Estimating the reliability of components in a series systems from masked failure data

Alex Towell

Abstract

We estimate the reliability of series systems from masked failure data consisting of right-censored system lifetimes and masked component cause of failures. Under a minimal set of conditions that permit us to ignore how the component cause of failures are masked, we estimate the sampling distribution of the MLE in a variety of situations. We find that as long as the masking does not always include a particular component as a potential cause of failure whenever another component is also a potential cause, the MLE is unique and consistent. However, this is often not a realistic situation in industrial settings. For example, a system may be repaired by replacing an entire circuit board consisting of multiple components, in which case whenever one of the components on that circuit board fails, all of the components on the circuit board are potential component causes of failure. In this case, we have a non-unique MLE, which provides less information about the reliability of the series system and its components.

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.

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

2 Statistical 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.

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 $f(x_1, \ldots, x_m)$.

2.1 Masked data

The object of interest is the (unknown) parameter value $\boldsymbol{\theta}$. To estimate this $\boldsymbol{\theta}$, we need *data*. Generally, we encounter three types of life data in this context: exact lifetimes, interval lifetimes (in which a lifetime is known to have occurred between two points in time), and right censored lifetimes (in which we only know that a lifetime is greater than some point in time, e.g., the experiment was terminated or suspended before the system failed).

In this paper, we limit our focus to right censored lifetimes. 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. Whether it is right-censored is a random variable defined as

$$\delta_i = 1_{T_i < \tau_i} \tag{2.5}$$

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. The right-censored time of the i^{th} system, S_i, is given by

$$S_i = \min\{\tau_i, T_i\}. \tag{2.6}$$

In addition to the above, we also have masked data about the component cause of failure in the form of candidate sets. A candidate set consists of some subset of component indices that plausibly contain the index 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.

Let's define the candidate set corresponding to the i^{th} system as c_i. Since the data generating process for candidate sets may be subject to chance variations, we model it as a random set $\mathcal{C}_i$. We may also represent

the candidate set $\mathcal{C}_i$ as a random Boolean vector $X_{i1}, \ldots, X_{im}$ where $X_{ij} = 1$ denotes that the j^{th} component index is in the candidate set for the i^{th} system.

When we refer to the sequence $a_1, \ldots, a_k$, we use the notation $\{a_i\}_{i \leq k}$. If it is clear from the context, we may also use $\{a_i\}$. Our random sample of masked data is give by

$$\{(S_i, \delta_i, \mathcal{C}_i)\}_{i \leq n}, \tag{2.7}$$

where $\mathcal{C}_i$ is the random candidate set, S_i is the right-censored lifetime, and δ_i is the indicator for right-censoring for the i^{th} series system. The masked data generation process for the sample in (2.7) is illustrated by Figure 1.

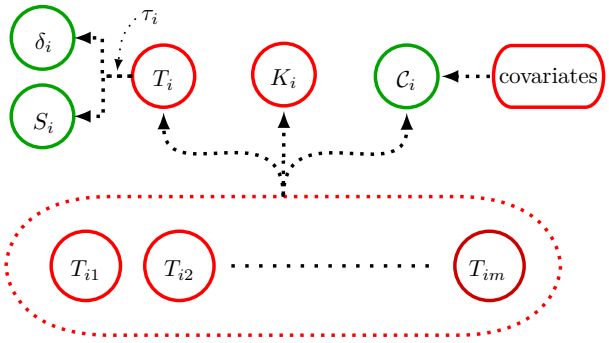


Figure 1: This figure showcases a dependency graph of the generative model for $(S_i, \delta_i, \mathcal{C}_i)$. The elements in green are observed in the sample, while the elements in red are unobserved (latent). We see that $\mathcal{C}_i$ is related to both the unobserved component lifetimes $T_{i1}, \ldots, T_{im}$ 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 $\mathcal{C}_i$.

An example of masked data 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 3 components (also known as an 3-out-of-3 system).

Table 1: Right-censored lifetime data with masked component cause of failure. The 3^{rd} system is right-censored ($\delta_3 = 1$) with a right-censoring time of $s_3 = \tau_3 = 5.4$. The candidate set for the 3^{rd} system is the empty set ($c_3 = \emptyset$), which indicates that the system has no failed components (therefore there cannot be a plausible subset of the components that contains the failed component).

System	Right-censored time (S_i)	Right censored (δ_i)	Candidate set (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	$\emptyset$

3 Series system lifetime

The previous section served as an introduction to key concepts in reliability theory: the reliability function, the probability density function (pdf), and the hazard function. Here, we dive deeper into these concepts and provide mathematical derivations for the metrics specific to series systems. These are not purely theoretical

explorations; these derivations are crucial in applying Maximum Likelihood Estimation to our masked data model.

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*

$$R(t;\boldsymbol{\theta}) = \prod_{j=1}^{m} R_j(t;\boldsymbol{\theta_j}). \tag{3.1}$$

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*

$$f(t;\boldsymbol{\theta}) = \sum_{j=1}^{m} f_j(t;\boldsymbol{\theta_j}) \prod_{\substack{k=1\\k\neq j}}^{m} R_k(t;\boldsymbol{\theta_j}). \tag{3.2}$$

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*

$$h(t;\boldsymbol{\theta}) = \sum_{j=1}^{m} h_j(t;\boldsymbol{\theta_j}). \tag{3.3}$$

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

$$\begin{aligned} f(t;\boldsymbol{\theta}) &= h(t;\boldsymbol{\theta}) R(t;\boldsymbol{\theta}) \\ &= \left\{ \sum_{j=1}^{m} h_j(t;\boldsymbol{\theta_j}) \right\} \left\{ \prod_{j=1}^{m} R_j(t;\boldsymbol{\theta_j}) \right\}, \end{aligned} \tag{3.4}$$

which we sometimes find to be a more convenient form than Equation (3.2).

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.

3.1 Component cause of failure

Whenver a series system fails, precisely one of the components failed to cause the system failure. We model the component cause of the series system failure as a random variable.

Definition 3. *The component cause of failure of the i^{th} 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*

$$f_{K_i,T_i}(j, t; \boldsymbol{\theta}) = h_j(t; \boldsymbol{\theta_j}) R_{T_i}(t; \boldsymbol{\theta}), \tag{3.5}$$

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. □

4 Likelihood Model for Masked Data

The likelihood is a function that quantifies the plausibility of observed data given specific parameter values, denoted here as $\boldsymbol{\theta}$. In our model, we will consider two types of masked data: *right-censored* system lifetime data, which obscures the true system lifetime, and *masking of the component cause of failure*, where the exact cause of failure is unknown but the system lifetime is precisely known. For each type of data, we will derive the *likelihood contributions*.

Given an i.i.d. random sample of masked data

$$\{(S_i, \delta_i, \mathcal{C}_i)\}_{i \leq n}$$

parameterized by $\boldsymbol{\theta} \in \boldsymbol{\Omega}$, the joint pdf is given by

$$f\big(\{(s_i, \delta_i, \mathcal{C}_i)\}_{i \leq n}; \boldsymbol{\theta}\big) = \prod_{i=1}^{n} f(s_i, \delta_i, \mathcal{C}_i; \boldsymbol{\theta}).$$

The joint pdf computes the likelihood that the observed data, $\{s_i, \delta_i, c_i\}_{i \leq n}$, occurs. When we fix the data and instead allow the parameter $\boldsymbol{\theta}$ to vary, we obtain what is called the likelihood function,

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

where

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

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

We present the following theorem for the likelihood contribution model. In subsequent subsections, we derive this result for each type of masked data, i.e., right-censored system lifetime data ($\delta_i = 0$), and masking of the component cause of failure ($\delta_i = 1$).

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

$$L_i(\boldsymbol{\theta}) = \begin{cases} R_{T_i}(s_i; \boldsymbol{\theta}) & \text{if } \delta_i = 0 \\ \beta_i R_{T_i}(s_i; \boldsymbol{\theta}) \sum_{j \in c_i} h_j(s_i; \boldsymbol{\theta_j}) & \text{if } \delta_i = 1. \end{cases} \tag{4.1}$$

Next, we derive the likelihood contributions for each type of masked data.

4.1 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 ultimate goal of our analysis is to estimate the parameters $\boldsymbol{\theta}$ that maximize the likelihood of our observed data. To achieve this, we first need to consider the joint distribution of the continuous system 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 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 we know the pdf $f_{T_i}(t_i; \boldsymbol{\theta})$ but we do not know $\Pr_{\boldsymbol{\theta}}\{\mathcal{C}_i = c_i | T_i = t_i\}$, i.e., the data generating process for candidate sets is not known. However, in order for the masked data $\mathcal{C}_i$ to provide information about $\boldsymbol{\theta}$, $\mathcal{C}_i$ must be correlated with the i^{th} system in some useful way, in which case the conditional distribution of $\mathcal{C}_i$ given $T_i = t_i$ may be important even though the object of statistical interest is the series

system rather than the candidate sets, which may be statistically problematic, e.g., $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n$ may not be identically distributed or there may be many important unobserved covariates.

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

$$\begin{aligned} f_{T_i,K_i,\mathcal{C}_i}(t_i, j, c_i; \boldsymbol{\theta}) &= f_{T_i,K_i}(t_i, k; \boldsymbol{\theta}) \Pr_{\boldsymbol{\theta}}\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\} \\ &= h_j(t_i; \boldsymbol{\theta_j}) R_{T_i}(t_i; \boldsymbol{\theta}) \Pr_{\boldsymbol{\theta}}\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\}. \end{aligned} \tag{4.2}$$

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 (4.2),

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

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

$$f_{T_i,\mathcal{C}_i}(t_i, c_i; \boldsymbol{\theta}) = R_{T_i}(t_i; \boldsymbol{\theta}) \sum_{j \in c_i} \left\{ h_j(t_i; \boldsymbol{\theta_j}) \Pr_{\boldsymbol{\theta}}\{\mathcal{C}_i = c_i | T_i = t_i, K_i = j\} \right\}. \tag{4.3}$$

When we try to find an MLE of $\boldsymbol{\theta}$ (see Section 5), 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 (4.3), 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:

$$L_i(\boldsymbol{\theta}) = R_{T_i}(t_i; \boldsymbol{\theta}) \sum_{j \in c_i} h_j(t_i; \boldsymbol{\theta_j}). \tag{4.4}$$

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 (4.4).

4.1.1 Conditional distribution of K_i given T_i and $\mathcal{C}_i$

This subsection is not necessary in our likelihood model, but it derives a useful result for making predictions about the component cause of failure.

Suppose we have jointly observed a candidate set and a series system failure time and we are interested in the probability that a particular component is the cause of the observed system failure.

Theorem 6. *Assuming Conditions 1 and 2, the conditional probability of K_i given $\mathcal{C}_i = c_i$ and $T_i = t_i$ is given by*

$$\Pr\{K_i = j | T_i = t_i, \mathcal{C}_i = c_i\} = \frac{h_j(t_i|\boldsymbol{\theta_j})}{\sum_{l \in c_i} h_l(t_i|\boldsymbol{\theta_l})} 1_{\{j \in c_i\}}. \tag{4.5}$$

Proof. The conditional probability $\Pr\{K_i = j | T_i = t_i, \mathcal{C}_i = c_i\}$ may be written as

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

By Theorem 4, $f_{K_i,T_i}(j, t_i; \boldsymbol{\theta}) = h_j(t_i; \boldsymbol{\theta}) R_{T_i}(t_i; \boldsymbol{\theta})$. We may make this substitution in the above equation and cancel the common factors $R_{T_i}(t_i; \boldsymbol{\theta})$ in the numerator and denominator, yielding

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

Assuming Condition 1, we may rewrite the above as

$$\Pr\{K_i = j | T_i = t_i, \mathcal{C}_i = c_i\} = \frac{\Pr_{\boldsymbol{\theta}}\{\mathcal{C}_i = c_i | K_i = j, T_i = t_i\} h_j(t_i; \boldsymbol{\theta})}{\sum_{l \in c_i} \Pr_{\boldsymbol{\theta}}\{\mathcal{C}_i = c_i | K_i = l, T_i = t_i\} h_j(t_i; \boldsymbol{\theta})}.$$

Assuming Condition 2, we may rewrite the above as

$$\Pr\{K_i = j | T_i = t_i, \mathcal{C}_i = c_i\} = \frac{\Pr_{\boldsymbol{\theta}}\{\mathcal{C}_i = c_i | K_i = j', T_i = t_i\} h_j(t_i; \boldsymbol{\theta_j})}{\Pr_{\boldsymbol{\theta}}\{\mathcal{C}_i = c_i | K_i = j', T_i = t_i\} \sum_{l \in c_i} h_l(t_i; \boldsymbol{\theta_l})},$$

where $j' \in c_i$. Finally, we may cancel the common probability masking factors in the numerator and denominator, obtaining the result

$$\Pr\{K_i = j | T_i = t_i, \mathcal{C}_i = c_i\} = \frac{h_j(t_i; \boldsymbol{\theta_j})}{\sum_{l \in c_i} h_l(t_i; \boldsymbol{\theta_l})}.$$

□

Frequently, we may not have any information at all about the component cause of failure, but we may still want to estimate the probability that a particular component is the cause of a system failure at a particular time. In this case, we may use the following corollary.

Corollary 1. *The probability that the j^{th} component is the cause of system failure given only that we know a system failure occured at time t_i is given by*

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

Proof. If we cannot narrow the component cause of failure down to some subset of the components, then we let c_i contain all m components, $c_i = \{1, \ldots, m\}$, and plug that into Equation 4.5. The result immediately follows. □

4.2 Right-censored data

As described in Section 2.1, 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 7. *The likelihood contribution of a right-censored observation ($\delta_i = 0$) is given by*

$$L_i(\boldsymbol{\theta}) = R_{T_i}(s_i; \boldsymbol{\theta}). \tag{4.6}$$

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 $\boldsymbol{\theta}$.

5 Maximum likelihood estimation

We use maximum likelihood estimation (MLE) to estimate the series system parameter vector $\boldsymbol{\theta}$ given the masked failure data described in Section 2.1. 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 likelhood estimate* (MLE) of $\boldsymbol{\theta}$. That is, $\hat{\boldsymbol{\theta}}$ is a value of $\boldsymbol{\theta}$ that satisfies

$$L(\hat{\boldsymbol{\theta}}) = \max_{\boldsymbol{\theta}\in\boldsymbol{\Omega}} L(\boldsymbol{\theta}). \tag{5.1}$$

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

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ell_i(\boldsymbol{\theta}) \tag{5.2}$$

where

$$\ell_i(\boldsymbol{\theta}) = \begin{cases} \sum_{j=1}^{m} \log R_j(s_i;\boldsymbol{\theta_j}) & \text{if } \delta_i = 0 \\ \sum_{j=1}^{m} \log R_j(s_i;\boldsymbol{\theta_j}) + \log\big(\sum_{j\in c_i} h_j(s_i;\boldsymbol{\theta_j})\big) & \text{if } \delta_i = 1. \end{cases} \tag{5.3}$$

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 (4.1), 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 8.

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

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

□

5.1 Solving the maximum likelihood equations

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

$$\frac{\partial}{\partial \theta_j} \ell(\boldsymbol{\theta}) = 0 \tag{5.4}$$

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 (5.4) exist, each solution that maximizes the likelihood function is a valid MLE.

If there is no closed-form solution to the maximum likelihood equations (5.4), 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,

$$\boldsymbol{\theta}^{(n+1)} = \boldsymbol{\theta}^{(n)} + \alpha^{(n)} \boldsymbol{d}^{(n)}, \tag{5.5}$$

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 (5.5) 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. There are many other methods that are more likely to find a global maximum, such as sometimes moving in a direction that decreases the likelihood of the guess. However, for our simulations, we found that we were always able to find a global maximum using the iterative method described in Equation (5.5).

We do as many iterations in Equation (5.5) 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, $\boldsymbol{\theta}^{(n)} \approx \hat{\boldsymbol{\theta}}$.

Two popular iterative techniques are Newton-Raphson method and gradient ascent, which respectively are defined by letting

$$\boldsymbol{d}^{(n)}_{\text{Newton-Raphson}} = J^{-1}(\boldsymbol{\theta}^{(n)}) \nabla \ell(\boldsymbol{\theta}^{(n)})$$

and

$$\boldsymbol{d}^{(n)}_{\text{gradient ascent}} = \nabla \ell(\boldsymbol{\theta}^{(n)}),$$

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 $\boldsymbol{\theta}^{(n)}$.

Depending upon the nature of the log-likelihood function, one or the other may work better in practice.[2]

5.2 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 (5.4) 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 that is the inverse of the Fisher information matrix, whose (i, j)-th component is given by

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

Since it is frequently problematic taking the expectation to derive $I(\theta)$, we instead use the *observed* information matrix, which is conditioned on the observed data,

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

Since we do not know $\boldsymbol{\theta}$, we estimate $J(\boldsymbol{\theta})$ 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 $\mathcal{N}(\hat{\boldsymbol{\theta}}, J^{-1}(\hat{\boldsymbol{\theta}}))$.

6 Components with Weibull distributed lifetimes

Consider a series system in which the components have Weibull distributed lifetimes. 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

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

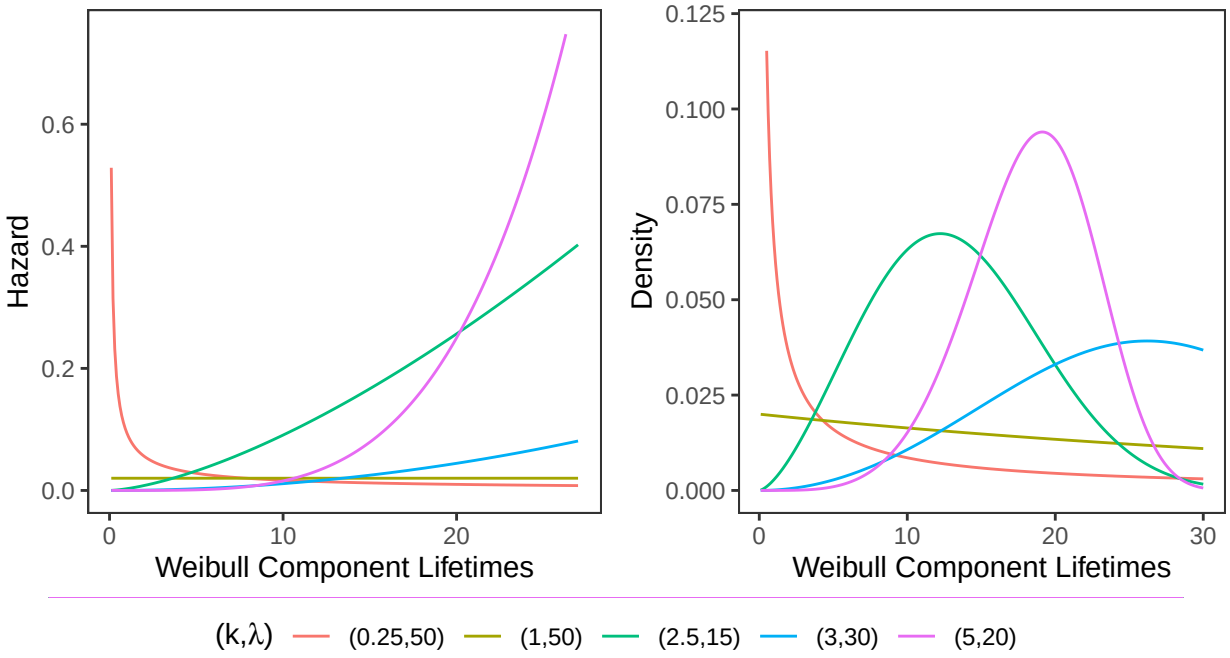
$$f_j(t; \lambda_j, k_j) = \frac{k_j}{\lambda_j}\left(\frac{t}{\lambda_j}\right)^{k_j - 1} \exp\left\{-\left(\frac{t}{\lambda_j}\right)^{k_j}\right\}, \tag{6.2}$$

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

where $t > 0$, $\lambda_j > 0$ is the scale parameter and $k_j > 0$ is the shape parameter.

The shape parameter k may be understood in the following way:

[2] For example, one approach may be more computationally efficient.



Plots of five different components with Weibull distributed lifetimes. Key observations:
(1) Components with a shape < 1 have decreasing hazards, e.g., component 1.
(2) Components with shapes > 1 have increasing hazards, e.g., components 3, 4, and 5.
(3) Components with shape = 1 have constant hazards, e.g., component 2.

Figure 2: Component lifetime plots

- If $k < 1$, then the hazard function decreases with respect to system lifetime, which may occur if defective items fail early and are weeded out.
- If $k > 1$, then the hazard function is increases with respect to time, which may occur as a result of an aging process.
- If $k = 1$, then the failure rate is constant, which means it is exponentially distributed.

See Figure 2 for plots of the hazard and pdf functions of five different Weibull distributed components. We will use these plots to illustrate the different shapes of the hazard and pdf functions. The first component has a shape parameter $k = 0.25$, which is less than 1, and so the hazard function decreases with respect to time. The second component has a shape parameter $k = 1$, and so the hazard function is constant. The third, fourth, and fifth components have shape parameters $k = 2.5$, $k = 3$, and $k = 5$, respectively, and so the hazard functions increase with respect to time.

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.

In Figure 3, we plot the hazard function and the pdf of the Weibull series system with the component lifetime parameters considered earlier,

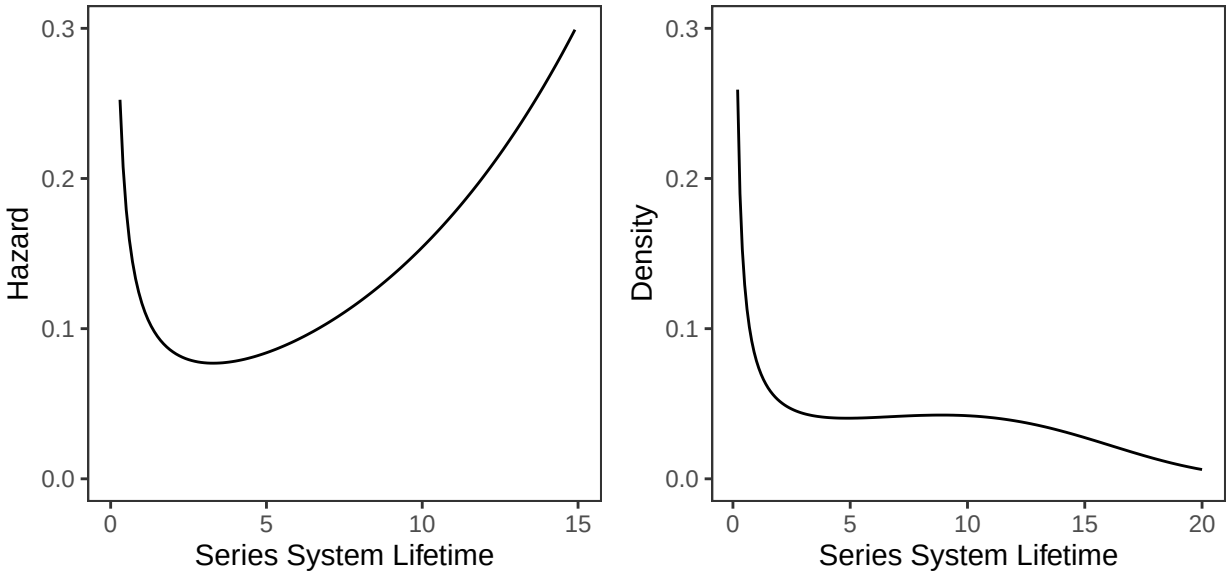
$$\begin{aligned} T_{i1} &\sim \text{WEI}(3, 30) \\ T_{i2} &\sim \text{WEI}(2, 50) \\ T_{i3} &\sim \text{WEI}(0.5, 15) \\ T_{i4} &\sim \text{WEI}(5, 20) \\ T_{i5} &\sim \text{WEI}(0.25, 50) \end{aligned}$$

for the i-th series system where $i = 1, \ldots, n$. By Theorem **??**, the series system has a random lifetime given by

$$T_i = \min\{T_{i1}, \ldots, T_{i5}\}.$$

where $\boldsymbol{\theta} = (k_1, \lambda_1, \ldots, k_5, \lambda_5)$.

The series system, due to being a mixture of Weibull components with different shapes, has both a high infant mortality rate and an aging process, which is reflected in the plot of the hazard function. The hazard



Plots of the hazard function and the pdf of a series system with the previously discussed Weibull components. Key observations:
(1) The hazard is initially large but decreases to some minimum before increasing again, exhibiting both a high infant mortality rate and an aging process. This is a pattern we see in nature (e.g., humans).
(2) The pdf has a rather unusual form, a result of being a combination of Weibull distributions.

Figure 3: System lifetime plots

is initially high then decreases to some minimum before increasing again. This is a pattern we see in nature, e.g., electronic appliances may fail early due to defects, but those that survive the initial period of high failure rate can be expected to last for a long time before finally wearing out due to an aging process.

The pdf of the series system also appears to be multimodal, where the modes correspond to the high infant mortality rate and the aging process.

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. □

The conditional pmf of K_i given T_i is given by

$$f(k;\boldsymbol{\theta}|t) = \frac{\frac{k_k}{\lambda_k}\left(\frac{t}{\lambda_k}\right)^{k_k-1}}{\sum_{j=1}^{m}\frac{k_j}{\lambda_j}\left(\frac{t}{\lambda_j}\right)^{k_j-1}}. \tag{6.7}$$

Proof. By Theorem **??**,

$$f(k;\boldsymbol{\theta}|t) = \frac{h_k(t;\boldsymbol{\theta_k})}{\sum_{j=1}^{m} h_j(t;\boldsymbol{\theta_j})}$$

where $h_1, \ldots, h_m$ are the failure rate functions of the m Weibull component lifetimes. □

The joint pdf of K_i and T_i is given by

$$f(k,t;\boldsymbol{\theta}) = \frac{k_k}{\lambda_j}\left(\frac{t}{\lambda_k}\right)^{k_k-1} \exp\left\{-\sum_{j=1}^{m}\left(\frac{t}{\lambda_j}\right)^{k_j}\right\}. \tag{6.8}$$

Proof. Plugging in the conditional probability and the marginal probability given respectively by Equations (**??**) and (**??**) completes the proof.

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

$$f(k,t;\boldsymbol{\theta}) = h_k(t;\boldsymbol{\theta_k})R_{T_i}(t;\boldsymbol{\theta})$$

where $R_{T_i}(t;\boldsymbol{\theta})$ is given by Equation (6.4) and h_k is the failure rate function of the k^{th} Weibull component. □

6.1 Weibull Likelihood Model for Masked Data

In Section 4, 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 9. *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 2. *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

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

6.2 Maximum likelihood estimation

We may find an MLE by solving the maximum likelihood equation (5.4), 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 5.1. 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 10. *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

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

Proof. The log-likelihood contribution for the i-th observation is given by Equation (6.10):

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

Shape parameters: $\partial \ell_i / \partial k_r$

Proof steps here.

Scale parameters: $\partial \ell_i / \partial \lambda_r$ We want to find the derivative of ℓ_i with respect to λ_r. The derivative of the sum of functions is the sum of the derivatives, so we can differentiate each term in the sum separately.

First, we differentiate

$$-\sum_{j=1}^{m}(t_i/\lambda_j)^{k_j}$$

with respect to λ_r. The derivative of this term with respect to λ_r is zero for all j not equal to r, and for $j = r$, we have $-(t_i/\lambda_r)^{k_r}$. The derivative of this term with respect to λ_r is

$$\frac{k_r}{\lambda_r}\left(\frac{t_i}{\lambda_r}\right)^{k_r}.$$

Second, we differentiate

$$\delta_i \log\left(\sum_{j \in c_i} \frac{k_j}{\lambda_j}\left(\frac{t_i}{\lambda_j}\right)^{k_j - 1}\right)$$

with respect to λ_r. The derivative of the logarithm of a function is the derivative of the function divided by the function itself. The function inside the logarithm is

$$\sum_{j \in c_i} \frac{k_j}{\lambda_j}\left(\frac{t_i}{\lambda_j}\right)^{k_j - 1}.$$

The derivative of this function with respect to λ_r is zero for all j not in c_i or not equal to r, and for $j = r$ in c_i, we have

$$\frac{k_r}{\lambda_r}\left(\frac{t_i}{\lambda_r}\right)^{k_r - 1}.$$

The derivative of this term with respect to λ_r is

$$\left(\frac{k_r}{\lambda_r}\right)^2\left(\frac{t_i}{\lambda_r}\right)^{k_r - 1}.$$

So, the derivative of the second term is

$$\delta_i \frac{\left(\frac{k_r}{\lambda_r}\right)^2\left(\frac{t_i}{\lambda_r}\right)^{k_r - 1}}{\sum_{j \in c_i} \frac{k_j}{\lambda_j}\left(\frac{t_i}{\lambda_j}\right)^{k_j - 1}}$$

if r in c_i and zero otherwise.

Adding the derivatives of the first and second terms, we get the partial derivative of ℓ_i with respect to λ_r,

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

if $r \in c_i$ and zero otherwise. This is the partial derivative of ℓ_i with respect to λ_r. □

These results follow from taking the partial derivatives of the log-likelihood contribution of the i-th system given by Equation (6.9). The proof is omitted for brevity, but the results have been verified by using a very precise numerical approximation of the gradient and verifying equality with the analytical gradient on different data sets and parameter values.

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

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

7 Simulation study: Weibull series system

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 systems (in the population) are not precisely modeled by any known 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. Then, our task is to use our likelihood model to find the best fit using maximum likelihood estimation.

7.1 Bernoulli candidate set model

In our simulation study, we must generate data that satisfies the masking conditions. In other words, we must generate data that satisfies the conditions described in Section 4.1.

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.

7.2 Identifiability

When estimating the parameters of latent components, we must be careful to ensure that the parameters are identifiable. In other words, we must ensure 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 with m components, and in the sample 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 one component, and then we would have $m - 1$ components to estimate instead. We lagely avoid this problem by using the Bernoulli candidate set model, but sometimes it may still arise by chance.

7.3 Optimization issue: 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.

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. We found that this allowed for convergence to MLEs more quickly and reliably.

7.4 Simulation study design

In order to make the simulation study representative of real-world scenarios, we must choose parameter values that are realistic. We base our parameters on the data from [3], which includes a study of the reliability of a series system with three Weibull components parameterized by $\boldsymbol{\theta}$ with shape parameters given by

$$k_1 = 1.2576, k_2 = 1.1635, k_3 = 1.1308,$$

and scale parameters given by

$$\lambda_1 = 994.3661, \lambda_2 = 908.9458, \lambda_3 = 840.1141.$$

In our Bernoulli candidate set model, we estimated that with around probability $p = 0.215$ for including each non-failed component in the candidate set and with a right-censoring time of $\tau = \infty$, we seem to reproduce their reported data.

Our approach is to extend this to a five component Weibull series system, and then 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.

The true parameter value $\boldsymbol{\theta}$ of the five component Weibull series system has shape parameters given by

$$k_1 = 1.2576, k_2 = 1.1635, k_3 = 1.1308, k_4 = 1.1802, k_5 = 1.3311$$

and scale parameters given by

$$\lambda_1 = 994.3661, \lambda_2 = 908.9458, \lambda_3 = 840.1141, \lambda_4 = 940.1141, \lambda_5 = 836.1123.$$

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.

7.4.1 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.

7.5 Analysis of the MLE

The primary purpose of the simulation study is to anlyze the performance of the MLE under different scenarios. We will compute the bias, variance, MSE, confidence interval widths, and coverage proabilities for confidence intervals with respect to a range of scenarios to quantify the performance of the MLE under different circumstances.

7.5.1 Bias

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,

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

Scenario: 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 4, 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. 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 4 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.

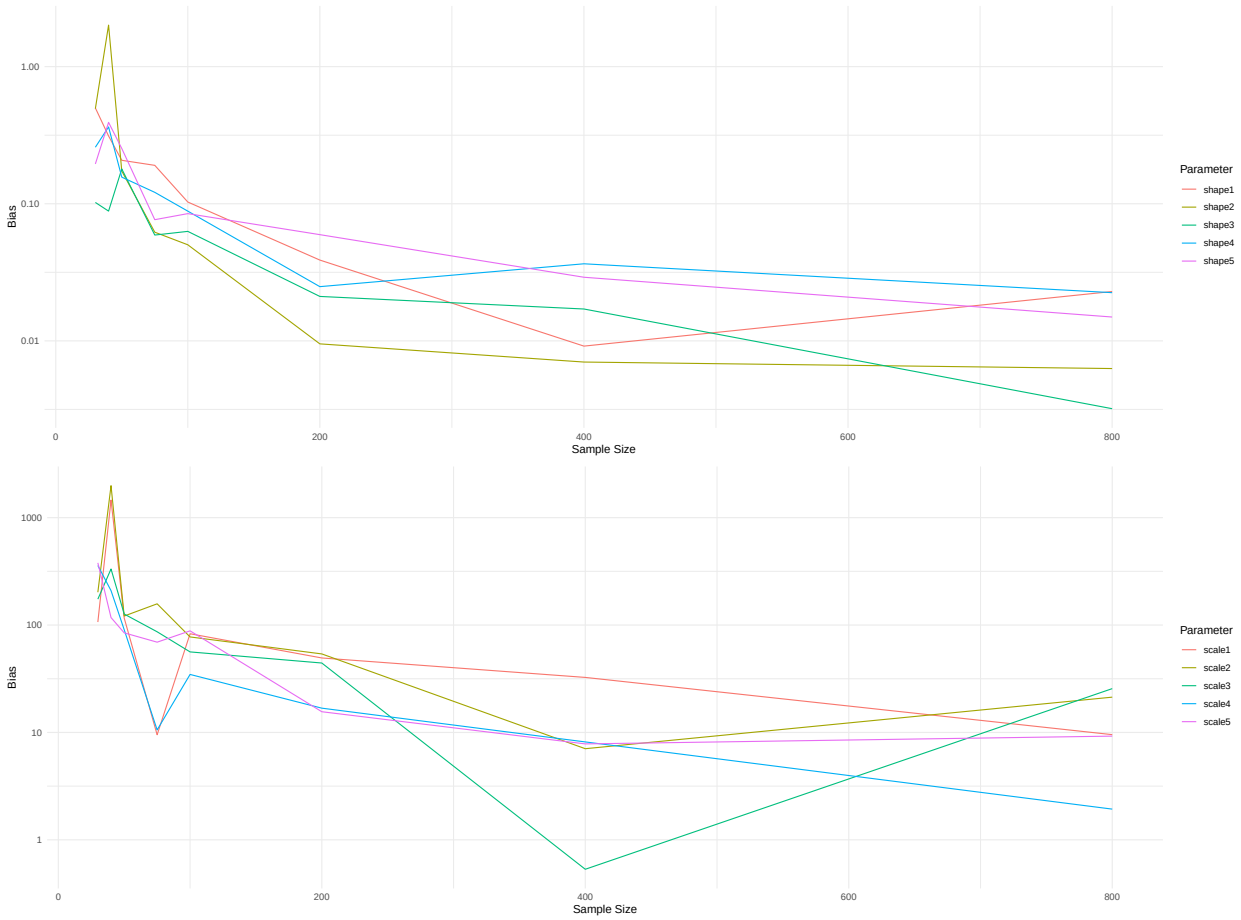


Figure 4: Bias vs. sample size (masking probability 0.2)

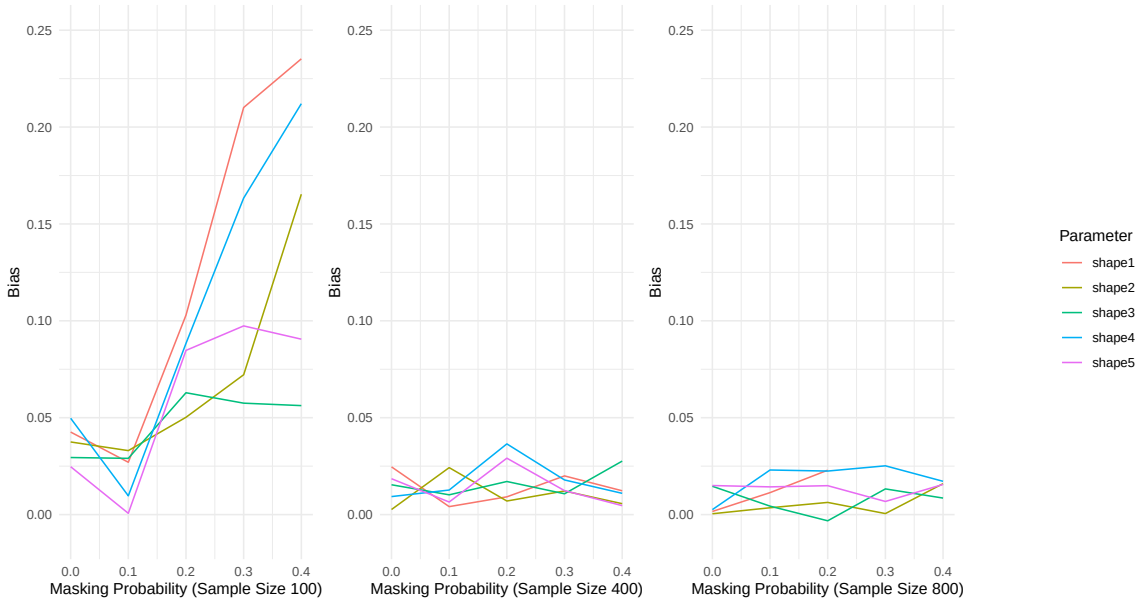


Figure 5: Shape Bias vs. masking probability for sample sizes 100, 400, and 800

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 5. However, for sample sizes of 400 and 800, the bias is relatively small and unaffected by the masking probability.

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 6.

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.

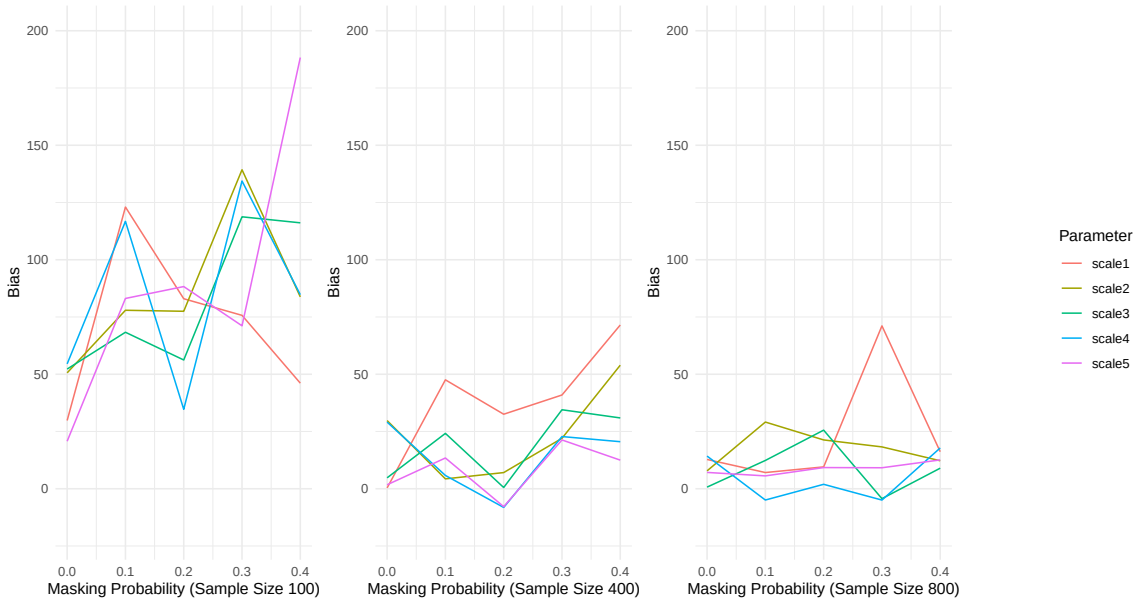


Figure 6: Scale Bias vs. masking probability for sample sizes 100, 400, and 800

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 7. 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 7:

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.

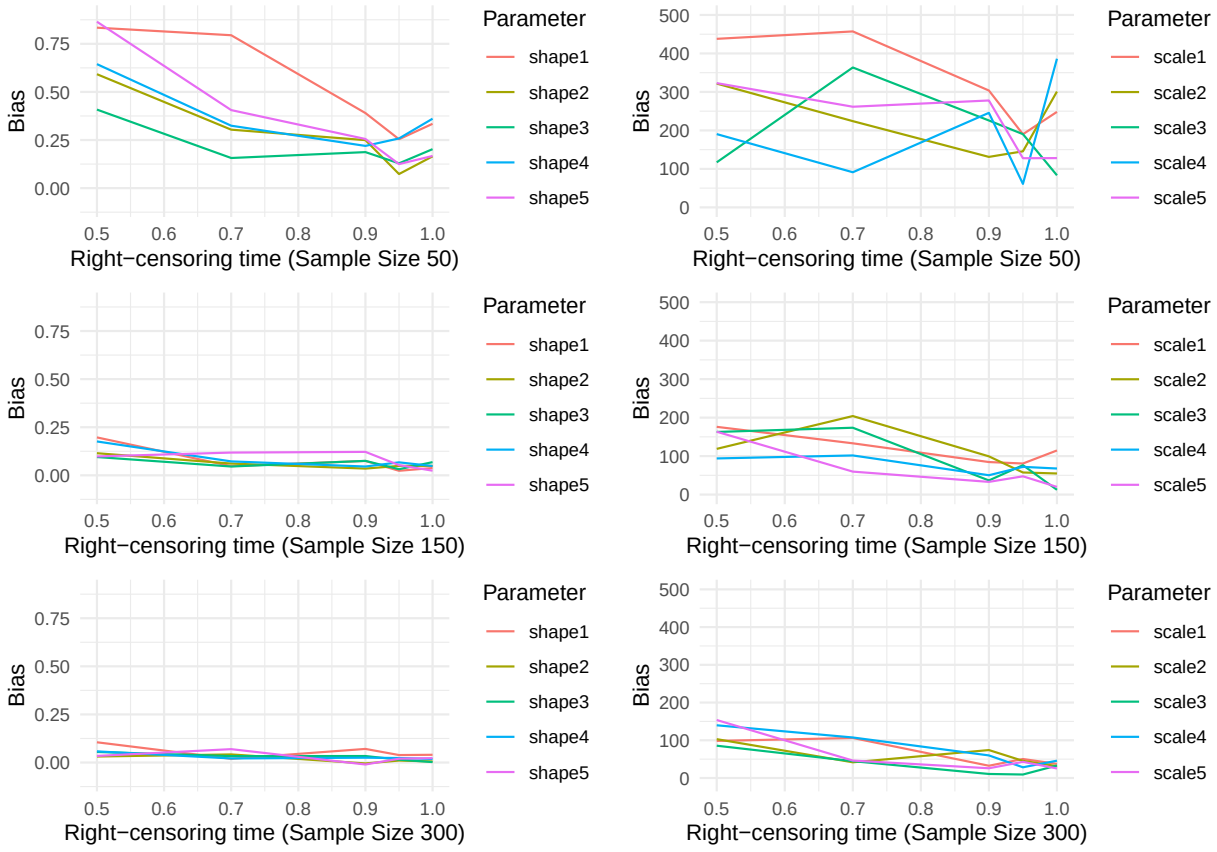


Figure 7: Bias vs. right-censoring time and sample sizes 50, 150, and 300

7.5.2 Coverage probability

We will use the inverse of the observed Fisher information matrix (FIM), which was defined in Section 5.2, to construct 95%-confidence intervals for $\boldsymbol{\theta}$ to compute the coverage probability of the MLE for each scenario. A confidence interval is said to be *well-calibrated* if the coverage probability is close to the nominal level, 95%.

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 (say, sample size and masking probability), we generate $R = 100$ data sets.
2. We find an MLE for each of R data sets.
3. We compute the 95%-confidence interval for each MLE using the inverse of the observed FIM.
4. We compute the coverage probability by calculating the proportion of times the true parameter values were contained in computed the 95%-confidence interval.

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.

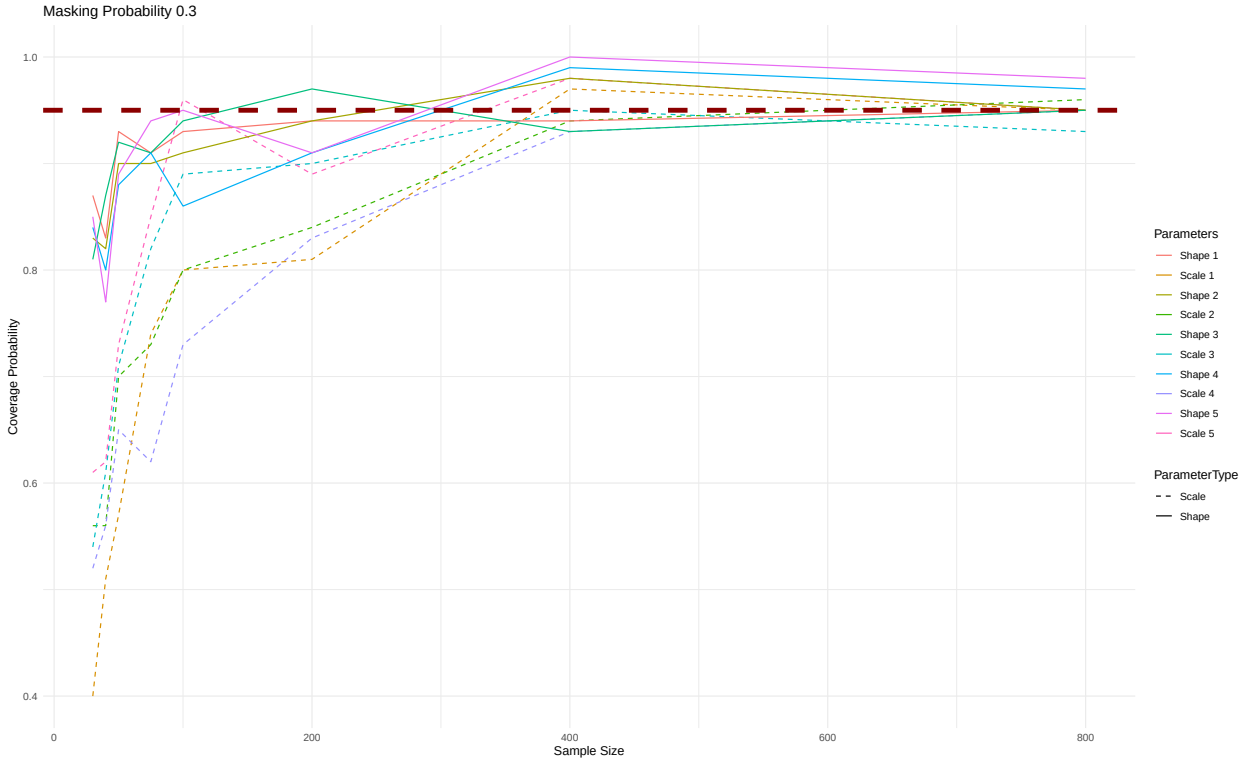


Figure 8: Coverage probability vs. sample size for masking probability 0.3

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.

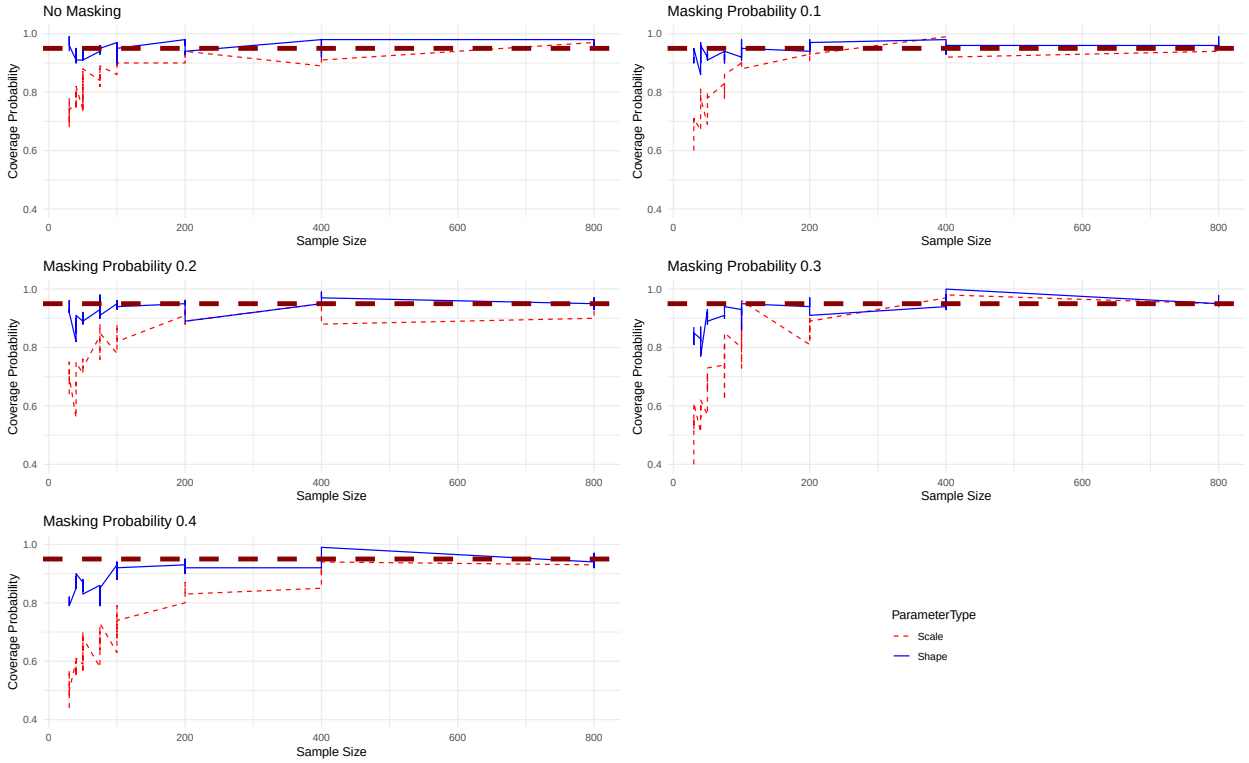


Figure 9: Coverage probability vs. sample size

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

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 7.6, 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.

7.5.3 Confidence interval width

We will use the inverse of the observed Fisher information matrix (FIM), which was defined in Section 5.2, to construct 95%-confidence intervals for $\boldsymbol{\theta}$ to compute the confidence interval width of the MLE for each scenario. The confidence interval width is defined as the difference between the upper and lower bounds of the confidence interval.

If the confidence intervals are well-calibrated, 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 intervals are not well-calibrated, then the confidence interval width is not particularly informative. See Section **??** for a discussion of well-calibrated confidence intervals, which we assume here.

Some key observations:

1. For the shape parameters, the confidence interval widths are small, but the true parameter values are also small. For the scale parameters, the confidence interval widths are large, but the true parameter values are also large.

2. For small sample sizes for every masking probability, the confidence interval widths are extremely variable, and since the coverage probability reported earlier showed they were not well-calibrated, we should not be confident in the MLE for small sample sizes. However, for large sample sizes, the confidence interval widths are relatively stable, and since the coverage probability reported earlier showed they were well-calibrated, we can be confident in the MLE for large sample sizes.

3. The confidence interval widths decrease at a decreasing rate as the sample size increases. This is expected since the inverse of the observed FIM scales as $1/n$, and so the confidence interval widths should scale as $1/\sqrt{n}$.

   On the one hand, this means that as the sample size increases, we see diminishing returns in the precision of the MLE, but on the other hand, the coverage probability is converging to the nominal level, 95%, as the sample size increases, so we can at least gain confidence that the true parameter value is more likely to be within the reported confidence interval at the reported confidence level.

4. The confidence interval widths are somewhat sensitive to the masking probability. As the masking probability increases, the confidence interval widths increase.

For the visualization code for the confidence widths, see Appendix C.

7.5.4 Variance

We estimate the precision of $\hat{\boldsymbol{\theta}}$ with the variance-covariance matrix,

$$\operatorname{Var}(\hat{\boldsymbol{\theta}}) = E\big((\hat{\boldsymbol{\theta}} - E(\hat{\boldsymbol{\theta}}))(\hat{\boldsymbol{\theta}} - E(\hat{\boldsymbol{\theta}}))'\big),$$

which is a p-by-p matrix, where p is the number of parameters, and again the the expectation is taken with respect to the empirical sampling distribution. We will only report the diagonal elements of the variance-covariance matrix, which are the variances of the individual components of $\hat{\boldsymbol{\theta}}$.

> Dr. Agustin: I didn't fill out variance and MSE sections. I have the data, and most of the code for visualization. Would you like for me to include any of these? I'm not sure about how large this paper should be.

7.5.5 Mean squared error

The mean squared error, MSE, is a measure of estimator error that incorporates both the bias and the variance, and is defined as

$$\operatorname{MSE}(\hat{\boldsymbol{\theta}}) = E\big((\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})'\big) = \operatorname{Var}(\hat{\boldsymbol{\theta}}) + \operatorname{b}(\hat{\boldsymbol{\theta}})\,\operatorname{b}(\hat{\boldsymbol{\theta}})',$$

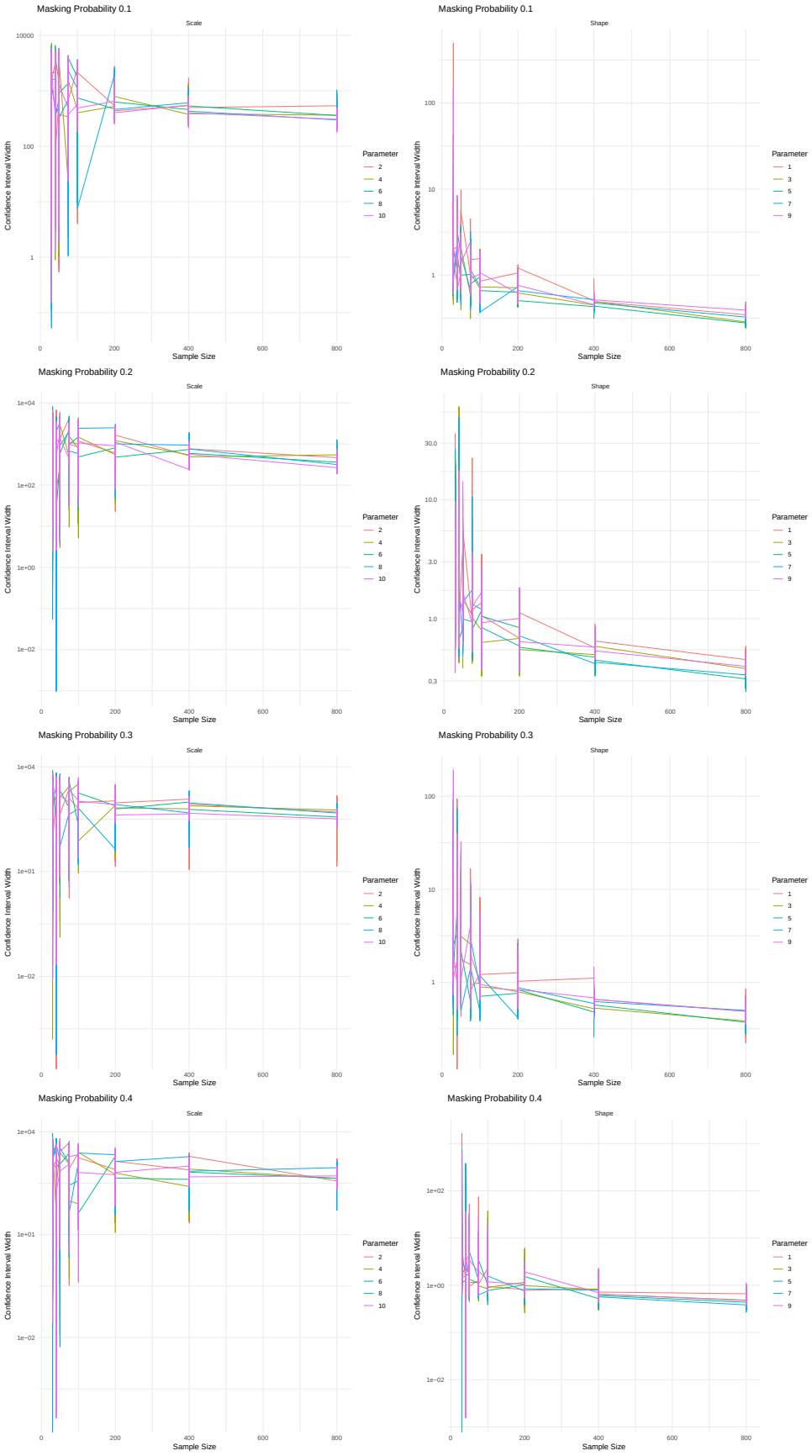


Figure 10: Confidence interval widths vs. sample size

which is a p-by-p matrix, where p is the number of parameters, and again the expectation is with respect to the empirical sampling distribution. We will only report the diagoanl elements of the MSE matrix, which are the mean squared errors of the individual components of $\hat{\boldsymbol{\theta}}$. Assuming the regularity conditions for the MLE are met, the MSE converges in probability to the variance, but for finite n, the MSE is a more informative measure of estimator error than the variance alone.

7.6 Bootstrapping the sampling distribution of the MLE

In a real-world scenario, we would not know how to generate the data from the underlying data generating process, and so we would not be able to compute the empirical sampling distribution of the MLE.

However, we can use the bootstrap method. The bootstrap method is a general method for estimating the sampling distribution of a statistic, in our case the MLE. The most common form of the bootstrap is the non-parametric bootstrap. In the non-parametric bootstrap, the random data is created by resampling with replacement from the original data. Since we do not know (nor attempt to model) the distribution of candidate sets, this non-parametric form is ideal, since we can simply resample from the original data to approximate its empirical sampling distribution of the MLE.

We use the Weibull series data in [3] to illustrate the bootstrap method. It's a real-world data set. We can estimate the sampling distribution of the MLE using the bootstrap method, and compare these results to the asymptotic theory. We expect that since the sample size is relatively small, the bootstrap method will give us a better estimate of the sampling distribution of the MLE than the asymptotic theory.

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 $\hat{\boldsymbol{\theta}}^{(1)}, \ldots, \hat{\boldsymbol{\theta}}^{(B)}$.

We could then compute the bias, variance, MSE, and coverage probability of the bootstrap replicates, and compare these results to the asymptotic theory.

Earlier, we observed the inverse FIM provided for to small of a variance of the component estimates for the MLE. So, we are going to use the bootstrap method to provide better estimates.

We should see that the variance of the bootstrap replicates is larger than the variance of the asymptotic theory, which should mitigate the earlier problem of the inverse FIM providing too small of a variance of the component estimates and thus resulting in poorly calibrated confidence intervals.

This is particularly important for small sample sizes before the asymptotic theory kicks in.

In Figure 11, we show the variance of the shape paraemetes for the bootstrap replicates (blue) and the asymptotic theory (green) for different sample sizes and differnet masking conditions. For small sample sizes, the variance of the bootstrap replicates is larger than the variance of the asymptotic theory, but they both converge to the same value as the sample size increases.

In Figure 12, we show the variance of the scale parameters. Again, we see the same pattern emerge, where the variance of the bootstrap replicates is larger than the variance of the asymptotic theory for small sample sizes, but they both converge to the same value as the sample size increases.

These corrections to the variance estimates should result in better calibrated confidence intervals for small sample sizes.

8 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.

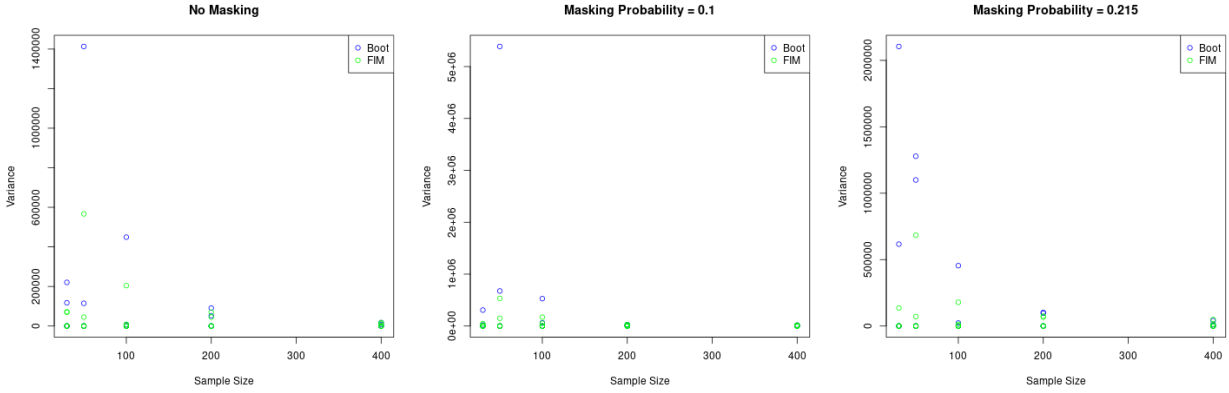


Figure 11: Shape Parameters

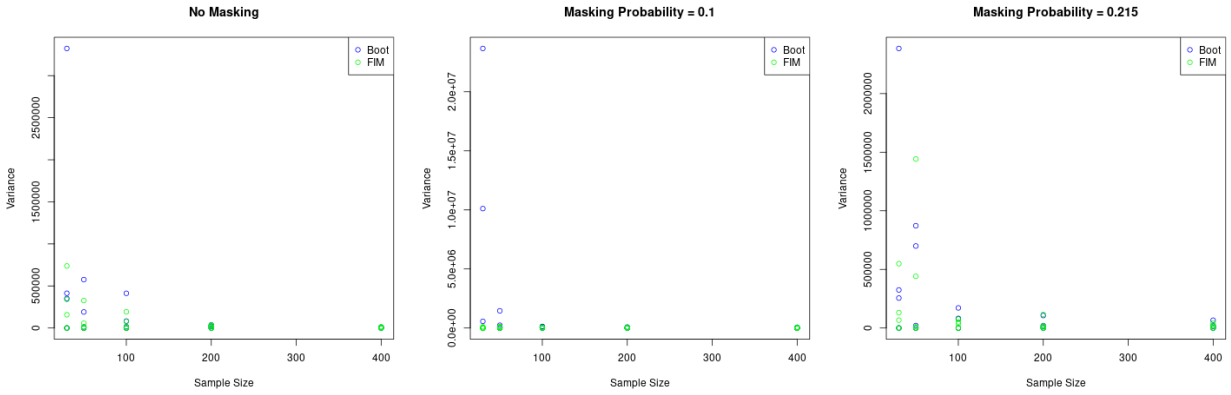


Figure 12: Scale Parameters

[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)

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

```r
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"

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)
result <- mclapply(
    1:nrow(params),
```

```
    function(i) sim.boot.run(sim.name, params$n[i], params$p[i], params$q[i]),
    mc.cores = ncores)
```

Appendix C: Confidence width visualizations

```
#######################################################################
# we visualize CI width as a function of                              #
# masking component cause of failure p from p = 0 to p = 0.4          #
# and sample size n. we have R = 100 replicates for each p and n.     #
#######################################################################
library(gridExtra)  # for arranging plots
library(ggplot2)
library(tidyverse)
library(dplyr)
library(algebraic.mle)
library(md.tools)
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)]
sim2_n <- c(30, 40, 50, 75, 100, 200, 400, 800)

##############################
# generate CI width function #
##############################
gen_ci_width <- function(sim_p_mles, sim_p) {

    ci_widths <- list()

    for (i in 1:length(sim_p)) {

        ci_widths[[i]] <- matrix(nrow = length(sim_p_mles[[i]]), ncol = length(theta))

        for (j in 1:length(sim_p_mles[[i]])) {
            theta.hat <- algebraic.mle::mle_numerical(sim_p_mles[[i]][[j]])
            # this is a m x 2 matrix, where m is the number of parameters
            ci <- confint(theta.hat, use_t_dist = FALSE)

            if (any(is.na(ci))) {
                next
            }

            ci_widths[[i]][j,] <- ci[,2] - ci[,1]
        }
    }
```

```
    ci_widths
}

###################################
# generate CI width plot function #
###################################
gen_ci_width_plot <- function(ci_widths, sim_n, p) {

    ci_widths_df <- do.call(rbind, lapply(seq_along(ci_widths), function(i) {
    data.frame(SampleSize = rep(sim_n[i], nrow(ci_widths[[i]])),
                CI_Width = as.vector(ci_widths[[i]]),
                Parameter = rep(seq(ncol(ci_widths[[i]])), each = nrow(ci_widths[[i]])),
                Type = rep(c("Shape", "Scale"), each = nrow(ci_widths[[i]]), times = ncol(ci_widths[[i]
    }))

    # Split the data frame based on parameter type (Scale or Shape)
    ci_widths_list <- split(ci_widths_df, ci_widths_df$Type)
    # Generate a plot for each type, using a log y scale
    plots <- lapply(ci_widths_list, function(df) {
        ggplot(df, aes(x = SampleSize, y = CI_Width, color = factor(Parameter))) +
            geom_line() +
            scale_y_log10() +
            labs(title = paste("Masking Probability", p),
                x = "Sample Size", y = "Confidence Interval Width", color = "Parameter") +
            facet_wrap(~ Type, scales = "free") +
            theme_minimal()
    })

    # Arrange the plots in a grid
    grid.arrange(grobs = plots, ncol = 2)
}

############
# p = 0.1 #
############
sim2_30_0.1 <- readRDS("results/sim-2/results_30_0.1_1.rds")
sim2_40_0.1 <- readRDS("results/sim-2/results_40_0.1_1.rds")
sim2_50_0.1 <- readRDS("results/sim-2/results_50_0.1_1.rds")
sim2_75_0.1 <- readRDS("results/sim-2/results_75_0.1_1.rds")
sim2_100_0.1 <- readRDS("results/sim-2/results_100_0.1_1.rds")
sim2_200_0.1 <- readRDS("results/sim-2/results_200_0.1_1.rds")
sim2_400_0.1 <- readRDS("results/sim-2/results_400_0.1_1.rds")
sim2_800_0.1 <- readRDS("results/sim-2/results_800_0.1_1.rds")

sim2_p_0.1_mles <- list(sim2_30_0.1$mles, sim2_40_0.1$mles, sim2_50_0.1$mles,
    sim2_75_0.1$mles, sim2_100_0.1$mles, sim2_200_0.1$mles, sim2_400_0.1$mles,
    sim2_800_0.1$mles)
ci_width_0.1 <- gen_ci_width(sim2_p_0.1_mles, sim2_n)
plts_p_0.1 <- gen_ci_width_plot(ci_width_0.1, sim2_n, 0.1)

############
# p = 0.2 #
############
```

```r
sim2_30_0.2 <- readRDS("results/sim-2/results_30_0.2_1.rds")
sim2_40_0.2 <- readRDS("results/sim-2/results_40_0.2_1.rds")
sim2_50_0.2 <- readRDS("results/sim-2/results_50_0.2_1.rds")
sim2_75_0.2 <- readRDS("results/sim-2/results_75_0.2_1.rds")
sim2_100_0.2 <- readRDS("results/sim-2/results_100_0.2_1.rds")
sim2_200_0.2 <- readRDS("results/sim-2/results_200_0.2_1.rds")
sim2_400_0.2 <- readRDS("results/sim-2/results_400_0.2_1.rds")
sim2_800_0.2 <- readRDS("results/sim-2/results_800_0.2_1.rds")

sim2_p_0.2_mles <- list(sim2_30_0.2$mles, sim2_40_0.2$mles, sim2_50_0.2$mles,
    sim2_75_0.2$mles, sim2_100_0.2$mles, sim2_200_0.2$mles, sim2_400_0.2$mles,
    sim2_800_0.2$mles)

ci_width_0.2 <- gen_ci_width(sim2_p_0.2_mles, sim2_n)
plts_p_0.2 <- gen_ci_width_plot(ci_width_0.2, sim2_n, 0.2)

############
# p = 0.3 #
############
sim2_30_0.3 <- readRDS("results/sim-2/results_30_0.3_1.rds")
sim2_40_0.3 <- readRDS("results/sim-2/results_40_0.3_1.rds")
sim2_50_0.3 <- readRDS("results/sim-2/results_50_0.3_1.rds")
sim2_75_0.3 <- readRDS("results/sim-2/results_75_0.3_1.rds")
sim2_100_0.3 <- readRDS("results/sim-2/results_100_0.3_1.rds")
sim2_200_0.3 <- readRDS("results/sim-2/results_200_0.3_1.rds")
sim2_400_0.3 <- readRDS("results/sim-2/results_400_0.3_1.rds")
sim2_800_0.3 <- readRDS("results/sim-2/results_800_0.3_1.rds")

sim2_p_0.3_mles <- list(sim2_30_0.3$mles, sim2_40_0.3$mles, sim2_50_0.3$mles,
    sim2_75_0.3$mles, sim2_100_0.3$mles, sim2_200_0.3$mles, sim2_400_0.3$mles,
    sim2_800_0.3$mles)

ci_width_0.3 <- gen_ci_width(sim2_p_0.3_mles, sim2_n)
plts_p_0.3 <- gen_ci_width_plot(ci_width_0.3, sim2_n, 0.3)

############
# p = 0.4 #
############
sim2_30_0.4 <- readRDS("results/sim-2/results_30_0.4_1.rds")
sim2_40_0.4 <- readRDS("results/sim-2/results_40_0.4_1.rds")
sim2_50_0.4 <- readRDS("results/sim-2/results_50_0.4_1.rds")
sim2_75_0.4 <- readRDS("results/sim-2/results_75_0.4_1.rds")
sim2_100_0.4 <- readRDS("results/sim-2/results_100_0.4_1.rds")
sim2_200_0.4 <- readRDS("results/sim-2/results_200_0.4_1.rds")
sim2_400_0.4 <- readRDS("results/sim-2/results_400_0.4_1.rds")
sim2_800_0.4 <- readRDS("results/sim-2/results_800_0.4_1.rds")

sim2_p_0.4_mles <- list(sim2_30_0.4$mles, sim2_40_0.4$mles, sim2_50_0.4$mles,
    sim2_75_0.4$mles, sim2_100_0.4$mles, sim2_200_0.4$mles, sim2_400_0.4$mles,
    sim2_800_0.4$mles)

ci_width_0.4 <- gen_ci_width(sim2_p_0.4_mles, sim2_n)
plts_p_0.4 <- gen_ci_width_plot(ci_width_0.4, sim2_n, 0.4)
```

```
################################################
# plot all together, plts_p_0.1 to plts_p_0.4 #
################################################
ci_plots <- plot_grid(plts_p_0.1, plts_p_0.2, plts_p_0.3, plts_p_0.4, ncol = 1)
ggsave("results/sim-2/ci-widths.pdf", ci_plots)
```