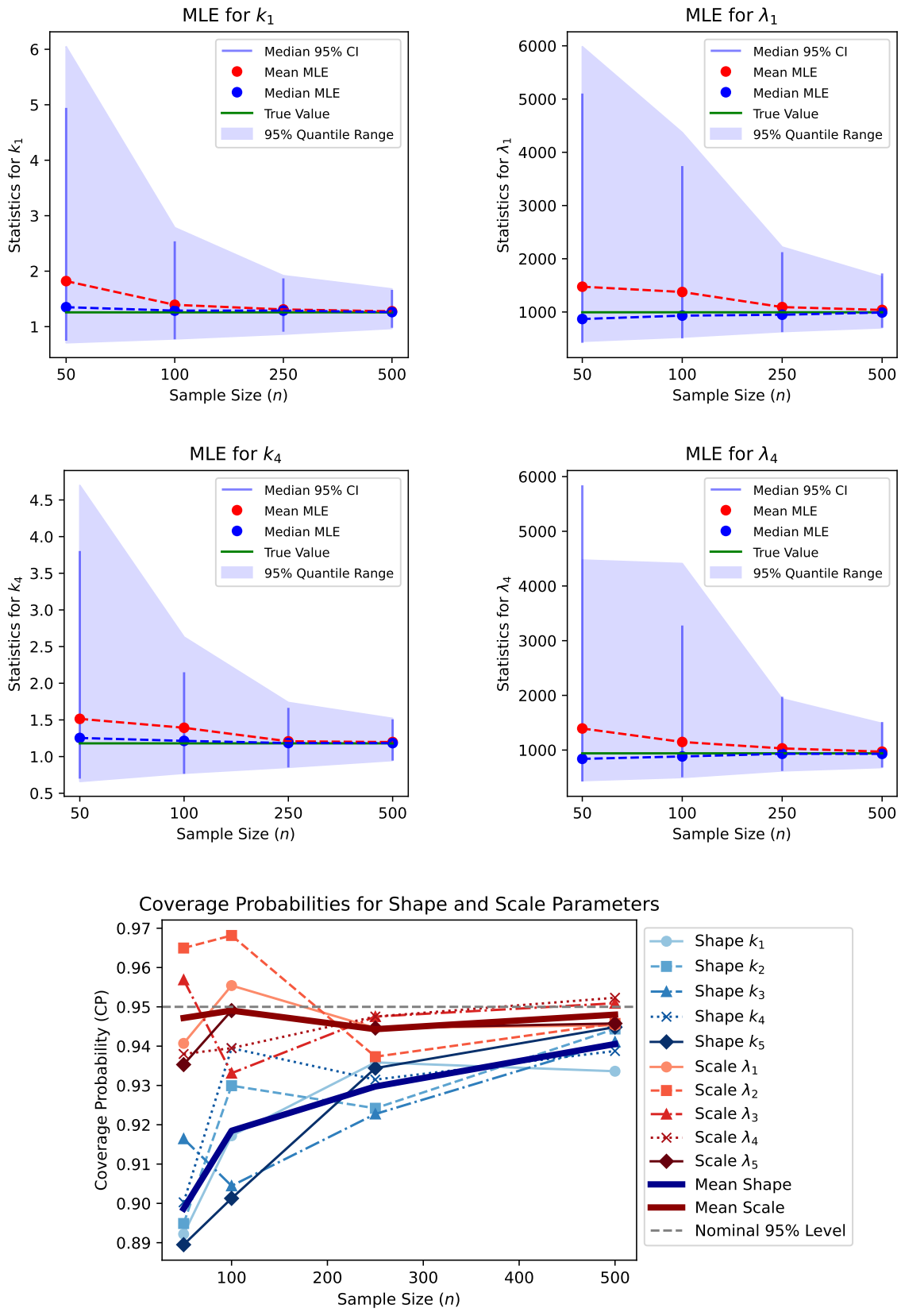

Figure 5: Sample Size vs MLEs ($p = 0.215, q = 0.825$)

7.6.2 Summary

In this scenario, we delve into the mitigating effects of sample size on the challenges identified in previous scenarios concerning right-censoring and masking. As the sample size increases, the precision of the MLEs for both shape and scale parameters notably improves, and the bias approaches zero. This is consistent with statistical theory and suggests that increasing the sample size can mitigate the effects of right-censoring and masking. The Bootstrapped (BCa) confidence intervals also narrow and become more reliable with larger samples, reinforcing the role of sample size in achieving robust estimates.

8 Future Work

This paper developed maximum likelihood techniques and simulation studies to estimate component reliability from masked failure data in series systems. The key results were:

- The likelihood model enabled rigorous inference from masked data via right-censoring and candidate sets.
- Despite masking and censoring, the MLE demonstrated accurate and robust performance in simulation studies.
- Bootstrap confidence intervals were reasonably well-calibrated, even for small samples.
- Estimation of shape parameters was more challenging than scale parameters.

Building on these findings, promising areas for future work include:

- Relaxing assumptions on masking conditions to further generalize the likelihood model.
- Assessing sensitivity to deviations from well-designed systems.
- Exploring regularization methods like data augmentation and penalized likelihood to improve shape parameter estimates.
- Incorporating predictors into a more general likelihood model.
- Evaluating the bootstrap for related statistics like prediction intervals.

The current results provide a solid foundation for extensions like these that can further refine the methods and expand their applicability. By leveraging the rigorous likelihood framework and simulation techniques validated in this study, future work can continue advancing the capability for statistical learning from masked reliability data.

9 Conclusion

This work presented maximum likelihood techniques to estimate component reliability from masked failure data in series systems. The methods demonstrated accurate and robust performance despite significant challenges introduced by masking and right-censoring.

Simulation studies revealed particular sensitivities in estimating shape parameters of component lifetimes that follow Weibull distributions.

Right-censoring and masking positively biased these shape estimates. However, with sufficiently large sample sizes around ($n > 200$), these biases were overcome, suggesting enough information existed in the data to mitigate censoring and masking effects. In contrast, scale parameters exhibited more robust properties even with smaller samples.

Despite the challenges, bootstrapping yielded reasonably well-calibrated confidence intervals even for small sample sizes. The modeling framework provides a statistically rigorous basis for quantifying latent component properties from limited observational data on system reliability. The simulation studies validate the methods and provide practical insights into their performance under varying real-world conditions. This advances the capability for statistical learning from masked system failure data.

A Series System with Weibull Component Lifetimes

These functions are implemented in the R library `wei.series.md.c1.c2.c3` and are available on GitHub. They are for the series system with Weibull component lifetimes with data described in Section 3. For clarity and brevity, we removed some of the functionality and safeguards in the actual code, but we provide the full code in the R package.

A.1 Log-likelihood Function

The log-likelihood function is the sum of the log-likelihood contributions for each system. For our series system with Weibull component lifetimes, we analytically derived the log-likelihood function in Theorem 6.2 and implemented it in the `loglik_wei_series_md_c1_c2_c3` R function below.

```r
#' Generates a log-likelihood function for a series system with Weibull
#' component lifetimes for masked data.
#'
#' @param df masked data frame
#' @param theta parameter vector
#' @returns log-likelihood function
loglik_wei_series_md_c1_c2_c3 <- function(df, theta) {
  n <- nrow(df)
  C <- md_decode_matrix(df, "x")
  m <- ncol(C)
  delta <- df[["delta"]]
  shapes <- theta[seq(1, length(theta), 2)]
  scales <- theta[seq(2, length(theta), 2)]
  t <- df[[lifetime]]
  s <- 0
  for (i in 1:n) {
    s <- s - sum((t[i] / scales)^shapes)
    if (delta[i]) {
      s <- s + log(sum(shapes[C[i, ]] / scales[C[i, ]] *
        (t[i] / scales[C[i, ]])^(shapes[C[i, ]] - 1)))
    }
  }
  return(s)
}
```

A.2 Score Function

The score function is the gradient of the log-likelihood function. For our series system with Weibull component lifetimes, we analytically derived the score function in Theorem 6.3 and implemented it in the `score_wei_series_md_c1_c2_c3` R function below.

```r
#' Computes the score function for a series system with Weibull component
#' lifetimes for masked data.
#'
#' @param df masked data frame
#' @param theta parameter vector
#' @returns score function
score_wei_series_md_c1_c2_c3 <- function(df, theta) {
  n <- nrow(df)
  C <- md_decode_matrix(df, "x")
  m <- ncol(C)
```

```
  delta <- df[["delta"]]
  t <- df[[lifetime]]
  shapes <- theta[seq(1, length(theta), 2)]
  scales <- theta[seq(2, length(theta), 2)]
  shapes.scr <- scales.scr <- rep(0, m)

  for (i in 1:n) {
    shapes.rt <- -(t[i] / scales)^shapes * log(t[i] / scales)
    scales.rt <- (shapes / scales) * (t[i] / scales)^shapes
    shapes.trm <- scales.trm <- rep(0, m)

    if (delta[i]) {
      c <- C[i, ]
      denom <- sum(shapes[c] / scales[c] *
        (t[i] / scales[c])^(shapes[c] - 1))

      shapes.num <- (t[i] / scales[c])^shapes[c] / t[i] *
        (1 + shapes[c] * log(t[i] / scales[c]))
      shapes.trm[c] <- shapes.num / denom

      scales.num <- (shapes[c] / scales[c])^2 *
        (t[i] / scales[c])^(shapes[c] - 1)
      scales.trm[c] <- scales.num / denom
    }

    shapes.scr <- shapes.scr + shapes.rt + shapes.trm
    scales.scr <- scales.scr + scales.rt - scales.trm
  }

  scr <- rep(0, length(theta))
  scr[seq(1, length(theta), 2)] <- shapes.scrs
  scr[seq(2, length(theta), 2)] <- scales.scrs
  return(scr)
}
```

A.3 Quantile Function

For our series system with Weibull component lifetimes, the quantile function is the inverse of the cdf F_{T_i}. By definition, the quantile p for the strictly monotonically increasing cdf F_{T_i} is the value t that satisfies $F_{T_i}(t;\theta) - p = 0$, and so we solve for t using Newton's method, in which the k^{th} iteration is given by

$$t^{(k+1)} = t^{(k)} - \frac{F_{T_i}(t^{(k)};\theta) - p}{f_{T_i}(t^{(k)};\theta)}.$$

We have derived a slightly more efficient method in the `qwei_series` R function below.

```
#' Quantile function for a series system with Weibull component lifetimes.
#'
#' @param p quantile
#' @param shapes shape parameters
#' @param scales scale parameters
#' @returns p-th quantile
qwei_series <- function(p, shapes, scales) {
```

```
  t0 <- 1
  repeat {
    t1 <- t0 - sum((t0 / scales)^shapes) + log(1 - p) /
      sum(shapes * t0^(shapes - 1) / scales^shapes)
    if (abs(t1 - t0) < tol) {
      break
    }
    t0 <- t1
  }
  return(t1)
}
```

A.4 Maximum Likelihood Estimation

We use the Newton-Raphson method for Maximum Likelihood Estimation (MLE) in a series system with Weibull component lifetimes. Numerical optimization is carried out using R's `optim` package and the **L-BFGS-B** method [4]. This quasi-Newton method approximates the Hessian using the gradient of the log-likelihood function (see Appendices A.1 and A.2). Bound constraints are applied to maintain positive shape and scale parameters.

```
#' L-BFGS-B solver for the series system with Weibull component lifetimes
#' given masked data.
#'
#' @param df masked data frame
#' @param theta0 initial guess
#' @return MLE solution
mle_lbfgsb_wei_series_md_c1_c2_c3 <- function(df, theta0) {
  optim(theta0,
    fn = function(theta) loglik_wei_series_md_c1_c2_c3(df,
      theta[seq(1, length(theta), 2)], theta[seq(2, length(theta), 2)]),
    gr = function(theta) score_wei_series_md_c1_c2_c3(df,
      theta[seq(1, length(theta), 2)], theta[seq(2, length(theta), 2)]),
    lower = rep(1e-9, length(theta0)),
    method = "L-BFGS-B",
    control = list(fnscale = -1))
}
```

B Simulation

Here is where simulation-specific Python and R code resides, such as scenario simulation.

B.1 Scenario Simulation

The following R code is the Monte-carlo simulation code for running the various scenarios described in Section 7.

```
#### Setup simulation parameters here ####
theta <- c(
  shape1 = 1.2576, scale1 = 994.3661,
  shape2 = 1.1635, scale2 = 908.9458,
  shape3 = 1.1308, scale3 = 840.1141,
  shape4 = 1.1802, scale4 = 940.1342,
  shape5 = 1.2034, scale5 = 923.1631
```

```
)

N <- c(30, 60, 100) # sample sizes to simulate
P <- c(.215) # masking probabilities to simulate
Q <- c(.825) # right censoring probabilities to simulate
R <- 1000L # number of simulations per scenario
B <- 1000L # number of bootstrap samples
max_iter <- 125L # max iterations for MLE
max_boot_iter <- 125L # max iterations for bootstrap MLE
n_cores <- detectCores() - 1 # number of cores to use for parallel processing
filename <- "data" # filename prefix for output files

#### Simulation code below here ####
file.meta <- paste0(filename, ".txt")
file.csv <- paste0(filename, ".csv")
shape <- theta[seq(1, length(theta), 2)]
scale <- theta[seq(2, length(theta), 2)]
m <- length(shape)

for (n in N) {
  for (p in P) {
    for (q in Q) {
      tau <- qwei_series(p = q, scales = scales, shapes = shapes)
      iter <- 0L
      while (iter < R) {
        retry <- FALSE
        tryCatch({
          repeat {
            df <- generate_guo_weibull_table_2_data(
              shapes = shapes, scales = scales, n = n, p = p, tau = tau)

            sol <- mle_lbfgsb_wei_series_md_c1_c2_c3(
              theta0 = theta, df = df, hessian = FALSE,
              control = list(maxit = max_iter, parscale = theta)
            )
            if (sol$convergence == 0) {
              break
            }
          }
          mle_solver <- function(df, i) {
            mle_lbfgsb_wei_series_md_c1_c2_c3(
              theta0 = sol$par, df = df[i, ], hessian = FALSE,
              control = list(maxit = max_boot_iter, parscale = sol$par)
            )$par
          }
          sol.boot <- boot(df, mle_sol, R = B, parallel = "multicore",
            ncpus = n_cores)
          ci <- confint(mle_boot(sol.boot), type = ci_method,
              level = ci_level)
          shape.ci <- ci[seq(1, length(theta), 2), ]
          scale.ci <- ci[seq(2, length(theta), 2), ]
        }, error = function(e) { retry <<- TRUE })
```

```
        shape.mle <- sol$par[seq(1, length(theta), 2)]
        scale.mle <- sol$par[seq(2, length(theta), 2)]
        row <- data.frame(
          n = n, p = p, q = q, tau = tau, B = B,
          shape = shape, scale = scale,
          shape.mle = shape.mle, scale.mle = scale.mle,
          shape.lower = shape.ci[, 1], shape.upper = shape.ci[, 2],
          scale.lower = scale.ci[, 1], scale.upper = scale.ci[, 2])
        write.table(row,
          file = file.csv, sep = ",", row.names = FALSE,
          col.names = !file.exists(file.csv), append = TRUE)
        iter <- iter + 1L
      }
    }
  }
}
```

B.2 Bernoulli Candidate Set Model

```
#' Bernoulli candidate sets for masked data.
#'
#' @param df masked data frame.
#' @param p prob vector (p[i] is Bernoulli prob for i-th system
#' @returns masked data frame with Bernoulli candidate set probs
md_bernoulli_cand_c1_c2_c3 <- function(df, p) {
  n <- nrow(df)
  p <- rep(p, length.out = n)
  Tm <- md_decode_matrix(df, "t")
  m <- ncol(Tm)
  Q <- matrix(p, nrow = n, ncol = m)
  Q[cbind(1:n, apply(Tm, 1, which.min))] <- 1
  Q[!df[["delta"]], ] <- 0
  df %>% bind_cols(md_encode_matrix(Q, prob))
}
```

B.3 Plot Generation for Sampling Distribution of MLE

The following Python code generates the plots relating the the sampling distribution of the MLE for the simulation studies.

```
################ Setup ################
x_col = 'p'
params = ['scale.1', 'shape.1', 'scale.4', 'shape.4']
param_labels = ['\lambda_1', 'k_1', '\lambda_4', 'k_4']

data = pd.read_csv(csv_file)
for param, param_label in zip(params, param_labels):
  plot_x_vs_mle(data, x_col, param)

################ Plotting Functions ################
def plot_x_vs_mle(raw_data, x_col, par):
  ps = par.split('.')
```

```
par_low = f'{ps[0]}.lower.{ps[1]}'
par_up = f'{ps[0]}.upper.{ps[1]}'
par_mle = f'{ps[0]}.mle.{ps[1]}'
x_vals = sorted(raw_data[x_col].unique())

median_mles = []
mean_mles = []
low_q = []
up_q = []
plt.figure(figsize=(4, 4))

for i, x in enumerate(x_vals):
  data = raw_data[raw_data[x_col] == x]
  low, up = np.percentile(data[par_low], 50), np.percentile(data[par_up], 50)
  mean_mle = data[par_mle].mean()
  median_mle = data[par_mle].median()
  mean_mles.append(mean_mle)
  median_mles.append(median_mle)
  low_q.append(np.percentile(data[par_mle], 2.5))
  up_q.append(np.percentile(data[par_mle], 97.5))

  plt.vlines(i, low, up, color='blue', label='Median 95% CI' if i == 0 else "")
  plt.plot(i, mean_mle, 'ro', label='Mean MLE' if i == 0 else "")
  plt.plot(i, median_mle, 'bo', label='Median MLE' if i == 0 else "")
  plt.plot(i, data[par].mean(), 'g-', label='True Value' if i == 0 else "")

plt.plot(np.arange(len(x_vals)), mean_mles, 'r--')
plt.plot(np.arange(len(x_vals)), median_mles, 'b--')

plt.fill_between(np.arange(len(x_vals)), low_q, up_q,
  color='blue', alpha=0.15, label='95% Quantile Range')

plt.xticks(np.arange(len(x_vals)), x_vals)
plt.xlabel(x_col)
plt.ylabel(par)
plt.title(f'MLE for ${par}$')
plt.legend(loc='upper left')

# decrease text side of legend
leg = plt.gca().get_legend()
for text in leg.get_texts():
  plt.setp(text, fontsize='small')

plt.tight_layout(h_pad=4.0, w_pad=2.5)
plt.savefig(f'plot-{x_col}-vs-{par}.pdf')
plt.close()
```

B.4 Plot Generation for Coverage Probabilities

The following Python code generates the plots relating the the coverage probabilities of the MLE for the simulation studies.

```
###### Setup ######
csv_file = "data.csv"
x_col = 'n'

###### Plotting Functions ######
data = pd.read_csv(csv_file)
cols = (
    [x_col] + [f'shape.{i}' for i in range(1, 6)] +
    [f'scale.{i}' for i in range(1, 6)] +
    [f'shape.lower.{i}' for i in range(1, 6)] +
    [f'shape.upper.{i}' for i in range(1, 6)] +
    [f'scale.lower.{i}' for i in range(1, 6)] +
    [f'scale.upper.{i}' for i in range(1, 6)]
)
col_data = data[cols].copy()

def compute_coverage(row, j):
    shape_within = (
        row[f'shape.lower.{j}'] <= row[f'shape.{j}'] <= row[f'shape.upper.{j}']
    )
    scale_within = (
        row[f'scale.lower.{j}'] <= row[f'scale.{j}'] <= row[f'scale.upper.{j}']
    )
    return pd.Series([shape_within, scale_within], index=[
        f'shape_coverage.{j}', f'scale_coverage.{j}']
    )

for j in range(1, 6):
    col_data[[f'shape_coverage.{j}', f'scale_coverage.{j}']] = (
        col_data.apply(lambda row: compute_coverage(row, j), axis=1)
    )

coverage = [f'shape_coverage.{j}' for j in range(1, 6)] + \
    [f'scale_coverage.{j}' for j in range(1, 6)]
cp = col_data.groupby(x_col)[coverage].mean().reset_index()

mean_shape_cp = cp[[f'shape_coverage.{j}' for j in range(1, 6)]].mean(axis=1)
mean_scale_cp = cp[[f'scale_coverage.{j}' for j in range(1, 6)]].mean(axis=1)
cp['mean_shape'] = mean_shape_cp
cp['mean_scale'] = mean_scale_cp

plt.figure(figsize=[7, 4])
shape_cmap = plt.get_cmap('Blues')
scale_cmap = plt.get_cmap('Reds')
line_styles = ['-', '--', '-.', ':', '-']
markers = ['o', 's', '^', 'x', 'D']

for j, color, ls, mk in zip(range(1, 6), shape_cmap(np.linspace(0.4, 1, 5)),
                            line_styles, markers):
    plt.plot(cp[x_col], cp[f'shape_coverage.{j}'],
             label=f'Shape $k_{j}$', color=color, linestyle=ls, marker=mk)

for j, color, ls, mk in zip(range(1, 6), scale_cmap(np.linspace(0.4, 1, 5)),
```

```
                            line_styles, markers):
    plt.plot(cp[x_col], cp[f'scale_coverage.{j}'],
             label=f'Scale $\lambda_{j}$', color=color, linestyle=ls, marker=mk)

plt.plot(cp[x_col], cp['mean_shape'], color='darkblue',
         linewidth=4, linestyle='-', label='Mean Shape')
plt.plot(cp[x_col], cp['mean_scale'], color='darkred',
         linewidth=4, linestyle='-', label='Mean Scale')
plt.axhline(y=0.95, color='grey', linestyle='--', label='Nominal 95% Level')
plt.xlabel(x_col)
plt.ylabel('Coverage Probability (CP)')
plt.gca().yaxis.set_major_formatter(FuncFormatter(lambda y, _: '{:.2f}'.format(y)))
plt.title('Coverage Probabilities for Shape and Scale Parameters')
plt.legend(loc='best', bbox_to_anchor=(1, 1))
plt.tight_layout(rect=[0, 0, 0.85, 1])
plt.savefig(f'{x_col}-vs-cp.pdf')
```


Acknowledgements

I would like to thank my advisor, Dr. Marcus Agustin, for his invaluable guidance and support throughout this research project and my graduate studies. His expertise and encouragement as I faced health challenges were invaluable. I am also grateful to my committee member, Dr. Beidi Qiang, for reviewing my work and providing helpful feedback. She is an exceptional teacher and I fondly remember her classes. Finally, I would also like to thank my committee member, _____, for reviewing my work and providing helpful feedback.

References

[1] Robert B. Abernethy, *New weibull handbook*, 5th ed., Abernethy, 2006.

[2] Marcus Agustin, *Systems in series*, John Wiley & Sons, Ltd, 2011.

[3] L.J. Bain and M. Engelhardt, *Introduction to probability and mathematical statistics*, second ed., Duxbury Press, 1992.

[4] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu, *A limited memory algorithm for bound constrained optimization*, SIAM Journal on Scientific Computing **16** (1995), no. 5, 1190–1208.

[5] George Casella and Roger L Berger, *Statistical inference*, Duxbury Advanced Series, 2002.

[6] B. Efron, *Better bootstrap confidence intervals*, Journal of the American Statistical Association **82** (1987), no. 397, 171–185.

[7] Bradley Efron and Robert J Tibshirani, *An introduction to the bootstrap*, CRC press, 1994.

[8] Frank M. Guess, Thom J. Hodgson, and John S. Usher, *Estimating system and component reliabilities under partial information on cause of failure*, Journal of Statistical Planning and Inference **29** (1991), 75–85.

[9] Huairui Guo, Pengying Niu, and F. Szidarovszky, *Estimating component reliabilities from incomplete system failure data*, Annual Reliability and Maintainability Symposium (RAMS) (2013), 1–6.

[10] John P Klein and Melvin L Moeschberger, *Survival analysis: techniques for censored and truncated data*, Springer Science & Business Media, 2005.

[11] Erich L. Lehmann and George Casella, *Theory of point estimation*, Springer Science & Business Media, 1998.

[12] Nassim Nicholas Taleb, *The black swan: The impact of the highly improbable*, Random House, 2007.

[13] C. F. Jeff Wu, *On the convergence properties of the em algorithm*, The Annals of Statistics **11** (1983), no. 1, 95–103.