Bootstrapping statistics of the maximum likelihood estimator of components in a series systems from masked failure data

Alex Towell

Abstract

We estimate the parameters of a series system with Weibull component lifetimes from relatively small samples consisting of right-censored system lifetimes and masked component cause of failure. Under a set of conditions that permit us to ignore how the component cause of failures are masked, we assess the bias and variance of the estimator. Then, we assess the accuracy of the boostrapped variance and calibration of the confidence intervals of the MLE under a variety of scenarios.

Contents

1 Introduction

Accurate reliability specifications of components of a multi-component system, such as mean time to failure (MTTF), can be quite valuable. For instance, such information may be used to help identify the component cause of system failure. However, this information is often not available, or if it is available, it may not be very accurate. In this case, we may try to estimate the reliability of each component from system failure data.

We are interested in estimating the reliability of *series systems*. In a series system, the system fails whenever any of its components fail. The famous statistician George Box once opined that all models are wrong, but some are useful, meaning that a theoretical model of real phenomenon is incapable of representing its exact behaviour, but it may still be useful. Indeed, sometimes it is even more useful, since a simplified model is more understandable and only includes the most salient features.

Since many real-world systems are greatly impaired whenever a set of its *critical* components fail, the series system model is often a very *useful* approximation of its internal structure. For example, people in a particular experiment may be regarded as series systems where the components are some relevant set of vital organs. This model is often a very useful abstraction that averages over many complicated, unobserved details that may only have a small effect on an average person's lifespan.

In this paper, the experimental unit is a particular type of series system that consists of m components and we seek to estimate the reliability of the components from *masked data* where we observe only partial information about the system, e.g., the component lifetimes are masked (not observed in the sample).

We imagine that we conduct an experiment where we obtain the lifetimes of the m components, which has a sample space $(0, \infty)^m$. An observed result is called an *outcome* of the experiment, e.g., $(1.2, 0.5, 1.1)$. However, we assume that this data is masked in the following ways:

1. The experiments may be right-censored, which means that the experiment or observation may be suspended before the system fails.

2. We can only observe the lifetime of the component with the minimum lifetime. This constraint is imposed by the fact that the system is arranged in series and thus stops working whenever one of the components fails.

3. It may not be easy or possible to identify the failed component. We consider the case where we can only observe a set of components that includes the failed component. We refer to this set of components as the candidate set.[1]

   This constraint may be due to a variety of reasons. A canonical example is given by supposing that when the series system fails, it is repaired by simultaneously replacing some subset of the components, thus preventing isolating the specific failed component.

We take a random sample of n such experiments, where the right-censored system lifetimes are i.i.d. when conditioned on the right-censoring time, but the candidate sets are assumed to be independent but not necessarily identically distributed. In later sections, we consider a set of realistic conditions that allow us to ignore the distribution of candidate sets, which allows us to construct a reduced likelihood function for the masked failure data. Finally, we find the MLE $\hat{\boldsymbol{\theta}}$ that maximizes the reduced likelihood function, which is the same MLE for "full" likelihood function where we know the distribution of candidate sets.

2 Series System Model

Consider a system with m components, indexed by $1, 2, \ldots, m$. Each component only has two possible states, *failed* and *functioning*. The state of the system also only has two possible states, *failed* and *functioning*. The lifetime of a system is the elapsed time from when the new, functioning system is put into operation until it fails for the *first time*. We are not interested in what happens to a system after it fails, e.g., we do not consider repairing systems and putting them back into operation under further observation.

[1] If the candidate set has only a single component, then we know the exact component cause of failure.

A system that is in a functioning state if and only if at least k of the m components are in a functioning state is denoted a k-out-of-m system. We narrow our scope to m-out-of-m systems, also known as *series systems* (with m components). A series system is functioning if and only if every component is functioning. Consequently, in a series system, precisely one component is the cause of failure.

Our sample consists of lifetime data for i.i.d. series systems. Since the component lifetimes are subject to chance variations, we denote the lifetime of the j^{th} component of the i^{th} system by the random variable T_{ij}. We assume that the m component lifetimes, $T_{i1}, \ldots, T_{im}$, are statistically independent and non-identical. In a series system, whenever any component fails, the system fails. Thus, the lifetime of the i^{th} system, T_i, is given by the component with the smallest lifetime,

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

We assume that the component lifetimes may be adequately modeled by some parametric probability distribution. In what follows, matrices and vectors are denoted in boldface, e.g., $\boldsymbol{x}$ is a vector. The i^{th} column and (i,j)-th element of a matrix $\boldsymbol{A}$ is denoted respectively by $\boldsymbol{A_j}$ and $\boldsymbol{A}_{ij}$. We let $\boldsymbol{\theta_j}$ denote the parameter vector of the j^{th} component, and each $\boldsymbol{\theta_j}$ for $j = 1, \ldots, m$ may be different sizes. We define the parameter vector of the entire series system by

$$\boldsymbol{\theta} = (\boldsymbol{\theta_1}, \ldots, \boldsymbol{\theta_m}).$$

The cumulative distribution function (cdf) for a random varible T is given by

$$F_T(t) = \Pr\{T \leq t\}, \tag{2.1}$$

where $\Pr\{T \in E\}$ denotes the probability that a random variable T realizes a value in E. If a random variable X is discrete, then its probability mass function (pmf) f_X is given by$f_X(x) = \Pr\{X = x\}$. The probability density function (pdf) for a continuous random varible T is given by

$$f_T(t) = \frac{d}{dt} F_T(t). \tag{2.2}$$

If a random variable T has a parametric distribution indexed by a parameter vector $\boldsymbol{\beta}$, we denote its pdf by $f_T(t; \boldsymbol{\beta})$ and likewise for other distribution functions, e.g., the i^{th} series system has a cdf denoted by $F_{T_i}(t; \boldsymbol{\theta})$. In the case of the lifetime distribution functions of components, we subscript the distribution functions by component index instead of the symbol for the random variable, e.g., the cdf of the j^{th} component for the i-th system is denoted by $F_j(t; \boldsymbol{\theta_j})$ instead of $F_{T_{ij}}(t; \boldsymbol{\theta_j})$. If it is clear from the context which random variable a distribution function is for, we may drop the subscripts, e.g., $F(t)$ instead of $F_T(t)$. Finally, as an abuse of notation, we often write a function as $f(t)$ when we really mean that f is a function of variable t.

There are two particularly important functions in survival analysis, the survival function and the hazard function.

Definition 1. *The survival function, $R_T(t)$, of a random lifetime T is the probability that it realizes a value larger than some specified duration of time t,*

$$\begin{aligned} R_T(t) &= \Pr\{T > t\} \\ &= 1 - F_T(t). \end{aligned} \tag{2.3}$$

In other words, $R_T(t)$ denotes the probability that T survives longer than t.

The hazard function is a bit more subtle. For a random lifetime T, the probability that a failure occurs between t and Δt given that no failure occurs before time t is given by

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

The *failure rate* is given by the above divided by the length of the time interval, Δt:

$$\frac{\Pr\{t < T < t + \Delta t\}}{\Delta t} \frac{1}{\Pr\{T > t\}} = \frac{R_T(t) - R(t + \Delta t)}{R_T(t)}.$$

Definition 2. *The hazard function $h_T(t)$ for a continuous random variable T is the instantaneous failure rate at time t, which is given by*

$$\begin{aligned} h_T(t) &= \lim_{\Delta t \to 0} \frac{\Pr\{t < T < t + \Delta t\}}{\Delta t} \frac{1}{\Pr\{T > t\}} \\ &= \frac{f_T(t)}{R_T(t)}. \end{aligned} \tag{2.4}$$

Two 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 $X_1, \ldots, X_m$ is denoted by $f(x_1, \ldots, x_m)$.

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 1. *The series system has a reliability function given by*

$$R(t; \boldsymbol{\theta}) = \prod_{j=1}^{m} R_j(t; \boldsymbol{\theta_j}). \tag{2.5}$$

Proof. The reliability function is defined as

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

which may be rewritten as

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

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

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

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

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

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

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

□

Theorem 1 shows that the system's overall reliability is the product of the reliabilities of its individual components. This property is inherent to 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. *The series system has a pdf given by*

$$f(t; \boldsymbol{\theta}) = \sum_{j=1}^{m} f_j(t; \boldsymbol{\theta_j}) \prod_{\substack{k=1 \\ k \neq j}}^{m} R_k(t; \boldsymbol{\theta_j}). \tag{2.6}$$

Proof. By definition, the pdf may be written as

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

By the product rule, this may be rewritten as

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

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

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

which simplifies to

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

□

Theorem 2 shows the pdf of the system lifetime as 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 3. *The series system has a hazard function given by*

$$h(t;\boldsymbol{\theta}) = \sum_{j=1}^{m} h_j(t;\boldsymbol{\theta_j}). \tag{2.7}$$

Proof. The i^{th} series system lifetime has a hazard function defined as

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

Plugging in expressions for these functions results in

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

which can be simplified to

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

□

Theorem 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,

$$h(t;\boldsymbol{\theta}) = \frac{f(t;\boldsymbol{\theta})}{R(t;\boldsymbol{\theta})},$$

and we can rearrange this to get

$$\begin{aligned} f(t;\boldsymbol{\theta}) &= h(t;\boldsymbol{\theta})R(t;\boldsymbol{\theta}) \\ &= \Big\{\sum_{j=1}^{m} h_j(t;\boldsymbol{\theta_j})\Big\}\Big\{\prod_{j=1}^{m} R_j(t;\boldsymbol{\theta_j})\Big\}, \end{aligned} \tag{2.8}$$

which we sometimes find to be a more convenient form than Equation (2.6).

In this section, we derived the mathematical forms for the system's reliability function, pdf, and hazard function. 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 model the component cause of the series system failure as a random variable.

Definition 3. *The component cause of failure of a series system is denoted 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.

Note that a more succinct way to define K_i is given by

$$K_i = \text{argmin}_j\big\{T_{ij} : j \in \{1,\ldots,m\}\big\}.$$

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

Theorem 4. *The joint pdf of the component cause of failure K_i and series system lifetime T_i is given by*

$$f_{K_i,T_i}(j,t;\boldsymbol{\theta}) = h_j(t;\boldsymbol{\theta_j})R_{T_i}(t;\boldsymbol{\theta}), \tag{2.9}$$

where $h_j(t;\boldsymbol{\theta_j})$ is the hazard function of the j^{th} component and $R_{T_i}(t;\boldsymbol{\theta})$ is the reliability function of the series system.

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

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

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,

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

Since $h_1(t;\boldsymbol{\theta_1}) = f_1(t;\boldsymbol{\theta_1})/R_1(t;\boldsymbol{\theta_1})$,

$$f_1(t;\boldsymbol{\theta_1}) = h_1(t;\boldsymbol{\theta_1})R_1(t;\boldsymbol{\theta_1}).$$

Making this substitution into the above expression for $f_{K_i,T_i}(j,t;\boldsymbol{\theta})$ yields

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

Generalizing from this completes the proof. □

3 Likelihood Model for Masked Data

The object of interest is the (unknown) parameter value $\boldsymbol{\theta}$. To estimate this $\boldsymbol{\theta}$, we need *data*. In our case, we call it *masked data* because we do not necessarily observe the event of interest, say a system failure, directly. We consider two types of masking: masking the system failure lifetime and masking the component cause of failure.

We generally encounter three types of system failure lifetime masking:

1. A system failure is observed at a particular point in time.
2. A system failure is observed to occur within a particular interval of time.
3. A system failure is not observed, but we know that the system survived at least until a particular point in time. This is known as *right-censoring* and can occur if, for instance, an experiment is terminated while the system is still functioning.

We generally encounter two types of component cause of failure masking:

1. The component cause of failure is observed.
2. The component cause of failure is not observed, but we know that the failed component is in some set of components. This is known as *masking* the component cause of failure.

Thus, the component cause of failure masking will take the form of candidate sets. A candidate set consists of some subset of component labels that plausibly contains the label of the failed component. The sample space of candidate sets are all subsets of $\{1,\ldots,m\}$, thus there are 2^m possible outcomes in the sample space.

In this paper, we limit our focus to observing *right censored* lifetimes and exact lifetimes but with masked component cause of failures. We consider a sample of n i.i.d. series systems, each of which is put into operation at some time and and observed until either it fails or is right-censored. We denote the right-censoring time of the i^{th} system by τ_i. We do not directly observe the system lifetime, T_i, but rather, we observe the right-censored lifetime, S_i, which is given by

$$S_i = \min\{\tau_i, T_i\}, \tag{3.1}$$

We also observe a right-censoring indicator, δ_i, which is given by

$$\delta_i = 1_{T_i<\tau_i} \tag{3.2}$$

where $1_{\text{condition}}$ is an indicator function that outputs 1 if *condition* is true and 0 otherwise. Here, $\delta_i = 1$ indicates the event of interest, a system failure, was observed.

If a system failure lifetime is observed, 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 $\{1,\ldots,m\}$. Since the data generating process for candidate sets may be subject to chance variations, it as a random set.

Consider we have an independent and identically distributed (i.i.d.) random sample of masked data, $D = \{D_1,\ldots,D_n\}$, where each D_i contanis the following:

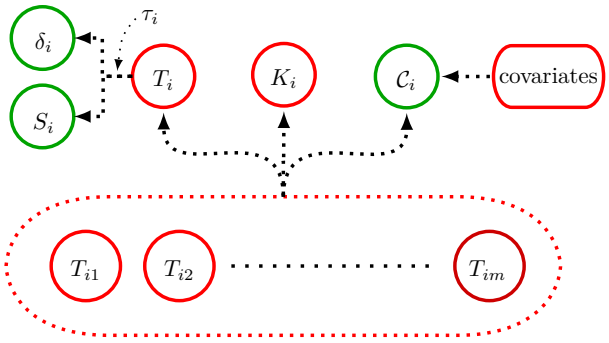

Figure 1: This figure showcases a dependency graph of the generative model for $D_i = (S_i, \delta_i, \mathcal{C}_i)$. The elements in green are observed in the sample, while the elements in red are unobserved (latent). We see that $\mathcal{C}_i$ is related to both the unobserved component lifetimes $T_{i1}, \ldots, T_{im}$ 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 seek a way to construct a reduced likelihood function in later sections that is not a function of the distribution of $\mathcal{C}_i$.

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

The masked data generation process is illustrated by Figure 1.

An example of masked data D for exact, right-censored system failure times with candidate sets that mask the component cause of failure can be seen in Table 1 for a series system with $m = 3$ components.

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

System	Right-censoring time (S_i)	Right censoring indicator (δ_i)	Candidate set ($\mathcal{C}_i$)
1	4.3	1	$\{1, 2\}$
2	1.3	1	$\{2\}$
3	5.4	0	$\emptyset$
4	2.6	1	$\{2, 3\}$
5	3.7	1	$\{1, 2, 3\}$
6	10	0	$\emptyset$

In statistical modeling, the likelihood is a powerful tool that allows us to estimate the parameters of a model given observed data. In more mathematical terms, the likelihood function quantifies how probable our observed data are for various possible values of the parameters in a statistical model. We denote these parameters as $\boldsymbol{\theta}$ in this discussion.

In our model, we assume the data is governed by a pdf, which is determined by a specific parameter, represented as $\boldsymbol{\theta}$ within the parameter space $\boldsymbol{\Omega}$. The joint pdf of the data D can be represented as follows:

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

where s_i is the observed system lifetime of the i^{th} system, δ_i is the observed right-censoring indicator of the i^{th} system, 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 $\boldsymbol{\theta}$. When we keep the data constant and allow the parameter $\boldsymbol{\theta}$ to vary, we obtain what is called the likelihood function L, defined as

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

where

$$L_i(\boldsymbol{\theta}) = f(s_i, \delta_i, c_i; \boldsymbol{\theta})$$

is the likelihood contribution of the i^{th} system. In other words, the likelihood function quantifies how likely different parameter values $\boldsymbol{\theta}$ are, given the observed data.

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

$$L_i(\boldsymbol{\theta}) = \begin{cases} R_{T_i}(s_i; \boldsymbol{\theta}) & \text{if } \delta_i = 0 \\ \beta_i R_{T_i}(s_i; \boldsymbol{\theta}) \sum_{j \in c_i} h_j(s_i; \boldsymbol{\theta_j}) & \text{if } \delta_i = 1, \end{cases} \tag{3.3}$$

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 and the component cause of failure is masked by the candidate set is c_i.

In the follow 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 parameters, $\boldsymbol{\theta}$, 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

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

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

We assume the pdf $f_{T_i}(t_i; \boldsymbol{\theta})$ is known, but we do not have knowledge of $\Pr_{\boldsymbol{\theta}}\{\mathcal{C}_i = c_i | T_i = t_i\}$, 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 $\boldsymbol{\theta}$, 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.,*

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

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

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

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 4, the joint pdf of T_i and K_i is given by

$$f_{T_i,K_i}(t_i, j; \boldsymbol{\theta}) = h_j(t_i; \boldsymbol{\theta_j}) R_{T_i}(t_i; \boldsymbol{\theta}),$$

where $h_j(t_i; \boldsymbol{\theta_j})$ is the hazard function for the j^{th} component and $R_{T_i}(t_i; \boldsymbol{\theta})$ is the reliability function of the system. Thus, the joint pdf of T_i, K_i, and $\mathcal{C}_i$ may be written as

$$\begin{aligned} f_{T_i,K_i,\mathcal{C}_i}(t_i, j, c_i; \boldsymbol{\theta}) &= f_{T_i,K_i}(t_i, k; \boldsymbol{\theta}) \Pr_{\boldsymbol{\theta}}\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\} \\ &= h_j(t_i; \boldsymbol{\theta_j}) R_{T_i}(t_i; \boldsymbol{\theta}) \Pr_{\boldsymbol{\theta}}\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\}. \end{aligned} \tag{3.4}$$

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 (3.4),

$$f_{T_i,\mathcal{C}_i}(t_i, c_i; \boldsymbol{\theta}) = R_{T_i}(t_i; \boldsymbol{\theta}) \sum_{j=1}^{m} \Big\{ h_j(t_i; \boldsymbol{\theta_j}) \Pr_{\boldsymbol{\theta}}\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\} \Big\}.$$

By Condition 1, $\Pr_{\boldsymbol{\theta}}\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\} = 0$ 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

$$f_{T_i,\mathcal{C}_i}(t_i, c_i; \boldsymbol{\theta}) = R_{T_i}(t_i; \boldsymbol{\theta}) \sum_{j \in c_i} \Big\{ h_j(t_i; \boldsymbol{\theta_j}) \Pr_{\boldsymbol{\theta}}\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\} \Big\}. \tag{3.5}$$

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

Condition 2. *Any of the components in the candidate set has an equal probability of being the cause of failure. That is, for a fixed $j \in c_i$,*

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

for all $j' \in c_i$.

According to [1], 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, $\Pr_{\boldsymbol{\theta}}\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\}$ may be factored out of the summation in Equation (3.5), and thus the joint pdf of T_i and $\mathcal{C}_i$ may be rewritten as

$$f_{T_i,\mathcal{C}_i}(t_i, c_i; \boldsymbol{\theta}) = \Pr_{\boldsymbol{\theta}}\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j'\} R_{T_i}(t_i; \boldsymbol{\theta}) \sum_{j \in c_i} h_j(t_i; \boldsymbol{\theta_j})$$

where $j' \in c_i$.

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

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

$$f_{T_i,\mathcal{C}_i}(t_i, c_i; \boldsymbol{\theta}) = \beta_i R_{T_i}(t_i; \boldsymbol{\theta}) \sum_{j \in c_i} h_j(t_i; \boldsymbol{\theta_j}).$$

When we fix the sample and allow $\boldsymbol{\theta}$ 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:

$$L_i(\boldsymbol{\theta}) = R_{T_i}(t_i; \boldsymbol{\theta}) \sum_{j \in c_i} h_j(t_i; \boldsymbol{\theta_j}). \tag{3.6}$$

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 (3.6).

3.2 Right-Censored Data

As described in Section **??**, we observe realizations of $(S_i, \delta_i, \mathcal{C}_i)$ where $S_i = \min\{T_i, \tau_i\}$ is the right-censored system lifetime, $\delta_i = 1_{\{T_i < \tau_i\}}$ is the right-censoring 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.

Theorem 6. *The likelihood contribution of a right-censored observation* ($\delta_i = 0$) *is given by*

$$L_i(\boldsymbol{\theta}) = R_{T_i}(s_i; \boldsymbol{\theta}). \tag{3.7}$$

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

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

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

$$L_i(\boldsymbol{\theta}) = R_{T_i}(s_i; \boldsymbol{\theta}).$$

□

When we combine the two likelihood contributions, we obtain the likelihood contribution for the i^{th} system shown in Theorem 5,

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

We use this result in the next section to derive the maximum likelihood estimator of $\boldsymbol{\theta}$.

4 Maximum Likelihood Estimation

We use maximum likelihood estimation (MLE) to estimate the series system parameter $\boldsymbol{\theta}$ given the masked data described in Section **??**. This is achieved by maximizing the likelihood function $L(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ so that, under the assumed model, the observed data is most likely. The point in the parameter space Ω that maximizes the likelihood function is called the maximum likelihood estimate.

According to [2], a point $\hat{\boldsymbol{\theta}}$ in $\boldsymbol{\Omega}$ at which $L(\boldsymbol{\theta})$ is a maximum is called the *maximum likelihood estimate* (MLE) of $\boldsymbol{\theta}$. That is, $\hat{\boldsymbol{\theta}}$ is a value of $\boldsymbol{\theta}$ that satisfies

$$L(\hat{\boldsymbol{\theta}}) = \max_{\boldsymbol{\theta} \in \boldsymbol{\Omega}} L(\boldsymbol{\theta}). \tag{4.1}$$

Essentially, the MLE is a point in the parameter space that is a maximum of the likelihood of the observed data,

$$\hat{\boldsymbol{\theta}} \in \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Omega}} L(\boldsymbol{\theta}).$$

Any point that maximizes the likelihood function also maximizes the log-likelihood function. Thus, for both computational and analytical reasons, we work with the log-likelihood function.

Theorem 7. *The log-likelihood function, $\ell(\boldsymbol{\theta})$, is given by the log of the likelihood function for our masked data model,*

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ell_i(\boldsymbol{\theta}) \tag{4.2}$$

where

$$\ell_i(\boldsymbol{\theta}) = \sum_{j=1}^{m} \log R_j(s_i;\boldsymbol{\theta_j}) + \delta_i \log\big(\sum_{j\in c_i} h_j(s_i;\boldsymbol{\theta_j})\big). \tag{4.3}$$

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

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

Since $\log(A \cdot B) = \log(A) + \log(B)$, we may rewrite the log-likelihood function as

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

By Equation (3.3), L_i is given by

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

We now consider these two cases separately, then combine them to obtain the result in Theorem 7.
Case 1: If the i-th system is right-censored, i.e., $\delta_i = 0$, then

$$\ell_i(\boldsymbol{\theta}) = \log R_{T_i}(s_i;\boldsymbol{\theta}) = \sum_{j=1}^{m} \log R_j(s_i;\boldsymbol{\theta_j}).$$

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

$$\begin{aligned} \ell_i(\boldsymbol{\theta}) &= \log R_{T_i}(s_i;\boldsymbol{\theta})\beta_i \sum_{j\in c_i} h_j(s_i;\boldsymbol{\theta_j}) \\ &= \log R_{T_i}(s_i;\boldsymbol{\theta}) + \log\beta_i + \log\big(\sum_{j\in c_i} h_j(s_i;\boldsymbol{\theta_j})\big). \end{aligned}$$

Since β_i is not a function of $\boldsymbol{\theta}$ (see Condition 3), $\log\beta_i$ is a constant with respect to $\boldsymbol{\theta}$ and so may be ignored in MLE (when we solve the maximum likelihood equations, any terms that do not depend on $\boldsymbol{\theta}$ will be eliminated by the gradient operator). Thus, we may rewrite the above equation as

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

□

4.1 Solving the MLE

According to [2], if $\boldsymbol{\Omega}$ is a Cartesian product of l intervals, partial derivatives of $L(\boldsymbol{\theta})$ exist, and the MLEs do not occur on the boundary of $\boldsymbol{\Omega}$, then the MLEs will be solutions of the simultaneous equations

$$\frac{\partial}{\partial \theta_j} \ell(\boldsymbol{\theta}) = 0 \tag{4.4}$$

for $j = 1, \ldots, p$ where p is the number of components in $\boldsymbol{\theta}$. We call these equations the *maximum likelihood equations*. If multiple solutions to Equation (4.4) exist, each solution that maximizes the likelihood function is a valid MLE.

If there is no closed-form solution to the maximum likelihood equations (4.4), we may use iterative root-finding methods to numerically approximate a solution. A general approach is, if we have a guess $\boldsymbol{\theta}^{(n)}$, take a step in a "promising" direction $\boldsymbol{d}^{(n)}$ to obtain the next guess,

$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}^{(n)} + \alpha^{(n)} \boldsymbol{d}^{(n)}, \tag{4.5}$$

where $\boldsymbol{\theta}^{(0)}$ is an initial guess in the parameter space Ω that is sufficiently close to the MLE $\hat{\boldsymbol{\theta}}$. In our case, we use the parameter vector $\boldsymbol{\theta}^{(0)} = \boldsymbol{\theta}$ as our initial guess, since we know the true value $\boldsymbol{\theta}$ in our simulation studies, but if plausible initial guesses are not known, then global methods may be used to find a good initial guess, like Simulated Annealing.

Assuming that at $\boldsymbol{\theta}^{(n)}$, a sufficiently small step in the direction $\boldsymbol{d}^{(n)}$ results in an improvement with respect to the log-likelihood function, we say that we *overshoot* if

$$\ell(\boldsymbol{\theta}^{(n+1)}) < \ell(\boldsymbol{\theta}^{(n)}).$$

The value $\alpha^{(n)}$ in Equation (4.5) is a positive real number chosen by a *line search* method so that we do not overshot, which has an optimal value given by

$$\alpha^{(n)} \in \operatorname{argmax}_{\alpha} \ell(\boldsymbol{\theta}^{(n)} + \alpha \boldsymbol{d}^{(n)}).$$

However, this may be too computationally expensive to compute, and so we use a less optimal but faster method known as *backtracking*. In the backtracking line search method, we determine $\alpha^{(n)}$ by initially letting $\alpha^{(n)} = 1$ and then, if we overshoot, redo the update with $\alpha^{(n)} \leftarrow r\alpha^{(n)}$, $0 < r < 1$, repeating until we do not overshoot.

Note that this is not necessarily the best course of action for finding global maximums, since depending on our initial guess $\boldsymbol{\theta}^{(0)}$, we may get stuck in a local maximum. A global search method, like Simulated Annealing, may be used to find better initial guesses.

We do as many iterations in Equation (4.5) as necessary to satisfy some *stopping condition*, which is usually something simple like the distance between $\boldsymbol{\theta}^{(n)}$ and $\boldsymbol{\theta}^{(n+1)}$ being sufficiently small. Under the right conditions, for sufficiently large n, $\boldsymbol{\theta}^{(n)} \approx \hat{\boldsymbol{\theta}}$.

We use a popular technique known as Newton-Raphson method, which is obtained by letting $\boldsymbol{d}^{(n)}$ be defined as

$$\boldsymbol{d}^{(n)} = -J^{-1}(\boldsymbol{\theta}^{(n)}) \nabla \ell(\boldsymbol{\theta}^{(n)}),$$

where $J(\boldsymbol{\theta}^{(n)})$ and $\nabla \ell(\boldsymbol{\theta}^{(n)})$ are respectively the observed Fisher information matrix (Hessian of the log-likelihood function) and the score (gradient of the log-likelihood function), each evaluated at $\boldsymbol{\theta}^{(n)}$.

4.2 Properties of the MLE

In this section, we discuss some properties of the MLE that are useful for making statistical inferences about the parameter vector $\boldsymbol{\theta}$. According to [2], if certain regularity conditions are satisfied, then solutions of the maximum likelihood equation (4.4) have the following desirable properties:

1. $\hat{\boldsymbol{\theta}}$ exists and is unique.
2. $\hat{\boldsymbol{\theta}}$ is an asymptotically unbiased estimator.

3. $\hat{\boldsymbol{\theta}}$ is asymptotically the UMVUE, the uniform minimum variance unbiased estimator.

4. $\hat{\boldsymbol{\theta}}$ is asymptotically normal with a mean $\boldsymbol{\theta}$ and a variance-covariance matrix that is the inverse of the Fisher information matrix (FIM), whose (i, j)-th component is given by

$$I(\boldsymbol{\theta})_{ij} = nE_{\boldsymbol{\theta}}\left(-\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log f(S_i, \delta_i, \mathcal{C}_i; \boldsymbol{\theta})\right).$$

Usually, the *observed* FIM is used instead of the FIM, which is a conditioned on the observed data,

$$J(\boldsymbol{\theta})_{ij} = -\frac{\partial}{\partial\theta_i\partial\theta_j}\ell(\boldsymbol{\theta}).$$

If $\boldsymbol{\theta}$ is unknown, $J(\boldsymbol{\theta})$ may be estimated with $J(\hat{\boldsymbol{\theta}})$. Thus, assuming the regularity conditions are satisfied, then approximately,

$$\hat{\boldsymbol{\theta}} \sim \mathcal{N}(\hat{\boldsymbol{\theta}}, J^{-1}(\hat{\boldsymbol{\theta}}))$$

and as the sample size goes to infinity, $\hat{\boldsymbol{\theta}}$ converges in distribution to $\mathcal{N}(\hat{\boldsymbol{\theta}}, J^{-1}(\hat{\boldsymbol{\theta}}))$.

This is the asymptotic sampling distribution of MLE, but for small samples, it can be a very poor approximation. In the next section, we discuss the bootstrap method. In particular, bootstrapping the variance and confidence intervals.

5 Bootstrapping the Variance and Confidence Intervals of the MLE

The bootstrap method is a powerful, general purpose tool for estimating the sampling distribution of a statistic, in our case statistics of the MLE, that does not rely on making strong assumptions about the underlying distribution of the data.

The most common form of the Bootstrap method is the non-parametric Bootstrap. In the non-parametric bootstrap, the random data is created by resampling with replacement from the original data and then computing the statistic of interest on the resampled data. This is repeated B times, giving us B bootstrap replicates of the statistic. The sampling distribution of the statistic is then approximated by the empirical distribution of the bootstrap replicates. Since we do not know (nor do we attempt to model) the way candidate sets are generated, this non-parametric form is ideal.

We are particularly interested in two statistics of the MLE: the variance and the confidence interval. We will bootstrap the confidence interval using the percentile method, which does not explicitly depend on the variance estimate. However, conceptually, they are still related: a higher variance should generally lead to a larger confidence interval.

In our simulation study, we will assess the performance of the bootstrapped variance by comparing it to the empirical variance of the MLE, and we will assess the performance of the bootstrapped confidence interval by computing its coverage probability, i.e., a 95% confidence interval should contain the true value 95% of the time. We say that a confidence interval has *good coverage* if its coverage probability is close to the nominal confidence level.

If the confidence intervals have good coverage, a small confidence interval width means we are more confident that the true value of $\boldsymbol{\theta}$ is close to the MLE, and a large confidence interval width means we are less confident that the true value of $\boldsymbol{\theta}$ is close to the MLE. However, if the confidence interval has poor coverage, then the confidence interval width is not particularly informative. Thus, we see that the confidence interval is only useful if it has good coverage, and so we will focus on assessing the coverage probability of the confidence intervals in our simulation study.

6 Simulation Study: Series System with Weibull Components

In the real world, systems are quite complex:

1. They are not perfect series systems.
2. The components in a system are not independent.
3. The lifetimes of the components are not precisely modeled by any named probability distributions.
4. The components may depend on many other unobserved factors.

With these caveats in mind, we model the data as coming from a Weibull series system of $m = 5$ components, and other factors, like ambient temperature, are either negligible (on the distribution of component lifetimes) or are more or less constant.

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

$$T_{ij} \sim \text{WEI}(\boldsymbol{\theta_j})$$

where $\boldsymbol{\theta_j} = (k_j, \lambda_j)$ for $j = 1, \ldots, m$. Thus, $\boldsymbol{\theta} = (\boldsymbol{\theta_1}, \ldots, \boldsymbol{\theta_m})' = \big(k_1, \lambda_1, \ldots, k_m, \lambda_m\big)$. The random variable T_{ij} has a reliability function, pdf, and hazard function given respectively by

$$R_j(t; \lambda_j, k_j) = \exp\left\{-\left(\frac{t}{\lambda_j}\right)^{k_j}\right\}, \tag{6.1}$$

$$f_j(t; \lambda_j, k_j) = \frac{k_j}{\lambda_j}\left(\frac{t}{\lambda_j}\right)^{k_j - 1} \exp\left\{-\left(\frac{t}{\lambda_j}\right)^{k_j}\right\}, \tag{6.2}$$

$$h_j(t; \lambda_j, k_j) = \frac{k_j}{\lambda_j}\left(\frac{t}{\lambda_j}\right)^{k_j - 1} \tag{6.3}$$

where $t > 0$ is the lifetime, $\lambda_j > 0$ is the scale parameter and $k_j > 0$ is the shape parameter. The shape parameters $k_1, \ldots, k_m$ have the following interpretations:

$k_j < 1$ The hazard function decreases with respect to time. For instance, this may occur as a result of defective components being weeded out early. This is known as the *infant mortality* phase.

$k_j = 1$ The hazard function is constant with respect to time. This is an idealized case that is rarely observed in practice, but may be useful for modeling purposes.

$k_j > 1$ The hazard function increases with respect to time. For instance, this may occur as a result of components wearing out. This is known as the *aging* phase.

The lifetime of the series system composed of m Weibull components has a reliability function given by

$$R(t; \boldsymbol{\theta}) = \exp\left\{-\sum_{j=1}^{m}\left(\frac{t}{\lambda_j}\right)^{k_j}\right\}. \tag{6.4}$$

Proof. By Theorem 1,

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

Plugging in the Weibull component reliability functions obtains the result

$$\begin{aligned} R(t; \boldsymbol{\theta}) &= \prod_{j=1}^{m} \exp\left\{-\left(\frac{t}{\lambda_j}\right)^{k_j}\right\} \\ &= \exp\left\{-\sum_{j=1}^{m}\left(\frac{t}{\lambda_j}\right)^{k_j}\right\}. \end{aligned}$$

□

The Weibull series system's hazard function is given by

$$h(t;\boldsymbol{\theta}) = \sum_{j=1}^{m} \frac{k_j}{\lambda_j}\left(\frac{t}{\lambda_j}\right)^{k_j-1}, \tag{6.5}$$

whose proof follows from Theorem 3.

The pdf of the series system is given by

$$f(t;\boldsymbol{\theta}) = \left\{\sum_{j=1}^{m} \frac{k_j}{\lambda_j}\left(\frac{t}{\lambda_j}\right)^{k_j-1}\right\} \exp\left\{-\sum_{j=1}^{m}\left(\frac{t}{\lambda_j}\right)^{k_j}\right\}. \tag{6.6}$$

Proof. By definition,

$$f(t;\boldsymbol{\theta}) = h(t;\boldsymbol{\theta})R(t;\boldsymbol{\theta}).$$

Plugging in the failure rate and reliability functions given respectively by Equations (6.4) and (6.5) completes the proof. □

6.1 System Reliability

A series system is only as reliable as its least reliable component. In order to make the simulation study representative of real-world scenarios, at least for systems designed to be reliable, we choose parameter values that are representative of real-world systems where there is no single component that is much less reliable than the others.

One way to define reliability is by the mean time to failure (MTTF), which is the expected value of the lifetime, which for the Weibull distribution is given by

$$\text{MTTF} = k\,\Gamma(1 + 1/\lambda),$$

where Γ is the gamma function.

We consider the data from [3], which includes a study of the reliability of a series system with three Weibull components with shape and scale parameters given by

$$\begin{aligned} k_1 &= 1.2576 \quad \lambda_1 = 994.3661 \\ k_2 &= 1.1635 \quad \lambda_2 = 908.9458 \\ k_3 &= 1.1308 \quad \lambda_3 = 840.1141. \end{aligned} \tag{6.7}$$

Our approach is to extend this system to a five component system by adding two more components with shape and scale parameters given by

$$\begin{aligned} k_4 &= 1.1802 \quad \lambda_4 = 940.1141 \\ k_5 &= 1.3311 \quad \lambda_5 = 836.1123. \end{aligned} \tag{6.8}$$

Table 2: Meean Time To Failure of Weibull Components and Series System

	MTTF
Component 1	924.8693
Component 2	862.1568
Component 3	803.5639
Component 4	888.2181
Component 5	768.6793
Series System	223.0336

As shown by Table 2, there are no components that are significantly less reliable than any of the others. Note that a series system in which, say, one of the components does have a significantly shorter MTTF would also pose significant challenges to estimating the parameters of the system from our masked failure data, since the failure time of the series system would be dominated by the failure time of the least reliable component.

6.2 Weibull Likelihood Model for Masked Data

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 8. *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*

$$L_i(\boldsymbol{\theta}) = \begin{cases} \exp\left\{-\sum_{j=1}^{m}\left(\frac{t_i}{\lambda_j}\right)^{k_j}\right\}\beta_i \sum_{j\in c_i}\frac{k_j}{\lambda_j}\left(\frac{t_i}{\lambda_j}\right)^{k_j-1} & \text{if } \delta_i = 1, \\ \exp\left\{-\sum_{j=1}^{m}\left(\frac{t_i}{\lambda_j}\right)^{k_j}\right\} & \text{if } \delta_i = 0. \end{cases} \tag{6.9}$$

Proof. By Theorem 5, the likelihood contribution of the i-th system is given by

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

By Equation (6.4), the system reliability function R_{T_i} is given by

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

and by Equation (6.3), the Weibull component hazard function h_j is given by

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

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

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

Corollary 1. *The log-likelihood contribution of the i-th system is given by*

$$\ell_i(\boldsymbol{\theta}) = -\sum_{j=1}^{m}\left(\frac{t_i}{\lambda_j}\right)^{k_j} + \delta_i \log\left(\sum_{j\in c_i}\frac{k_j}{\lambda_j}\left(\frac{t_i}{\lambda_j}\right)^{k_j-1}\right) \tag{6.10}$$

where we drop any terms that do not depend on $\boldsymbol{\theta}$ since they do not affect the MLE.

Since the systems are independent, the log-likelihood of the entire sample of n observations is given by

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n}\ell_i(\boldsymbol{\theta}). \tag{6.11}$$

6.3 Numerically Solving the MLE

We may find an MLE by solving the maximum likelihood equation (4.4), i.e., a point $\hat{\boldsymbol{\theta}} = (\hat{k}_1, \hat{\lambda}_1, \ldots, \hat{k}_m, \hat{\lambda}_m)$ satisfying $\nabla_\theta \ell(\hat{\boldsymbol{\theta}}) = \mathbf{0}$, where $\nabla_{\boldsymbol{\theta}}$ is the gradient of the log-likelihood function with respect to $\boldsymbol{\theta}$, otherwise known as the score.

To solve these ML equations, we use the Newton-Raphson method described in Section 4.1. In order to use the Newton-Raphson method, we need to compute the gradient and Hessian of the log-likelihood function.

We analytically derive the gradient of the log-likelihood function (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:

1. The Hessian is not necessarily needed since we often use some faster method to approximate the Hessian, e.g., the BFGS method. Technically, we could also numerically approximate the gradient too, but the gradient is much easier to derive than the Hessian, and moreover, knowing the score precisely also enables us to more accurately approximate the Hessian by taking the Jacobian of the gradient.

2. The Hessian is more difficult to derive than the score, and so it is more likely that we will make a mistake when deriving the Hessian.

The following theorem derives the score function.

Theorem 9. *The score function of the log-likelihood contribution of the i-th Weibull series system is given by*

$$\nabla \ell_i(\boldsymbol{\theta}) = \left(\frac{\partial \ell_i(\boldsymbol{\theta})}{\partial k_1}, \frac{\partial \ell_i(\boldsymbol{\theta})}{\partial \lambda_1}, \cdots, \frac{\partial \ell_i(\boldsymbol{\theta})}{\partial k_m}, \frac{\partial \ell_i(\boldsymbol{\theta})}{\partial \lambda_m} \right)', \tag{6.12}$$

where

$$\frac{\partial \ell_i(\boldsymbol{\theta})}{\partial k_r} = -\left(\frac{t_i}{\lambda_r}\right)^{k_r} \log\left(\frac{t_i}{\lambda_r}\right) + \frac{\frac{1}{t_i}\left(\frac{t_i}{\lambda_r}\right)^{k_r}\left(1 + k_r \log\left(\frac{t_i}{\lambda_r}\right)\right)}{\sum_{j \in c_i} \frac{k_j}{\lambda_j}\left(\frac{t_i}{\lambda_j}\right)^{k_j - 1}} 1_{\delta_i = 1 \wedge r \in c_i} \tag{6.13}$$

and

$$\frac{\partial \ell_i(\boldsymbol{\theta})}{\partial \lambda_r} = \frac{k_r}{\lambda_r}\left(\frac{t_i}{\lambda_r}\right)^{k_r} - \frac{\left(\frac{k_r}{\lambda_r}\right)^2\left(\frac{t_i}{\lambda_r}\right)^{k_r - 1}}{\sum_{j \in c_i} \frac{k_j}{\lambda_j}\left(\frac{t_i}{\lambda_j}\right)^{k_j - 1}} 1_{\delta_i = 1 \wedge r \in c_i} \tag{6.14}$$

The result follows from taking the partial derivatives of the log-likelihood contribution of the i-th system given by Equation (6.9). 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 same of their gradients, and so the score function of the entire sample is given by

$$\nabla \ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \nabla \ell_i(\boldsymbol{\theta}). \tag{6.15}$$

6.3.1 Issues with the MLE

Identifiability: When estimating the parameters of latent components, we must be careful to ensure that the parameters are identifiable such that the likelihood function is maximized at a unique point. If the likelihood function is not maximized at a unique point, then the MLE is not unique, and a lot of the theory we have developed so far breaks down.

One way in which this problem may arise is if the data is not informative enough. For example, if we have a series system and in the observed masked data component 1 is in the candidate set if and only if component 2 is in candidate set, then we do not have enough information to estimate the parameters of component 1 and component 2 separately. In this case, we could combine these two components into a single component. We lagely avoid this problem by using the Bernoulli candidate set model, but sometimes it may still arise by chance.

Parameter rescaling: When the parameters under investigation span different orders of magnitude, parameter rescaling can significantly improve the performance and reliability of optimization algorithms. Parameter rescaling gives an optimizer a sense of the typical size of each parameter, enabling it to adjust its steps accordingly. This is crucial in scenarios like ours, where shape and scale parametes are a few orders of magnitude apart. Without rescaling, the optimization routine may struggle, taking numerous small steps for larger parameters and overshooting for smaller ones.

Speed of convergence was particularly important in our case, since in our simulation study, we employ the Bootstrap method to estimate the sampling distribution of the MLE, which requires us to estimate the MLE for many data sets. We found that parameter rescaling significantly improved the speed of convergence, which allowed us to run our simulation study in a tractable amount of time.

In the `optim` algorithm in the R package `stats`, we achieve this result by assigning a `parscale` vector in line with the parameter magnitudes. It does not matter what the values of the `parscale` vector are, only

their relative magnitudes. It does not need to be very precise, but since we are doing a simulation study, we know the true parameter values and can use that information to scale them appropriately.

6.4 Simulation Design

In this section, we describe the design of our simulation study. We first describe the simulation scenarios we consider, and then we describe how we generate data for each scenario.

6.4.1 Bernoulli Candidate Set Model

In our simulation study, 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, each non-failed component is included in the candidate set with a fixed probability p, independently of all other components and independently of $\boldsymbol{\theta}$, and the failed component is always included in the candidate set.

6.4.2 Right-Censoring Model

We employ a very simple right-censoring model, where the right-censoring time τ is fixed and independent of $\boldsymbol{\theta}$ and the censoring time S_i of the i^{th} system.

We parameterize τ by quantiles of the series system, e.g., if $q = 0.8$, then $\tau(q)$ is the 80% quantile of the series system such that 80% of the series systems are observed (fail before time $\tau(q)$) and 20% of the series systems fail after time $\tau(q)$ (are right-censored).

6.4.3 Scenarios

we vary the sample size n, the Bernoulli masking probability p of including each non-failed component in the candidate set, and the right-censoring time τ. We then analyze the performance of the MLE under these various scenarios.

Here is an outline of the simulation study analysis:

1. Set up simulation parameters for various scenarios of interest, such as generating data to examine the relationship between bias and masking probability for different sample sizes and right-censoring times.

2. Generate R data sets for each scenario (some combination of n, p, and τ).

3. Estimate the parameters for each data set, giving us R estimates of the parameters. We use these data sets as an empirical estimate of the sampling distribution of the MLE for each scenario.

4. Using the empirical sampling distribution of the MLE, estimate various performance measures of the MLE, like bias, variance, MSE, and coverage probability for each scenario.

5. Analyze and visualize the results, e.g., by plotting the bias, variance, MSE, and coverage probability as a function of n for different combinations of p and τ.

   We then interpret the results and discuss the performance of the MLE estimator under various conditions. We expect that as $n \to \infty$, the bias and MSE will go to 0 and the coverage probability will go to 0.95 (when constructing 95% confidence intervals). Of course, we do not expect these results to hold for finite n, but we would like to see how the bias, MSE, and coverage probability change as we vary n, p, and τ.

For how we generate a scenario, see Appendix A.

So, now we just resample from the data with replacement, and fit the Weibull series model to each bootstrap sample. We do this $B = 1000$ times, giving us B bootstrap replicates of the MLE $\hat{\boldsymbol{\theta}}^{(1)}, \ldots, \hat{\boldsymbol{\theta}}^{(B)}$.

As a ground truth, we will use the empirical distribution of the MLE under our data model under a variety of simulation scenarios where we vary the sample size, the right censoring time, and the so-called masking probability of the candidate sets, where a higher masking probability means that the candidate sets are more likely to contain non-failed components.

6.4.4 Verification

To verify that our likelihood model is correct, we load the Table 2 data from [3] and fit the Weibull series model to the data to see if we can recover the MLE they reported. When we fit the Weibull series model to this data by maximizing the likelihood function, we obtain the following fit for the shape and scale parameters given respectively by

$$\hat{k}_1 = 1.2576, \hat{k}_2 = 1.1635, \hat{k}_3 = 1.1308,$$

and

$$\hat{\lambda}_1 = 994.3661, \hat{\lambda}_2 = 908.9458, \hat{\lambda}_3 = 840.1141,$$

which is in agreement with the MLE they reported. Satisfied that our likelihood model is correct, we proceed with the simulation study.

6.5 Bias, variance, and MSE of the MLE

First, we estimate the bias, variance, and MSE of the MLE under various scenarios. This is useful for understanding the accuracy and precision of the MLE under different conditions. It is unrelated to the bootstrap method, but it is useful to compute these quantities before we assess the bootstrapped variance and confidence intervals.

A measure of the accuracy of $\hat{\boldsymbol{\theta}}$ is the bias, which is defined as

$$\mathrm{b}(\hat{\boldsymbol{\theta}}) = E(\hat{\boldsymbol{\theta}}) - \boldsymbol{\theta}.$$

We cannot analytically derive the bias, so we estimate the bias using the empirical sampling distribution,

$$\hat{\mathrm{b}}(\hat{\theta}_j) = E_{\hat{\boldsymbol{\theta}} \sim \text{data}}(\hat{\boldsymbol{\theta}}) - \theta_j.$$

We estimate the precision of $\hat{\theta}_j$ with the variance and MSE. The variance of $\hat{\boldsymbol{\theta}}$ is defined as

$$\mathrm{Var}(\hat{\theta}_j) = E_{\hat{\boldsymbol{\theta}} \sim \text{data}}\big((\hat{\theta}_j - E_{\hat{\boldsymbol{\theta}} \sim \text{data}}(\hat{\theta}_j))^2\big),$$

where the expectation is taken with respect to the empirical sampling distribution. The mean squared error is a measure of estimator error that incorporates both the bias and the variance, and is defined as

$$\mathrm{MSE}(\hat{\theta}_j) = E_{\hat{\boldsymbol{\theta}} \sim \text{data}}\big((\hat{\theta}_j - \theta_j)^2\big).$$

Assuming the regularity conditions for the MLE are met, the MSE converges in probability to the variance.

6.6 Simulation Scenarios

We consider many different scenarios, where we vary the sample size n, the masking probability p, and the right-censoring time τ. We then analyze the performance of the MLE under these various scenarios by estimating the bias, variance, and MSE of the MLE.

Absolute bias vs. sample size with a masking probability but no right-censoring

In this scenario, we want to see the bias of the MLE as a function of the sample size n from $n = 30$ to $n = 800$ for a fixed masking probability $p = 0.2$ and no right-censoring ($\tau = \infty$). Recall that the masking probability is the probability of including each non-failed component.

In Figure 2, we plot the absolute bias $|\operatorname{bias}(\hat{\theta})|$ on a log scale against the sample size. However, because the absolute bias is quite large for small sample sizes and small for large sample sizes, we use a log scale.

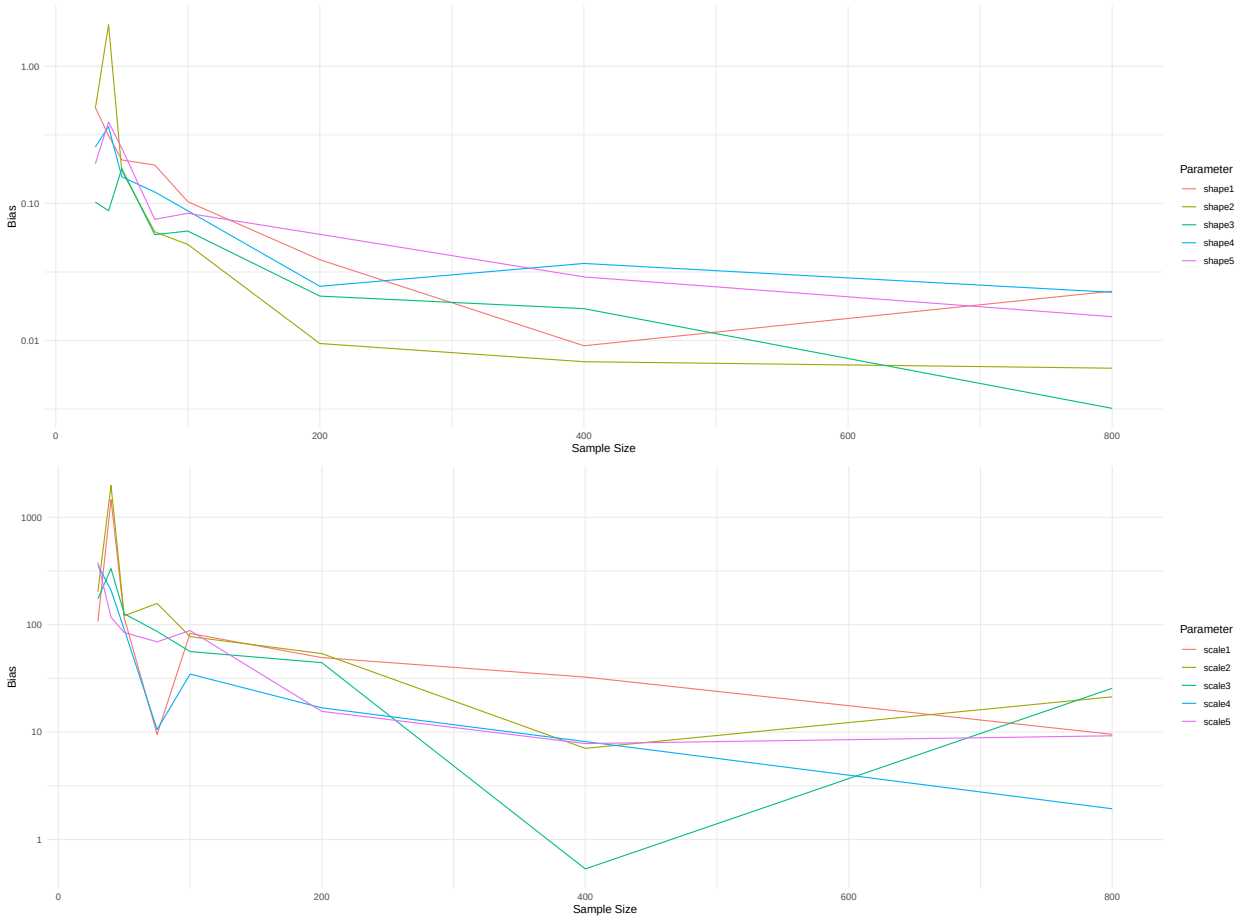

Figure 2: Bias vs. sample size (masking probability 0.2)

Furthermore, we show the absolute bias for the shape and scale parameters separately, since the scale parameters are much larger than the shape parameters.

Here are some important observations Figure 2 reveals:

1. For both shape and scale parameters, we see that the absolute bias seems to be decreasing to zero as the sample size increases. This is not surprising since we expect the MLE to be consistent, i.e., $\hat{\boldsymbol{\theta}}$ converges in probability to $\boldsymbol{\theta}$ as the sample size increases to infinity. Still, it is reassuring to see that the bias seems to be behaving as expected.

2. For the shape parameters, which are small (the shape parameters have true values a little larger than 1), the bias is relatively large for sample sizes up to 100.

3. For the scale parameters, which are quite large (the scale parameters have true values around 1000). Like with the shape parameters, the bias is relatively large for sample sizes up to 100, but seems to stabilize and reach relatively small values after that.

Scenario: Bias vs. sample size and masking probability and no right-censoring

Now, we take a larger view and plot the bias (without taking its absolute value as we had done previously) against the masking probabilities $p = 0$ (no masking) to $p = 0.4$ (significant masking) for sample sizes 100, 400, and 800.

For the shape parameters, at a sample of size 100, we see significant bias and we also see that it is very sensitive to the masking probability. See Figure 3. However, for sample sizes of 400 and 800, the bias is relatively small and unaffected by the masking probability.

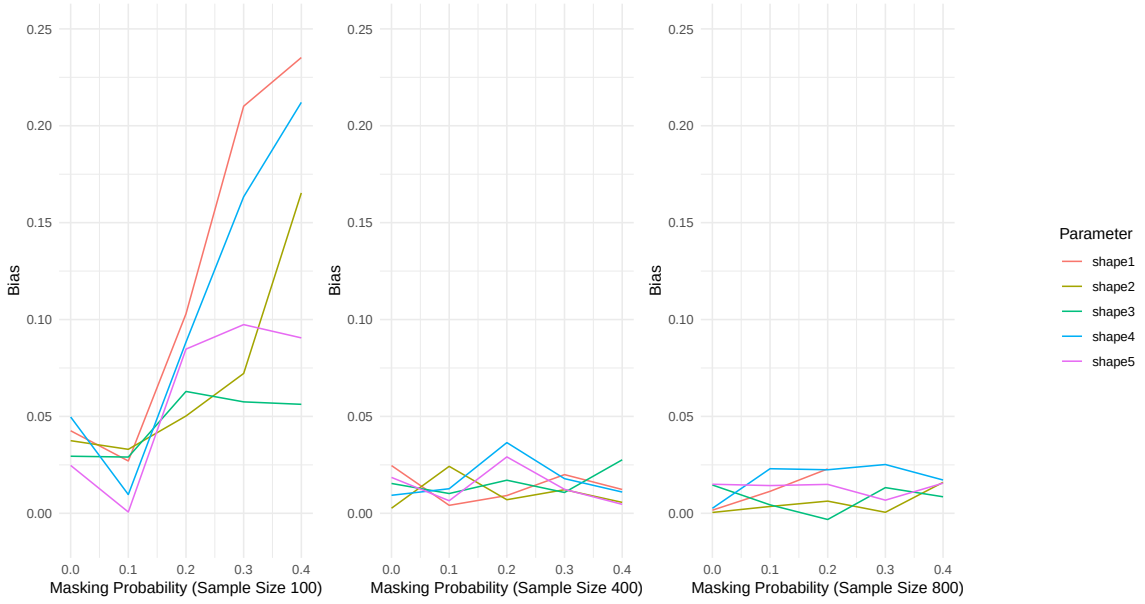

Figure 3: Shape Bias vs. masking probability for sample sizes 100, 400, and 800

For the scale parameters, a similar pattern emerges, although we see that even for sample size 400, there is evidence that the bias is still affected by the masking probability. See Figure 4.

The smallest bias, as expected, occurs for sample sizes of 800. The bias for λ_1 (scale parameter 1) at the masking probability 0.3 is an interesting case, since it jumps up at that point for some reason. We used only $R = 100$ replications, so it is plausible it would decrease with more replications. Regardless, the overall trend is that the bias decreases as the sample size increases, and its dependence on the masking probability is relatively small with sufficiently large sample sizes.

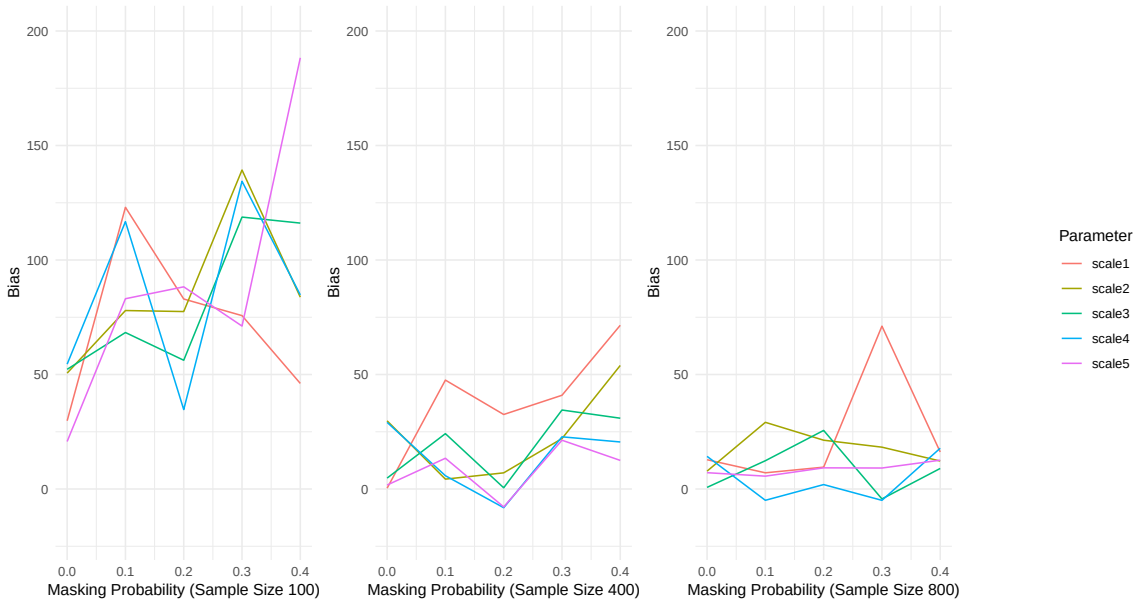

Figure 4: Scale Bias vs. masking probability for sample sizes 100, 400, and 800

Scenario: Bias vs. right-censoring time and sample size with a fixed masking probability In this scenario, we want to isolate the effect of the right-censoring time τ on the bias. We fix the masking probability to $p = 0.215$, in line with the masking probability we estimate for the Table 2 data set in [3].

We plot the bias against the right-censoring time for sample sizes 50, 150, and 300. See Figure 5. On the x-axis, we report the right-censoring time as a quantile of the Weibull series distribution so that we can more clearly see the effect of the right-censoring on the bias, e.g., the 50% quantile is the time at which 50% of the systems are expected to fail.

A few observations about Figure 5:

1. The bias decreases as the right-censoring time increases. This is expected since we have more information about the system when the right-censoring time is larger.

2. The bias decreases as the sample size increases, which is also expected since we have more information about the system when the sample size is larger.

3. The bias is relatively small for sample sizes 150 and 300, but for sample size 50, the bias is quite large, particularly for the shape parameters. This is not surprising since the sample size is quite small, and so we do not expect the MLE to be very accurate.

6.6.1 Variance

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x gridExtra::combine() masks dplyr::combine()
## x dplyr::filter()      masks stats::filter()
## x dplyr::lag()         masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
## Rows: 64000 Columns: 55
## -- Column specification --------------------------------------------------------
```

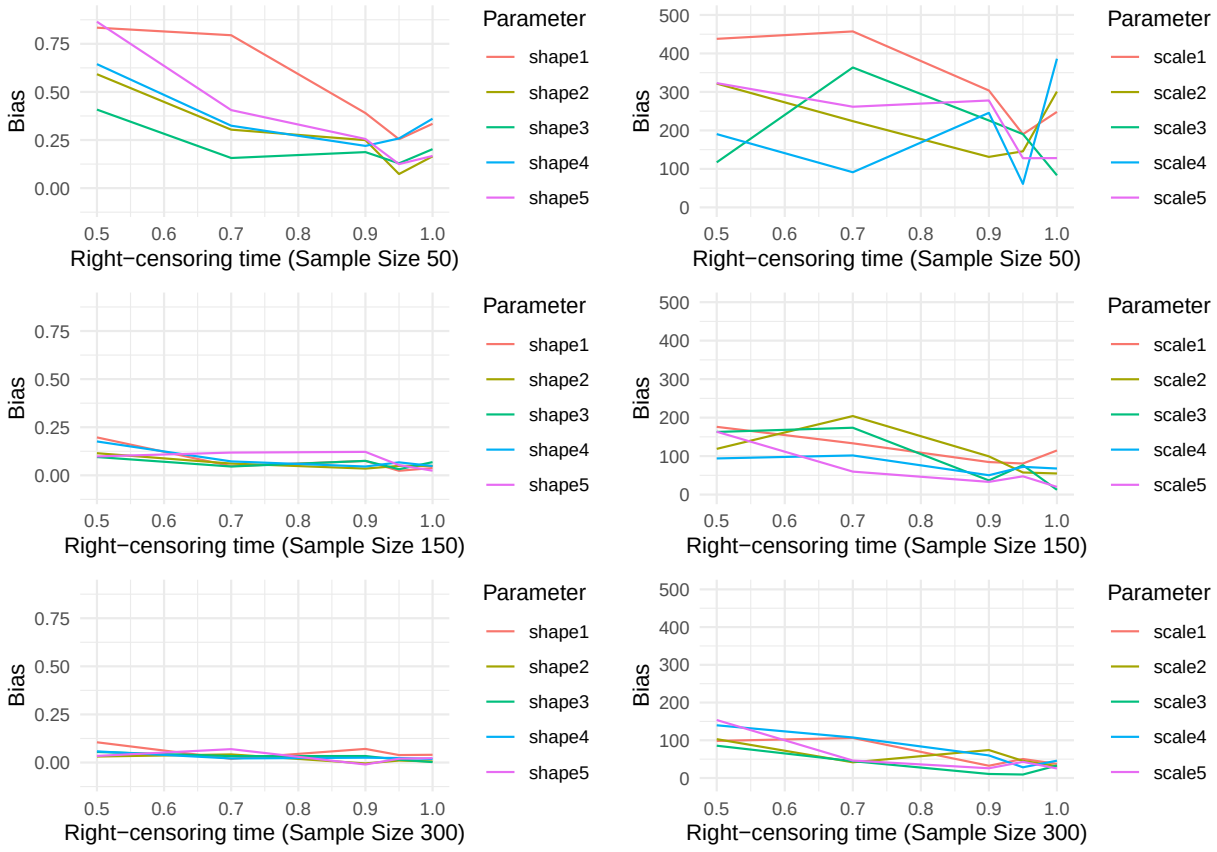

Figure 5: Bias vs. right-censoring time and sample sizes 50, 150, and 300

```
## Delimiter: ","
## dbl (45): n, p, q, tau, mle.1, mle.2, mle.3, mle.4, mle.5, mle.6, mle.7, mle...
## lgl (10): coverages.1, coverages.2, coverages.3, coverages.4, coverages.5, c...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

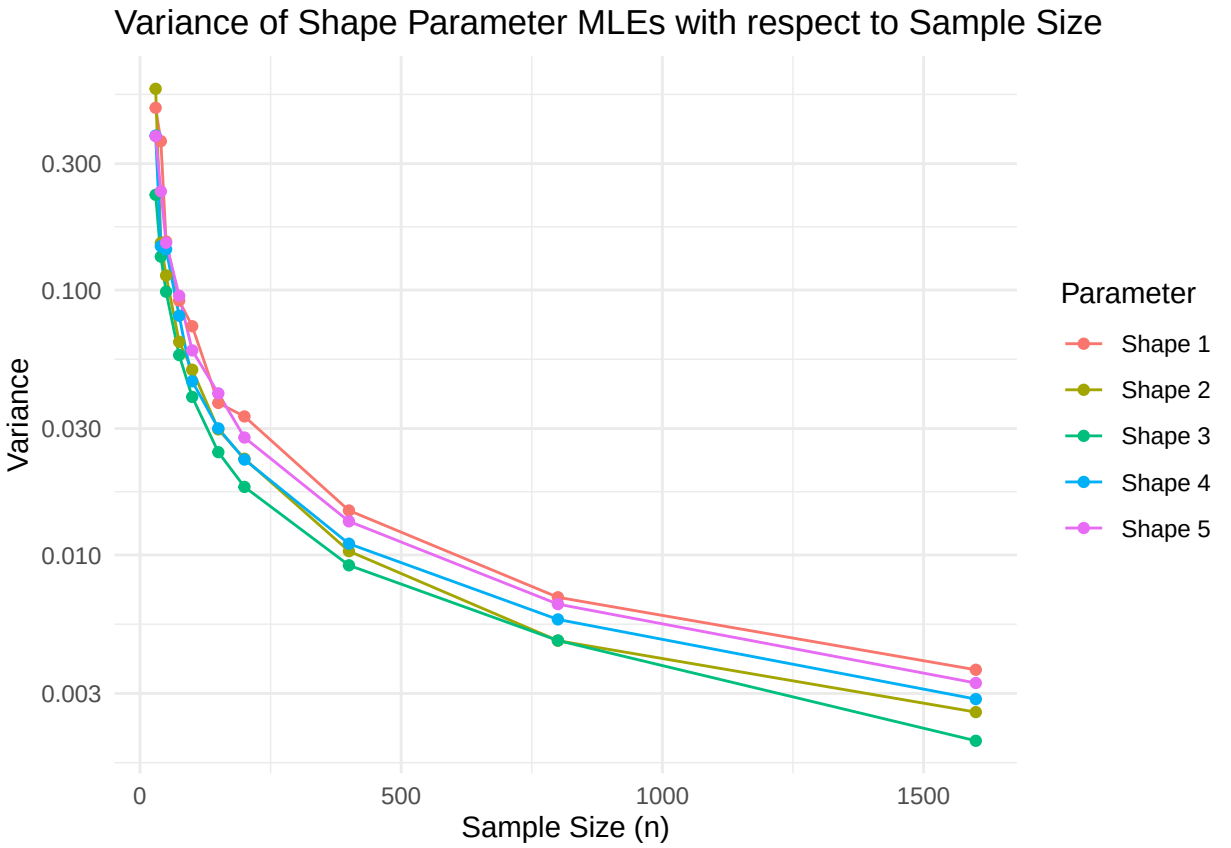

Figure 6: Variance vs. sample size

6.7 Coverage Probability of Bootstrapped Confidence Intervals

Under a variety of scenarios, we will bootstrap a 95%-confidence interval for $\boldsymbol{\theta}$ using the percentile method, and we will evaluate its accuracy by computing the coverage probability.

We want the coverage probability to be close to the nominal level, 95%, because if the coverage probability is too low, then we will be too confident in the precision and accuracy of the MLE, and if the coverage probability is too high, then we will not be confident enough in the precision and accuracy of the MLE.

To estimate the coverage probability, we use the following procedure:

1. For a given scenario, we generate $R = 300$ data sets.
2. We find an MLE for each of R data sets.
3. We bootstrap the 95%-confidence interval for each MLE.
4. We compute the coverage probability by computing the proportion of times the true parameter $\boldsymbol{\theta}$ is contained in 95%-confidence interval.

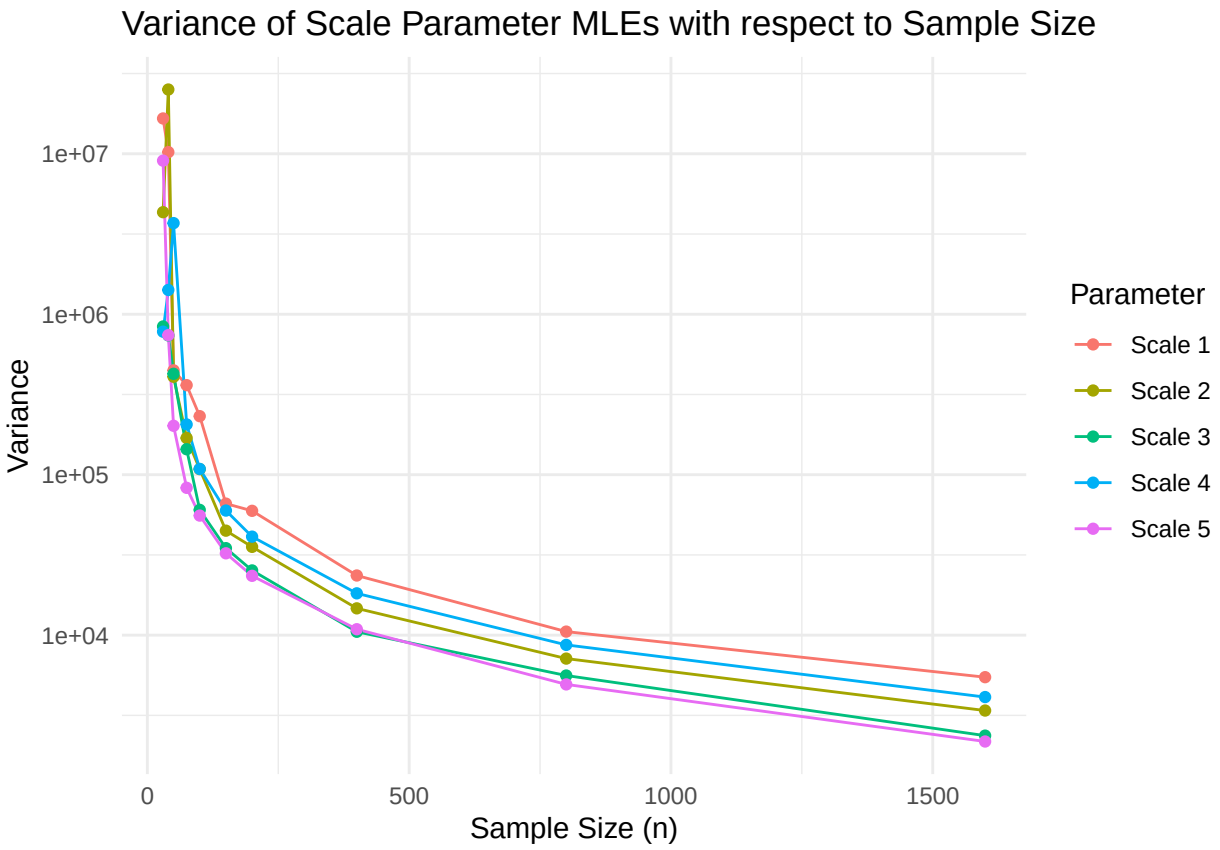

Figure 7: Variance vs. sample size

6.8 Simulation Scenarios

Scenario: Coverage probability vs. sample size with a fixed masking probability and no right-censoring

We want to isolate the effect of the coverage probability as a function of the sample size n. We fix the masking probability to $p = 0.2$ and without right-censoring ($\tau = \infty$) and vary the sample size from $n = 30$ to $n = 800$. See Figure 8.

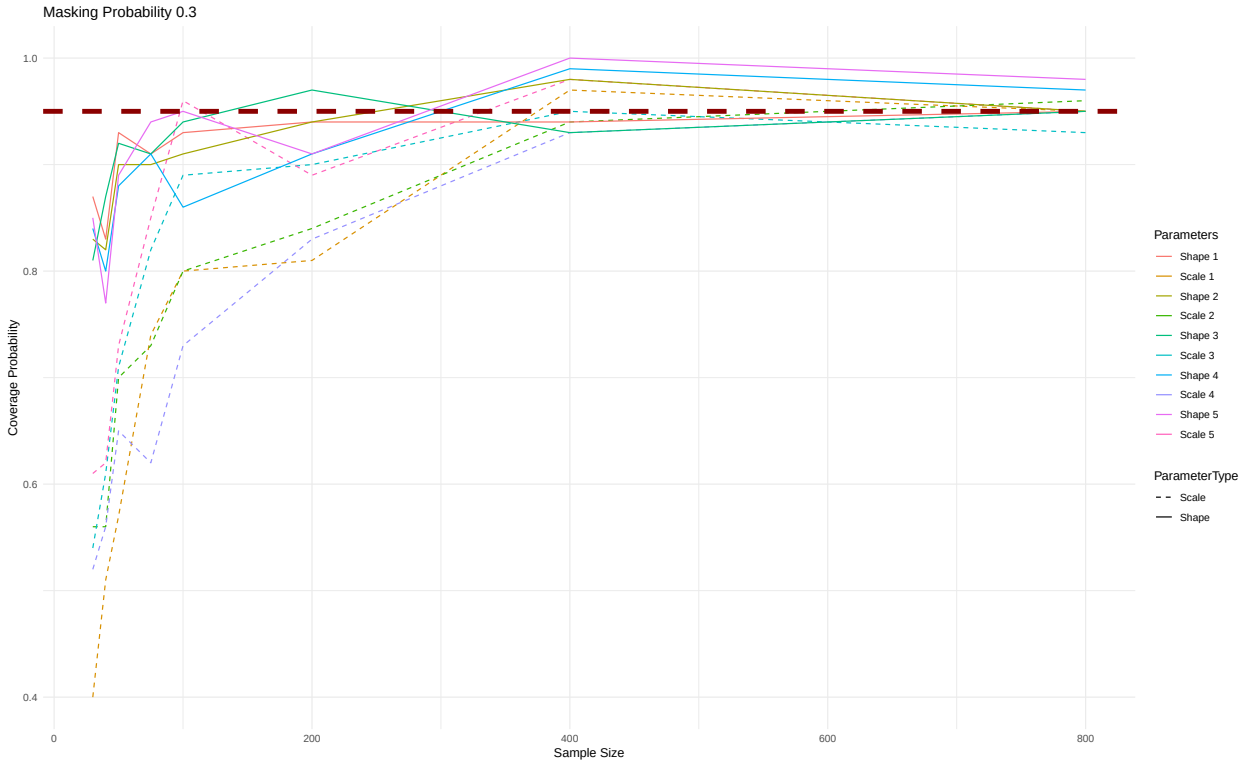

Figure 8: Coverage probability vs. sample size for masking probability 0.3

Here are some key observations:

1. It is immediately obvious that the scale parameters (dashed lines) have a much lower coverage probability than the shape parameters (solid lines), particularly for small sample sizes less than $n = 200$.

   In general, the scale parameters appear to be more difficult to estimate than the shape parameters.

2. As the sample size increases, the coverage probability for the shape parameters and scale parameters approaches the nominal level, 95%.

   This suggests that the sampling distribution of the MLE is converging in distribution to a multivariate normal distribution with mean $\boldsymbol{\theta}$ and variance-covariance given by the inverse of the FIM, consistent with the asymptotic theory.

Scenario: Coverage probability vs. sample size and masking probability without right-censoring

We want to get a larger picture of how the coverage probability depends on the sample size n and masking probability p. We fix the right-censoring time to $\tau = \infty$ and vary the sample size from $n = 30$ to $n = 800$ and vary the masking probability from $p = 0$ (no masking) to $p = 0.4$ and then compute the coverage probability for each combination of sample size n and masking probability p.

The results of this analysis are summarized by Figure 9. Here are some key observations:

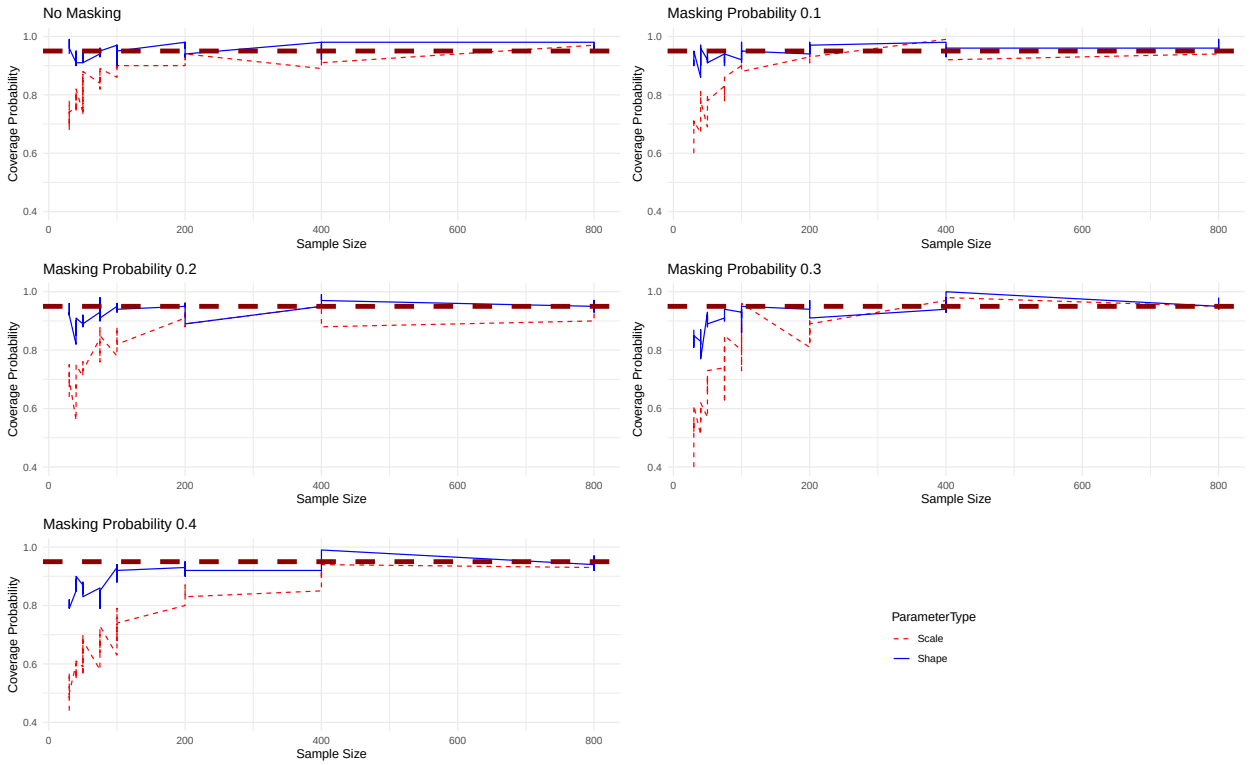

Figure 9: Coverage probability vs. sample size

1. For sample sizes $n \leq 100$, the coverage probability for the shape parameters is close to the nominal level, 95%, only for small masking probabilities. However, as the sample size increases, the coverage probability for the shape parameters quickly approaches the nominal level, 95%, for all masking probabilities reported here.

2. For the scale parameters, the coverage probability is too low for all sample sizes $n < 200$ for all masking probabilities reported here. For small sample sizes, the confience intervals particularly for the scale parameters, should probably be taken with a grain of salt.

In Section 5, we explore an alternative way to construct confidence intervals using the bootstrap method, which is generally a more accurate way to compute confidence intervals. Unlike the inverse of the observed FIM, it does not assume that the sampling distribution of the MLE is asymptotically normal, and so it is more robust to violations of this assumption.

7 Conclusion

We have developed a likelihood model for series systems with latent components and right-censoring. We have provided evidence that, as long as certain regularity conditions are met, the MLE is asymptotically unbiased and consistent.

References

Please see below for a full list of references.

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

[2] Engelhardt, B. (1992). Introduction to probability and mathematical statistics. 294.
[3] Guo, H., Niu, P. and Szidarovszky, F. (2013). Estimating component reliabilities from incomplete system failure data. *Annual Reliability and Maintainability Symposium (RAMS)* 1–6.

Appendix

Appendix A: Simulation of scenarios using the Monte-carlo data-generating process

```
#################################################################
# Simulation data generating process for specified scenario     #
# (n, p, q), where:                                             #
#    - n is a vector of sample sizes                            #
#     - n is a vector of sample sizes                           #
#     - p is a vector of masking probabilities                  #
#     - q is a vector of right-censoring quantiles of the       #
#       Weibull series distribution.                            #
#################################################################

# here is the R libary we developed for this project
library(wei.series.md.c1.c2.c3)

# for parallel processing
library(parallel)

# you can set a seed for reproducibility of the experimental run
# however, if you use parallel processing, this simple approach will not work.
set.seed(1234)

################################################
# Here is an example of how to run a scenario  #
################################################

# set the simulation name to be used in the file names
sim.name <- "sim-2"
# set the sample sizes
ns <- c(30, 40, 50, 75, 100, 200, 400, 800)
# set the masking probabilities
ps <- seq(0, 0,1, 0.2, 0.3, 0.4)
# set the right-censoring quantiles
qs <- c(0.5, 0.6, 0.7, 0.8, 0.9, 0.95)
# set the number of replicates
R <- 100
# set the number of CPU cores to use
ncores <- 4

# true parameter values
theta <- c(shape1 = 1.2576, scale1 = 994.3661,
           shape2 = 1.1635, scale2 = 908.9458,
           shape3 = 1.1308, scale3 = 840.1141,
           shape4 = 1.1802, scale4 = 940.1141,
           shape5 = 1.3311, scale5 = 836.1123)
```

```r
shapes <- theta[seq(1, length(theta), 2)]
scales <- theta[seq(2, length(theta), 2)]

# helps the MLE optimization routine converge more quickly and reliably
# by scaling the parameters to be of similar magnitude
parscale <- c(1, 1000, 1, 1000, 1, 1000, 1, 1000, 1, 1000)

sim.run <- function(sim.name, n, p, q, R = 1000) {
    mles <- list()
    problems <- list()

    tau <- wei.series.md.c1.c2.c3::qwei_series(
        p = q, scales = scales, shapes = shapes)

    cat("n =", n, ", p =", p, ", q = ", q, ", tau = ", tau, "\n")

    for (r in 1:R) {
        result <- tryCatch({

            df <- wei.series.md.c1.c2.c3::generate_guo_weibull_table_2_data(
                shapes = shapes,
                scales = scales,
                n = n,
                p = p,
                tau = tau)

            sol <- wei.series.md.c1.c2.c3::mle_nelder_wei_series_md_c1_c2_c3(
                df = df,
                theta0 = theta,
                reltol = 1e-7,
                parscale = parscale,
                maxit = 2000L)
            mles <- append(mles, list(sol))

            if (r %% 10 == 0) {
                cat("r = ", r, ": ", sol$par, "\n")
            }

        }, error = function(e) {
            cat("Error at iteration", r, ":")
            print(e)
            problems <- append(problems, list(list(
                error = e, n = n, p = p, q = q, tau = tau, df = df)))
        })
    }

    if (length(mles) != 0) {
        saveRDS(list(n = n, p = p, q = q, tau = tau, mles = mles),
            file = paste0("./results/", sim.name, "/results_", n, "_", p, "_", q, ".rds"))
    }

    if (length(problems) != 0) {
        saveRDS(list(n = n, p = p, q = q, tau = tau, problems = problems),
```

```
            file = paste0("./problems/", sim.name, "/problems_", n, "_", p, "_", q, ".rds"))
    }
}

params <- expand.grid(n = ns, p = ps, q = qs)
result <- mclapply(
    1:nrow(params),
    function(i) sim.run(sim.name, params$n[i], params$p[i], params$q[i], R),
    mc.cores = ncores)
```

Appendix B: Simulation of scenarios using the Bootstrap method

```
################################################################
# in this scenario, we want to see how we can use the bootstrap
# method to estimate the confidence intervals more precisely (better calibration
# of confidence intervals) for small sample sizes.
# we'll use it to construct a 95% confidence interval for the estimator. we'll
# compare this result to the asymptotic theory confidence interval.
# finally, we'll generate CIs by each method, asymptotic (inverse FIM) and
# bootstrap (cov), and compare the coverage probabilities.
################################################################

library(boot)
library(parallel)
library(wei.series.md.c1.c2.c3)

theta <- c(shape1 = 1.2576, scale1 = 994.3661,
           shape2 = 1.1635, scale2 = 908.9458,
           shape3 = 1.1308, scale3 = 840.1141,
           shape4 = 1.1802, scale4 = 940.1141,
           shape5 = 1.3311, scale5 = 836.1123)

shapes <- theta[seq(1, length(theta), 2)]
scales <- theta[seq(2, length(theta), 2)]

# number of CPU cores to use in bootstrap for parallel processing
ncores <- 4

# helps the MLE optimization routine converge more quickly and reliably
parscale <- c(1, 1000, 1, 1000, 1, 1000, 1, 1000, 1, 1000)

#set.seed(134849131)

# sample sizes
ns <- c(30, 50, 100, 200, 400)
# masking probabilities, no masking and 21.5% masking
ps <- c(0, 0.215)
# quantiles of weibull series distribution, no right-censoring and 25% right-censoring
qs <- c(1, 0.75)

sim.name <- "sim-1-boot"
```

```r
sim.boot.run <- function(n, p, q, R = 1000) {

    problems <- list()

    tau <- wei.series.md.c1.c2.c3::qwei_series(
        p = q, scales = scales, shapes = shapes)

    cat("n =", n, ", p =", p, ", q = ", q, ", tau = ", tau, "\n")

    result <- tryCatch({
        df <- wei.series.md.c1.c2.c3::generate_guo_weibull_table_2_data(
            shapes = shapes,
            scales = scales,
            n = n,
            p = p,
            tau = tau)

        sol <- wei.series.md.c1.c2.c3::mle_nelder_wei_series_md_c1_c2_c3(
            df = df,
            theta0 = theta,
            reltol = 1e-7,
            parscale = parscale,
            maxit = 2000L)

        cat("mle: ", sol$par, "\n")

        sol.boot <- boot(df, function(df, i) {
            sol <- wei.series.md.c1.c2.c3::mle_nelder_wei_series_md_c1_c2_c3(
                df = df[i, ],
                theta0 = sol$par,
                reltol = 1e-7,
                parscale = parscale,
                maxit = 1000L)
            cat("boot: ", sol$par, "\n")
            sol$par
        }, ncpus = ncores, R = R)

        saveRDS(list(n = n, p = p, q = q, tau = tau, mle = sol, mle.boot = sol.boot),
            file = paste0("./results/", sim.name, "/results_", n, "_", p, "_", q, ".rds"))

        }, error = function(e) {
            print(e)
            problems <- append(problems, list(list(
                error = e, n = n, p = p, q = q, tau = tau, df = df)))
        })

    if (length(problems) != 0) {
        saveRDS(list(n = n, p = p, q = q, tau = tau, problems = problems),
                file = paste0("./problems/", sim.name, "/problems_", n, "_", p, "_", q, ".rds"))
    }
}

params <- expand.grid(n = ns, p = ps, q = qs)
```

```r
result <- mclapply(
    1:nrow(params),
    function(i) sim.boot.run(sim.name, params$n[i], params$p[i], params$q[i]),
    mc.cores = ncores)
```