Reliability Estimation in Series Systems

Maximum Likelihood Techniques for Right-Censored and Masked Failure Data

Alex Towell

2023-10-10

Reliability Estimation in Series Systems

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

Alex Towell

Context & Motivation

Reliability in **series systems** is like a chain's strength – determined by its weakest link.

- Essential for system design and maintenance.

Main Goal: Estimate individual component reliability from *failure data*.

Challenges:

- *Masked* component-level failure data.
- *Right-censoring* in system-level failure data.

Our Response:

- Derive techniques to interpret such ambiguous data.
- Aim for precise and accurate reliability estimates for individual components using maximum likelihood estimation (MLE) and bootstrapped (BCa) confidence intervals.

Context & Motivation

Reliability in **series systems** is like a chain's strength – determined by its weakest link.

- Essential for system design and maintenance.

Main Goal: Estimate individual component reliability from *failure data*.

Challenges:

- *Masked* component-level failure data.
- *Right-censoring* in system-level failure data.

Our Response:

- Derive techniques to interpret such ambiguous data.
- Aim for precise and accurate reliability estimates for individual components using maximum likelihood estimation (MLE) and bootstrapped (BCa) confidence intervals.

- **Chain Analogy**: Think of a series system as a chain. Its reliability, just like a chain's strength, is determined by its weakest link or component. When any component fails, the whole system does.
- **Reliability Importance**: Understanding the reliability of each component is essential for the design and maintenance of these systems.
- **Data Challenge**: The data we rely on can come with its own challenges. We sometimes encounter ambiguous data like right-censored information or masked component-level failures, where we don't know precisely which component failed.
- **Aim**: Our goal is to interpret such ambiguous data and provide accurate reliability estimates for each component, which includes providing correctly specified 9595using the BCa method.

Core Contributions

Likelihood Model for **series systems**.

- Accounts for *right-censoring* and *masked component failure*.
- Can easily incorporate additional failure data.

Specifications of Conditions:

- Assumptions about the masking of component failures.
- Simplifies and makes the model more tractable.

Simulation studies:

- Components with *Weibull* lifetimes.
- Evaluate MLE and confidence intervals under different scenarios.

R Library: Methods available on GitHub.

- github.com/queelius/wei.series.md.c1.c2.c3

Core Contributions
Likelihood Model for series systems.
- Accounts for right-censoring and masked component failure.
- Can easily incorporate additional failure data.
Specifications of Conditions:
- Assumptions about the masking of component failures.
- Simplifies and makes the model more tractable.
Simulation studies:
- Components with Weibull lifetimes.
- Evaluate MLE and confidence intervals under different scenarios.
R Library: Methods available on GitHub.
- github.com/queelius/wei.series.md.c1.c2.c3

Our core contributions can be broken down into several parts:

- **Likelihood model**: We've derived a likelihood model for series systems that accounts for the ambiguous data.
- **Explain conditions**: We've clarified the conditions this model assumes about the masking of component failures. These conditions simplify the model and make it more tractable.
- **Validated with simulation study**: We've validated our model with extensive simulations using Weibull distributions to gauge its performance under various scenarios.
- **R Library**: For those interested, we made our methods available in an R Library hosted on GitHub.

Section 1

Series System

Section 1
Series System

Series System

$$x_1 \to x_2 \to x_3 \to x_4 \to x_5$$

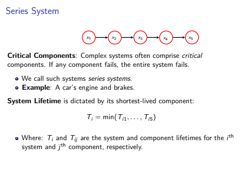
Critical Components: Complex systems often comprise *critical* components. If any component fails, the entire system fails.

- We call such systems *series systems*.
- **Example**: A car's engine and brakes.

System Lifetime is dictated by its shortest-lived component:

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

- Where: T_i and T_{ij} are the system and component lifetimes for the i^{th} system and j^{th} component, respectively.



- **Critical Components**: Many complex systems have components that are essential to their operation.
- **Series System**: If any of these components fail, the entire system fails. We call these series systems.
- **Car**: Think of a car - if the engine or brakes fail, the car can't be operated.
- **Lifetime**: Its lifetime is the lifetime of its shortest-lived component.
- **Notation**: For reference, we show the math notation we'll use throughout the talk.

Reliability Function

Reliability Function represents the probability that a system or component functions beyond a specified time.

- Essential for understanding longevity and dependability.

Series System Reliability: Product of the reliability of its components:

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

- Here, $R_{T_i}(t;\boldsymbol{\theta})$ and $R_j(t;\boldsymbol{\theta_j})$ are the reliability functions for the system i and component j, respectively.

Relevance:

- Forms the foundation for most reliability studies.
- Integral to our likelihood model, e.g., right-censoring events.

Reliability Function

Reliability Function represents the probability that a system or component functions beyond a specified time.

- Essential for understanding longevity and dependability.

Series System Reliability: Product of the reliability of its components:

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

- Here, $R_{T_i}(t;\boldsymbol{\theta})$ and $R_j(t;\boldsymbol{\theta_j})$ are the reliability functions for the system i and component j, respectively.

Relevance:

- Forms the foundation for most reliability studies.
- Integral to our likelihood model, e.g., right-censoring events.

- **Reliability Function** The reliability function tells us the chance a component or system functions past a specific time. It's our key metric for longevity.
- **Product of Component Reliability**: In a series system, the overall reliability is the product of its component reliabilities. So, if even one component has a low reliability, it can impact the whole system.
- **Relevance**: Why does this matter to us? This concept is foundational to our studies, especially when we're handling right-censored data.

Hazard Function: Understanding Risks

Hazard Function: Measures the immediate risk of failure at a given time, assuming survival up to that moment.

- Reveals how the risk of failure evolves over time.
- Guides maintenance schedules and interventions.

Series System Hazard Function is the sum of the hazard functions of its components:

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

- Components' risks are additive.

Hazard Function: Understanding Risks

Hazard Function: Measures the immediate risk of failure at a given time, assuming survival up to that moment.

- Reveals how the risk of failure evolves over time.
- Guides maintenance schedules and interventions.

Series System Hazard Function is the sum of the hazard functions of its components:

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

- Components' risks are additive.

Hazard Function: Let's shift focus to the hazard function. Essentially, it's a way to gauge the immediate risk of failure, especially if it's been functioning up until that point.
Series Hazard Function Lastly, the hazard function for a series system is just the sum of the hazard functions of its components.

Additive: We see that the component risks are additive.

Joint Distribution of Component Failure and System Lifetime

Our likelihood model depends on the **joint distribution** of the system lifetime and the component that caused the failure.

- **Formula**: Product of the failing component's hazard function and the system reliability function:

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

- **Single Point of Failure**: A series system fails due to one component's malfunction.
- **Representation**:
  - K_i: Component causing the i^{th} system's failure.
  - $h_j(t; \boldsymbol{\theta_j})$: Hazard function for the j^{th} component.

Joint Distribution of Component Failure and System Lifetime

Our likelihood model depends on the **joint distribution** of the system lifetime and the component that caused the failure.

- **Formula**: Product of the failing component's hazard function and the system reliability function:

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

- **Single Point of Failure**: A series system fails due to one component's malfunction.
- **Representation**:
  - K_i: Component causing the i^{th} system's failure.
  - $h_j(t; \boldsymbol{\theta_j})$: Hazard function for the j^{th} component.

- **Joint Distribution** In our likelihood model, understanding the joint distribution of a system's lifetime and the component that led to its failure is fundamental.
- **Formula**: It is the product of the failing component's hazard function and the system reliability function.
- **Unique Cause**: Which emphasizes that in a series system, failure can be attributed to a single component's malfunction.
- **Notation**: Here, K_i denotes the component responsible for the failure.

Component Failure & Well-Designed Series Systems

The **marginal probability** of component failure helps predict the cause of failure.

- **Derivation**: Marginalize the joint distribution over the system lifetime:

$$\Pr\{K_i = j\} = E_{\boldsymbol{\theta}}\left[\frac{h_j(T_i; \boldsymbol{\theta_j})}{h_{T_i}(T_i; \boldsymbol{\theta_l})}\right].$$

Well-Designed Series System: Components exhibit comparable chances of causing system failures.

- **Relevance**: Our simulation study employs a (reasonably) well-designed series system.

Component Failure & Well-Designed Series Systems

The **marginal probability** of component failure helps predict the cause of failure.

- **Derivation**: Marginalize the joint distribution over the system lifetime:

$$\Pr\{K_i = j\} = E_{\boldsymbol{\theta}}\left[\frac{h_j(T_i; \boldsymbol{\theta_j})}{h_{T_i}(T_i; \boldsymbol{\theta_l})}\right].$$

Well-Designed Series System: Components exhibit comparable chances of causing system failures.

- **Relevance**: Our simulation study employs a (reasonably) well-designed series system.

- **Marginal**: We can use this joint distribution to calculate the marginal probability of component failure.
- **Expected Value**: When we do so, we find that it is the expected value of the ratio of component and system hazard functions.
- **Well-Designed**: We say that a series system is *well-designed* if each components has a comparable chance of failing.
- **Relevance**: Our simulation study is based on a reasonably well-designed series system.

Section 2

Likelihood Model

Section 2

Likelihood Model

Likelihood Model

Likelihood measures how well our model parameters ($\boldsymbol{\theta}$) explain the data. Each system contributes to the **total likelihood** via its *likelihood contribution*:

$$L(\boldsymbol{\theta}|\text{data}) = \prod_{i=1}^{n} L_i(\boldsymbol{\theta}|\text{data}_i).$$

where $\textbf{data}_i$ is the data for the i^{th} system and L_i is its contribution.

Our model handles the following data: **Right-Censored**: Experiment ends before failure (Event Indicator: $\delta_i = 0$). - Contribution is system reliability: $L_i(\boldsymbol{\theta}) = R_{T_i}(\tau; \boldsymbol{\theta})$. **Masked Failure**: Failure observed, but the failed component is masked by a *candidate set*. More on its contribution later.

System	Right-Censored Lifetime	Event Indicator	Candidate Set
1	1.1	1	$\{1, 2\}$
2	5	0	$\emptyset$

Likelihood Model

Likelihood measures how well our model parameters ($\boldsymbol{\theta}$) explain the data. Each system contributes to the **total likelihood** via its *likelihood contribution*:

$$L(\boldsymbol{\theta}|\text{data}) = \prod_{i=1}^{n} L_i(\boldsymbol{\theta}|\text{data}_i).$$

where $\textbf{data}_i$ is the data for the i^{th} system and L_i is its contribution.

Our model handles the following data: **Right-Censored**: Experiment ends before failure (Event Indicator: $\delta_i = 0$). - Contribution is system reliability: $L_i(\boldsymbol{\theta}) = R_{T_i}(\tau; \boldsymbol{\theta})$. **Masked Failure**: Failure observed, but the failed component is masked by a *candidate set*. More on its contribution later.

System	Right-Censored Lifetime	Event Indicator	Candidate Set
1	1.1	1	$\{1, 2\}$
2	5	0	$\emptyset$

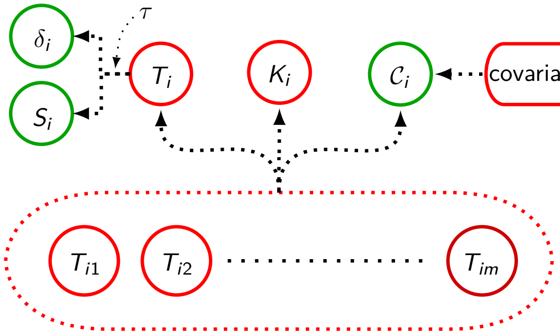
Let's talk about the likelihood model, which is a way of measuring how well our model explains the data.

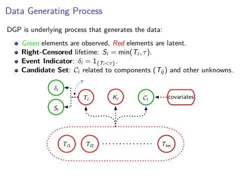
- **Total likelihood** is the product of the likelihood contributions of each system.
- **Contributions**: Our likelihood model deals with right-censoring and masked cause of failure.
- **Right-Censoring** occurs when the experiment ends before the system fails. Its contribution is just the system reliability, since the event indicator tells us if the system was right-censored.
- **Masking** occurs when we observe a failure but we don't know the precise component cause. Instead, we observe a candidate set of components that could have failure. More on this later.
- Here's an example of observed data.
- **System 1**: We see that the system failed at 1.1. We don't know which component failed, but we know it was either component 1 or 2

Data Generating Process

DGP is underlying process that generates the data:

- *Green* elements are observed, *Red* elements are latent.
- **Right-Censored** lifetime: $S_i = \min(T_i, \tau)$.
- **Event Indicator**: $\delta_i = 1_{\{T_i < \tau\}}$.
- **Candidate Set**: $\mathcal{C}_i$ related to components (T_{ij}) and other unknowns.





- **DGP**: Let's discuss the data generating process to motivate our model.
- **Graph**: Here's the graph: green is observed and red is latent.
- **Infer**: We don't get to see the red elements, but we can infer most of them from the green elements.
- **Green**: So, let's focus on the green elements.
- **Right-censoring** time is the minimum of the system lifetime and the right-censoring time.
- **Event** indicator is 1 if the system fails before the right-censoring time, 0 otherwise.
- The candidate sets are related to the component lifetimes and many other factors.
- **Difficult** to model. Seek a simple model that is valid under certain assumptions, which we discuss next.

Likelihood Contribution: Masked Failure Conditions

The candidate sets that **mask** the failed component are assumed to satisfy the following conditions:

Candidate Set Contains Failed Component: The candidate set includes the failed component.

Equal Probabilities Across Candidate Sets: The probability of of the candidate set is constant across different components within it.

Masking Probabilities Independent of Parameters: The masking probabilities when conditioned on the system lifetime and the failed component aren't functions of the system parameter.

- Reasonable conditions in many industrial settings.

Likelihood Contribution: Masked Failure Conditions

The candidate sets that **mask** the failed component are assumed to satisfy the following conditions:

Candidate Set Contains Failed Component: The candidate set includes the failed component.

Equal Probabilities Across Candidate Sets: The probability of of the candidate set is constant across different components within it.

Masking Probabilities Independent of Parameters: The masking probabilities when conditioned on the system lifetime and the failed component aren't functions of the system parameter.

- Reasonable conditions in many industrial settings.

- The right-censoring contribution is straightforward. But the masked failure contribution is a bit more complicated.
- **Masking** occurs when a system fails but the precise failed component is masked by a candidate set.
- **Tractable**: To make problem more tractable, we introduce certain conditions.
- **Condition 1**: The candidate set always includes the failed component.
- **Condition 2**: The probability of the candidate set is constant across different components within it.
- **Condition 3**: The masking probabilities when conditioned on the system lifetime and the failed component aren't functions of the system parameter.
- **Reasonable**: These conditions are often reasonable.

Likelihood Contribution: Derivation for Masked Failures

Take the **joint distribution** of T_i, K_i, and $\mathcal{C}_i$ and marginalize over K_i:

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

Apply **Condition 1** to get a sum over candidate set:

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

Apply **Condition 2** to move probability outside the sum:

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

Apply **Condition 3** to remove the probability's dependence on $\boldsymbol{\theta}$:

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

Result: $L_i(\boldsymbol{\theta}) \propto \sum_{j \in c_i} f_{T_i,K_i}(t_i, j; \boldsymbol{\theta}) = R_{T_i}(t_i; \boldsymbol{\theta}) \sum_{j \in c_i} h_j(t_i; \boldsymbol{\theta_j})$.

Likelihood Contribution: Derivation for Masked Failures
Take the joint distribution of T_i, K_i, and $\mathcal{C}_i$ and marginalize over K_i:
Apply Condition 1 to get a sum over candidate set:
Apply Condition 2 to move probability outside the sum:
Apply Condition 3 to remove the probability's dependence on $\boldsymbol{\theta}$:
Result: $L_i(\boldsymbol{\theta}) \propto \sum_{j \in c_i} f_{T_i,K_i}(t_i, j; \boldsymbol{\theta}) = R_{T_i}(t_i; \boldsymbol{\theta}) \sum_{j \in c_i} h_j(t_i; \boldsymbol{\theta_j})$.

- **Derive**: Here, we derive the likelihood contribution for masked failures.
- **Joint**: To start, we use the joint distribution of the system lifetime, the failed component, and the candidate set.
- **Marginalize**: We marginalize over the failed component, since we don't know which component failed.
- Apply Condition 1 to get a sum over the candidate set instead.
- Apply Condition 2 to move the probability outside the sum.
- Apply condition 3 to remove the probability's dependence on $\boldsymbol{\theta}$.
- The result: the likelihood contribution is proportional to the product of the system reliability and the sum of the component hazards in the masking set.

Methodology: Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE): Maximize the likelihood function:

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

Solution: Numerically solved system of equations for $\hat{\boldsymbol{\theta}}$:

$$\nabla_{\boldsymbol{\theta}} \log L(\hat{\boldsymbol{\theta}}) = \mathbf{0}.$$

Methodology: Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE): Maximize the likelihood function:

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

Solution: Numerically solved system of equations for $\hat{\boldsymbol{\theta}}$:

$$\nabla_{\boldsymbol{\theta}} \log L(\hat{\boldsymbol{\theta}}) = \mathbf{0}.$$

- **MLE**: We use the standard MLE approach.
- **ArgMax**: We find the parameter values that maximize the log-likelihood function.
- **Solution**: Since there is no closed-form solution, we numerically solve it.

Bootstrap Confidence Intervals (CIs)

Confidence Intervals (CI) quantify the uncertainty in our estimate.

Asymptotic Sampling Distribution of MLE is a popular choice for constructing CIs.

- **Challenge**: Asymptotic distribution may not be accurate for small sample sizes.
  - Particularly since we're dealing with right-censoring and masking.

Bootstrapped CIs: Resample data and obtain MLE for each.

- Use **percentiles** of bootstrapped MLEs for CIs.

Correctly Specified CIs:

- Desired: Coverage probability near 95%. ($>$ 90% acceptable.)
- **Challenge**: Actual coverage may deviate.

BCa adjustments counteracts bias and skewness in estimates.

Bootstrap Confidence Intervals (CIs)
Confidence Intervals (CI) quantify the uncertainty in our estimate.
Asymptotic Sampling Distribution of MLE is a popular choice for constructing CIs.
Challenge: Asymptotic distribution may not be accurate for small sample sizes.
Particularly since we're dealing with right-censoring and masking.
Bootstrapped CIs: Resample data and obtain MLE for each.
Use percentiles of bootstrapped MLEs for CIs.
Correctly Specified CIs:
Desired: Coverage probability near 95%. (> 90% acceptable.)
Challenge: Actual coverage may deviate.
BCa adjustments counteracts bias and skewness in estimates.

- **Goal**: Need a way to measure the uncertainty in our estimate.
- **CIs** are a popular; they help us pin down the likely range of values for our parameters.
- **Bootstrap** the CIs, since there is a lot of bias and variability in our estimate due to the masking and censoring in our small data sets and the asymptotic distribution is not likely to be accurate.
- **Specified**: We want our CIs to be correctly specified, meaning they cover the true parameter value around 95
- **BCa**: But they may be too low or too high; we use the BCa method to adjust for bias and skewness in the estimate. A coverage probability above 90% is acceptable.

Challenges with MLE on Masked Data

We discovered some challenges with the MLE.

Convergence Issues: Flat likelihood regions were observed due to the ambiguity in the masked data and small sample sizes.

Bootstrap Issues: Bootstrap relies on the Law of Large Numbers.

- It might not represent the true variability for small samples.
- Due to censoring and masking, the effective sample size is reduced.

Mitigation: In simulation study, we discard non-convergent samples for the MLE on original data, but keep all resamples for the bootstrap.

- Helps ensure robustness of the results, while acknowledging the inherent complexities introduced by masking and censoring.
- We report convergence rates in our simulation study.

Challenges with MLE on Masked Data

We discovered some challenges with the MLE.

Convergence Issues: Flat likelihood regions were observed due to the ambiguity in the masked data and small sample sizes.

Bootstrap Issues: Bootstrap relies on the Law of Large Numbers.

- It might not represent the true variability for small samples.
- Due to censoring and masking, the effective sample size is reduced.

Mitigation: In simulation study, we discard non-convergent samples for the MLE on original data, but keep all resamples for the bootstrap.

- Helps ensure robustness of the results, while acknowledging the inherent complexities introduced by masking and censoring.
- We report convergence rates in our simulation study.

Like any model, ours has its challenges:

- **Masking**: Masking and censoring, combined with small sample sizes, can cause flat likelihood regions, which can lead to convergence issues.
- **Small**: For small samples, bootstrapping may not always capture the true variability in the data **Approach**: We take the following approach in our simulation study. **Discard**: We discard non-convergent samples for the MLE on original data but retain all MLEs for the resampled data. **Robustness**: This helps ensure the robustness of our results while acknowledging the inherent complexities introduced by masking and censoring. **Convergence Rate**: We report the convergence rate in our simulation study.

Section 3

Simulation Study

Section 3
Simulation Study

Series System: Weibull Components

The lifetime of the j^{th} component in the i^{th} system:

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

- λ_j is the **scale** parameter
- k_j is the **shape** parameter:
  - $k_j < 1$: Indicates infant mortality.
  - $k_j = 1$: Indicates random failures.
  - $k_j > 1$: Indicates wear-out failures.

Recall that for a series system:

- **Series Reliability** is the product of the component reliabilities.
- **Hazard** is the sum of the component hazard functions.
- **Likelihood**: $L(\boldsymbol{\theta}) \propto \prod_{i=1}^{n} R_{T_i}(t_i; \boldsymbol{\theta}) \left[\sum_{j \in c_i} h_j(t_i; \boldsymbol{\theta_j})\right]^{\delta_i}$.

Series System: Weibull Components

The lifetime of the j^{th} component in the i^{th} system:

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

- λ_j is the **scale** parameter
- k_j is the **shape** parameter:
  - $k_j < 1$: Indicates infant mortality.
  - $k_j = 1$: Indicates random failures.
  - $k_j > 1$: Indicates wear-out failures.

Recall that for a series system:

- **Series Reliability** is the product of the component reliabilities.
- **Hazard** is the sum of the component hazard functions.
- **Likelihood**: $L(\boldsymbol{\theta}) \propto \prod_{i=1}^{n} R_{T_i}(t_i; \boldsymbol{\theta}) \left[\sum_{j \in c_i} h_j(t_i; \boldsymbol{\theta_j})\right]^{\delta_i}$.

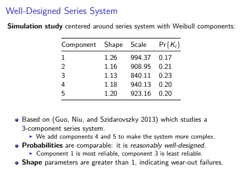
- **Weibull**: We model a series system with Weibull components.
- **Component Functions**: Hazard and reliability functions are well-known for Weibull.
- **Shape** parameter tells us a lot about the failure characteristics.
- **Increasing**: When the function is increasing, think of it as wearing-out over time.
- **Decreasing**: If it's decreasing, it usually signals some early-life challenges.
- **Series System**: Recall that for a series system, the reliability is the product of the component reliabilities and the hazard function is the sum of the component hazard functions.
- **Likelihood**: The likelihood is the same as before, we've just reproduced it here.

Well-Designed Series System

Simulation study centered around series system with Weibull components:

Component	Shape	Scale	$\Pr\{K_i\}$
1	1.26	994.37	0.17
2	1.16	908.95	0.21
3	1.13	840.11	0.23
4	1.18	940.13	0.20
5	1.20	923.16	0.20

- Based on (Guo, Niu, and Szidarovszky 2013) which studies a 3-component series system.
  - We add components 4 and 5 to make the system more complex.
- **Probabilities** are comparable: it is *reasonably well-designed*.
  - Component 1 is most reliable, component 3 is least reliable.
- **Shape** parameters are greater than 1, indicating wear-out failures.



- **Centered**: This study is centered around a series system with Weibull components.
- **Based**: It's based on a paper that studies a 3-component series system.
- **Added**: We added components 4 and 5 to make it more complex.
- **Probability**: We show the probability of each component being the cause of failure.
- **Well-Designed**: The probabilities are comparable, so no weak links. It's reasonably well-designed. Component 1 is most reliable, component 3 is least.
- **Parameters**: We show the shape and scale parameters for each component.
- **Wear-Out**: The shape parameters are greater than 1, indicating components are likely to fail due to wear-out.

Data Generation

Latent Component Lifetimes are generated for each system in the study.

Right-censoring: In our simulation study, we independently control the probability q (quantile) of right-censoring by finding the value τ that satisfies $\Pr\{T_i < \tau\} = q$.

- $S_i = \min(T_i, \tau)$ and $\delta_i = 1_{\{T_i < \tau\}}$.

Masking Component Failures: The *Bernoulli Masking Model* is used to mask component cause of failure, parameterized by masking probability p.

- p chosen independently: at the extremes, if $p = 0$ there is no masking, and if $p = 1$, there is total masking.
- We describe the process and how it satisfies the masking conditions next.

Data Generation

Latent Component Lifetimes are generated for each system in the study.

Right-censoring: In our simulation study, we independently control the probability q (quantile) of right-censoring by finding the value τ that satisfies $\Pr\{T_i < \tau\} = q$.

- $S_i = \min(T_i, \tau)$ and $\delta_i = 1_{\{T_i < \tau\}}$.

Masking Component Failures: The *Bernoulli Masking Model* is used to mask component cause of failure, parameterized by masking probability p.

- p chosen independently: at the extremes, if $p = 0$ there is no masking, and if $p = 1$, there is total masking.
- We describe the process and how it satisfies the masking conditions next.

- **Data Generation**: We generate the latent component lifetimes for the series system we just discussed.
- **Observed Data**: Then, we generate the data we actually see, which is based on the component data.
- **Right-Censoring**: We control the probability of right-censoring by finding the value of τ that satisfies the quantile q. Then, we set the right-censoring time to be the minimum of the system lifetime and τ. The event indicator is 1 if the system fails before τ, 0 otherwise.
- **Masking**: We use a Bernoulli masking model to mask the component cause of failure. We parameterize the level of masking by the masking probability, p.
- We parameterize the level of masking by the masking probability, p, which specifies that each non-failed component has a p probability of masking the failed component by including it in the candidate set.

Data Generation: Satisfying Masking Conditions

We generate the candidate sets for each system in the study.

Satisfying Masking Conditions:

- **Condition 1**: The failed component deterministically placed in candidate set.
- **Condition 2**: By using a Bernoulli distribution with a constant probability p for all components, probability of a candidate set is constant as we vary which component failed within set.
- **Condition 3**: Masking only depends on the fixed parameter p and doesn't interact with the system parameter θ.

Data Generation: Satisfying Masking Conditions

We generate the candidate sets for each system in the study.

Satisfying Masking Conditions:

- **Condition 1**: The failed component deterministically placed in candidate set.
- **Condition 2**: By using a Bernoulli distribution with a constant probability p for all components, probability of a candidate set is constant as we vary which component failed within set.
- **Condition 3**: Masking only depends on the fixed parameter p and doesn't interact with the system parameter θ.

- **Masking**: We use a Bernoulli masking model for masking the failed component.
- This satisifes the masking failure conditions in the following ways:
- **Condition 1**: The failed component is deterministically placed in the candidate set.
- **Condition 2**: The probability of masking is the same for all components, so the probability of the candidate set is constant across components.
- **Condition 3**: The masking probability is independent of the parameters.

Performance Metrics

Objective: Evaluate the MLE and BCa confidence intervals' performance across various scenarios.

- **MLE Evaluation**:
  - **Accuracy**: Proximity of the MLE's expected value to the actual value.
  - **Precision**: Consistency of the MLE across samples.
- **BCa Confidence Intervals Evaluation**:
  - **Accuracy**: Confidence intervals (CIs) should cover true parameters around 95% of the time.
    - Coverage probability (CP)
  - **Precision**: Assessed by the width of the CIs.

Both accuracy and precision are crucial for confidence in the analysis.

Performance Metrics

Objective: Evaluate the MLE and BCa confidence intervals' performance across various scenarios.

- **MLE Evaluation**:
  - **Accuracy**: Proximity of the MLE's expected value to the actual value.
  - **Precision**: Consistency of the MLE across samples.
- **BCa Confidence Intervals Evaluation**:
  - **Accuracy**: Confidence intervals (CIs) should cover true parameters around 95% of the time.
    - Coverage probability (CP)
  - **Precision**: Assessed by the width of the CIs.

Both accuracy and precision are crucial for confidence in the analysis.

- **Objective**: We want to evaluate the performance of the MLE and BCa confidence intervals across various scenarios.
- **MLE Evaluation**: We evaluate the MLE in terms of accuracy and precision.
- **Accuracy**: Accuracy is the proximity of the MLE's expected value to the actual value.
- **Precision**: Precision is the consistency of the MLE across samples.
- **BCa Confidence Intervals Evaluation**: We evaluate the BCa confidence intervals in terms of accuracy and precision.
- **Accuracy**: Accuracy is measured by the coverage probability, which is the proportion of times the confidence interval covers the true parameter.
- **Precision**: Precision is assessed by the width of the confidence interval.

Scenario: Impact of Right-Censoring

Simulation Setup: - Vary the right-censoring quantile (q): 60% to 100%. - Fixed the masking probability p to 21.5% - Fixed the sample size n to 90.

Background

- **Right-Censoring**: No failure observed.
- **Impact**: Reduces the effective sample size.
- **MLE**: Bias and precision affected by censoring.

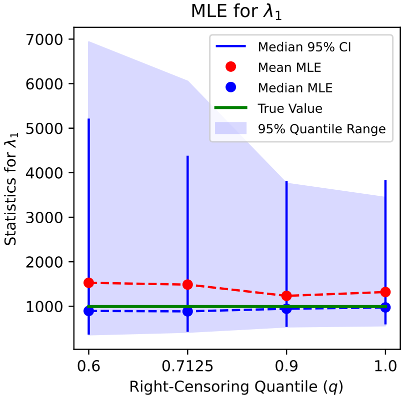
Scenario: Impact of Right-Censoring

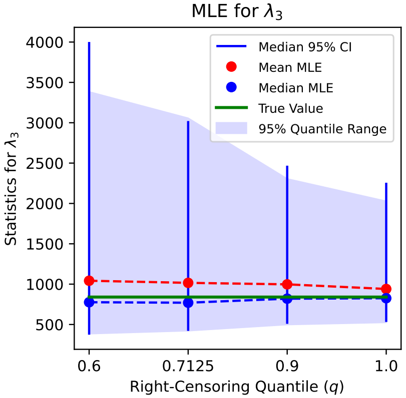
Simulation Setup: - Vary the right-censoring quantile (q): 60% to 100%. - Fixed the masking probability p to 21.5% - Fixed the sample size n to 90.

Background

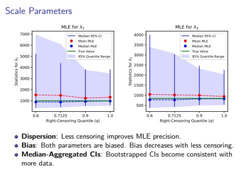
- **Right-Censoring**: No failure observed.
- **Impact**: Reduces the effective sample size.
- **MLE**: Bias and precision affected by censoring.

Scale Parameters

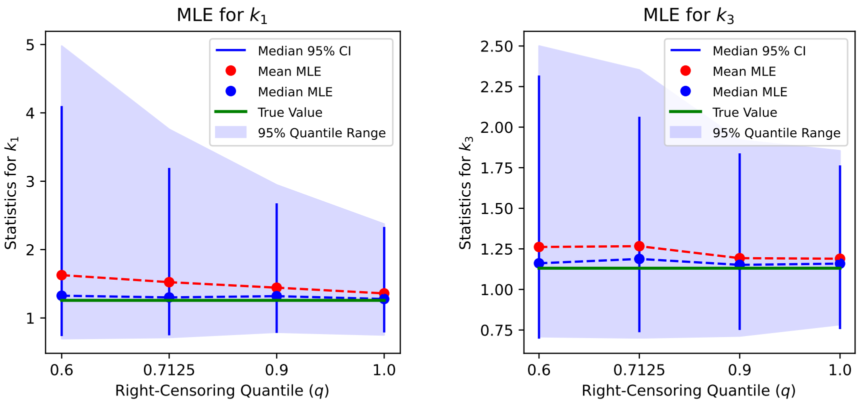




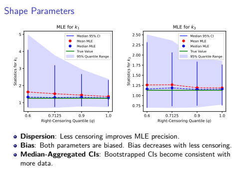
- **Dispersion**: Less censoring improves MLE precision.
- **Bias**: Both parameters are biased. Bias decreases with less censoring.
- **Median-Aggregated CIs**: Bootstrapped CIs become consistent with more data.



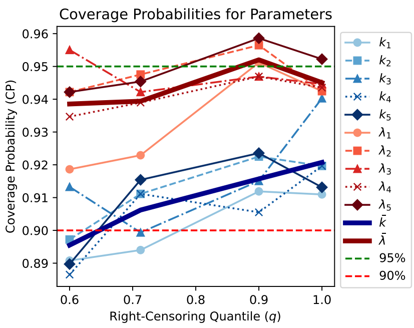
Shape Parameters

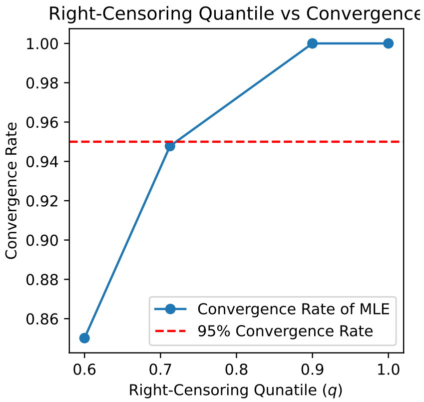


- **Dispersion**: Less censoring improves MLE precision.
- **Bias**: Both parameters are biased. Bias decreases with less censoring.
- **Median-Aggregated CIs**: Bootstrapped CIs become consistent with more data.

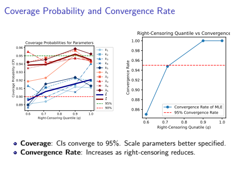


Coverage Probability and Convergence Rate





- **Coverage**: CIs converge to 95%. Scale parameters better specified.
- **Convergence Rate**: Increases as right-censoring reduces.



Conclusion

- MLE precision improves, bias drops with decreased right-censoring.
- BCa CIs perform well, particularly for scale parameters.
- MLE of most reliable component more affected by right-censoring.

Conclusion

- MLE precision improves, bias drops with decreased right-censoring.
- BCa CIs perform well, particularly for scale parameters.
- MLE of most reliable component more affected by right-censoring.

Impact of Masking Probability

Simulation Setup: - Vary the masking probability p: 0.1 to 0.7. - Fixed the right-censoring quantile q to 0.825. - Fixed the sample size n to 90.

Background

- **Masking** adds ambiguity in identifying the failed component.
- Impacts on MLE:
  - **Ambiguity**: Higher p increases uncertainty in parameter adjustment.
  - **Bias**: Similar to right-censoring, but affected by both p and q.
  - **Precision**: Reduces as p increases.

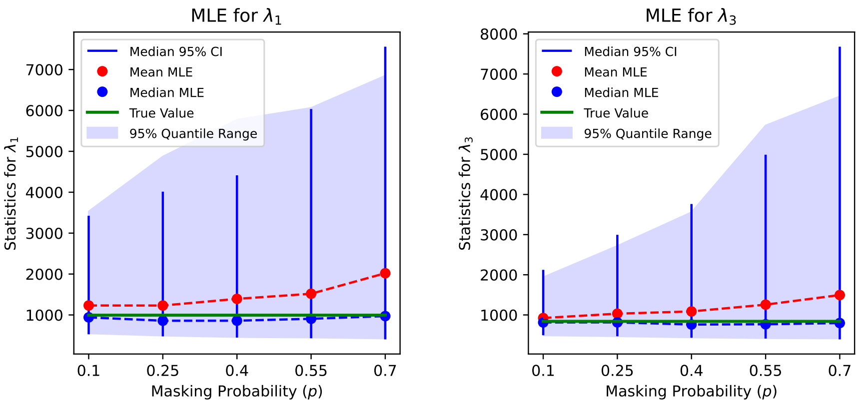
Impact of Masking Probability

Simulation Setup: - Vary the masking probability p: 0.1 to 0.7. - Fixed the right-censoring quantile q to 0.825. - Fixed the sample size n to 90.

Background

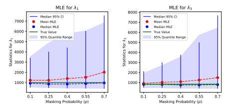
- **Masking** adds ambiguity in identifying the failed component.
- Impacts on MLE:
  - **Ambiguity**: Higher p increases uncertainty in parameter adjustment.
  - **Bias**: Similar to right-censoring, but affected by both p and q.
  - **Precision**: Reduces as p increases.

Scale Parameters



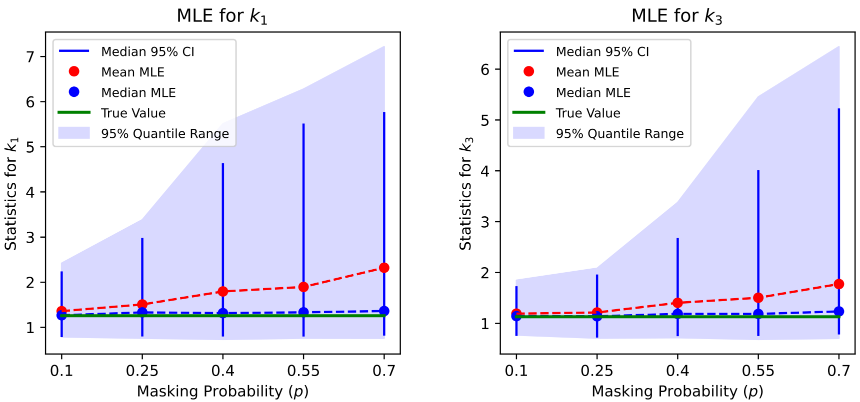
- **Dispersion**: Increases with p, indicating reduced precision.
- **Bias**: Positive bias rises with p.
- **Median-Aggregated CIs**: Widen and show asymmetry as p grows.

Scale Parameters



- **Dispersion**: Increases with p, indicating reduced precision.
- **Bias**: Positive bias rises with p.
- **Median-Aggregated CIs**: Widen and show asymmetry as p grows.

Shape Parameters

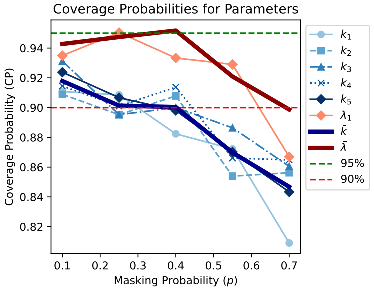


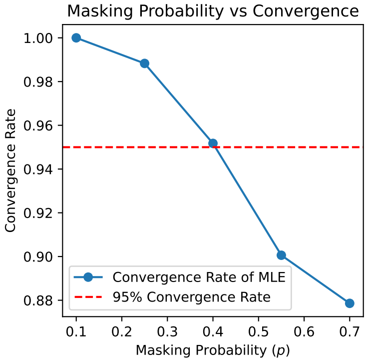
- **Dispersion**: Increases with p, indicating reduced precision.
- **Bias**: Positive bias rises with p.
- **Median-Aggregated CIs**: Widen and show asymmetry as p grows.

Shape Parameters

- **Dispersion**: Increases with p, indicating reduced precision.
- **Bias**: Positive bias rises with p.
- **Median-Aggregated CIs**: Widen and show asymmetry as p grows.

Coverage Probability and Convergence Rate

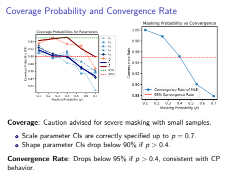




Coverage: Caution advised for severe masking with small samples.

- Scale parameter CIs are correctly specified up to $p = 0.7$.
- Shape parameter CIs drop below 90% if $p > 0.4$.

Convergence Rate: Drops below 95% if $p > 0.4$, consistent with CP behavior.



Conclusion

- Masking influences MLE precision, coverage probability, and introduces bias.
- The bias is positive and increases with p, essentially adding a censoring-like effect to the components.
- Despite significant masking, scale parameters have commendable CI coverage.

Conclusion

- Masking influences MLE precision, coverage probability, and introduces bias.
- The bias is positive and increases with p, essentially adding a censoring-like effect to the components.
- Despite significant masking, scale parameters have commendable CI coverage.

Impact of Sample Size

Assess the impact of sample size on MLEs and BCa CIs.

Simulation Setup: - Vary sample size n: 50 to 500. - Fixed the masking probability p to 0.215. - Fixed the right-censoring quantile q to 0.825.

Background

- **Sample Size**: Number of systems observed.
- **Impact**: More data reduces uncertainty in parameter estimation.
- **MLE**: Mitigates biasing effects of right-censoring and masking.

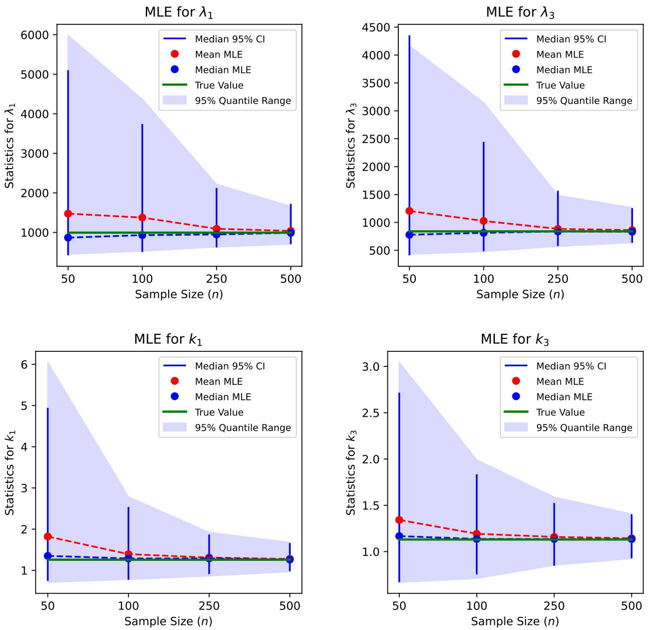
Impact of Sample Size

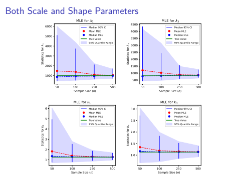
Assess the impact of sample size on MLEs and BCa CIs.

Simulation Setup: - Vary sample size n: 50 to 500. - Fixed the masking probability p to 0.215. - Fixed the right-censoring quantile q to 0.825.

Background
- Sample Size: Number of systems observed.
- Impact: More data reduces uncertainty in parameter estimation.
- MLE: Mitigates biasing effects of right-censoring and masking.

Both Scale and Shape Parameters





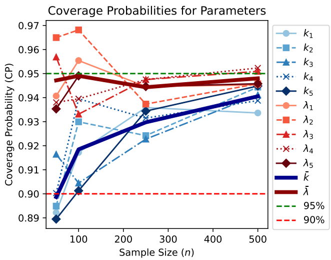
Parameters

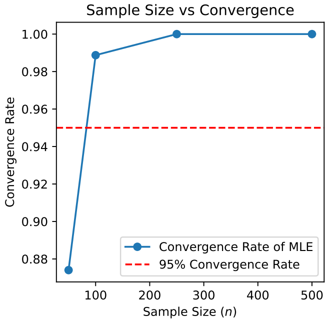
- **Dispersion**:
  - Dispersion reduces with n—indicating improved precision.
  - Disparity observed between components k_1, λ_1 and k_3, λ_3.
- **Bias**:
  - High positive bias initially, but diminishes around $n = 250$.
  - Enough sample data can counteract right-censoring and masking effects.
- **Median-Aggregated CIs**:
  - CIs tighten as n grows—showing more consistency.
  - Upper bound more dispersed than lower, reflecting the MLE bias direction.

Parameters

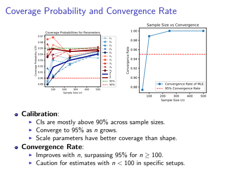
- **Dispersion**:
  - Dispersion reduces with n—indicating improved precision.
  - Disparity observed between components k_1, λ_1 and k_3, λ_3.
- **Bias**:
  - High positive bias initially, but diminishes around $n = 250$.
  - Enough sample data can counteract right-censoring and masking effects.
- **Median-Aggregated CIs**:
  - CIs tighten as n grows—showing more consistency.
  - Upper bound more dispersed than lower, reflecting the MLE bias direction.

Coverage Probability and Convergence Rate





- **Calibration**:
  - CIs are mostly above 90% across sample sizes.
  - Converge to 95% as n grows.
  - Scale parameters have better coverage than shape.
- **Convergence Rate**:
  - Improves with n, surpassing 95% for $n \geq 100$.
  - Caution for estimates with $n < 100$ in specific setups.



Conclusion

- Sample size significantly mitigates challenges from right-censoring and masking.
- MLE precision and accuracy enhance notably with growing samples.
- BCa CIs become narrower and more reliable as sample size increases.

Conclusion

- Sample size significantly mitigates challenges from right-censoring and masking.
- MLE precision and accuracy enhance notably with growing samples.
- BCa CIs become narrower and more reliable as sample size increases.

Section 4

Conclusion

Section 4
Conclusion

Conclusion

Key Findings

- Employed maximum likelihood techniques for component reliability estimation in series systems with masked failure data.
- Methods performed robustly despite masking and right-censoring challenges.

Simulation Insights

- Right-censoring and masking introduce positive bias; more reliable components are most affected.
- Shape parameters harder to estimate than scale.
- Large samples can counteract these challenges.

Conclusion

Key Findings

- Employed maximum likelihood techniques for component reliability estimation in series systems with masked failure data.
- Methods performed robustly despite masking and right-censoring challenges.

Simulation Insights

- Right-censoring and masking introduce positive bias; more reliable components are most affected.
- Shape parameters harder to estimate than scale.
- Large samples can counteract these challenges.

Conclusion (cont.)

Confidence Intervals

- Bootstrapped BCa CIs demonstrated commendable coverage probabilities, even in smaller sample sizes.

Takeaways

- Framework offers a rigorous method for latent component property estimation from limited observational data.
- Techniques validated to provide practical insights in diverse scenarios.
- Enhanced capability for learning from obscured system failure data.

Conclusion (cont.)

Confidence Intervals
- Bootstrapped BCa CIs demonstrated commendable coverage probabilities, even in smaller sample sizes.

Takeaways
- Framework offers a rigorous method for latent component property estimation from limited observational data.
- Techniques validated to provide practical insights in diverse scenarios.
- Enhanced capability for learning from obscured system failure data.

Future Work and Discussion

Directions to enhance learning from masked data:

- **Relax Masking Conditions**: Assess sensitivity to violations and and explore alternative likelihood models.
- **System Design Deviations**: Assess estimator sensitivity to deviations.
- **Homogenous Shape Parameter**: Analyze trade-offs with the full model.
- **Bootstrap Techniques**: Semi-parametric approaches and prediction intervals.
- **Regularization**: Data augmentation and penalized likelihood methods.
- **Additional Likelihood Contributions**: Predictors, etc.

Future Work and Discussion

Directions to enhance learning from masked data:

- **Relax Masking Conditions**: Assess sensitivity to violations and and explore alternative likelihood models.
- **System Design Deviations**: Assess estimator sensitivity to deviations.
- **Homogenous Shape Parameter**: Analyze trade-offs with the full model.
- **Bootstrap Techniques**: Semi-parametric approaches and prediction intervals.
- **Regularization**: Data augmentation and penalized likelihood methods.
- **Additional Likelihood Contributions**: Predictors, etc.