

Masterarbeit

Konfidenzintervalle für kombinierte
Wählermodelle

von

Hannes Christiansen

betreut durch

Prof. Dr. Jürgen Franke

in Kooperation mit

Fraunhofer Institut für Techno- und Wirtschaftsmathematik

Dr. Sascha Feth

Kurzfassung

Am Fraunhofer Institut für Techno- und Wirtschaftsmathematik in Kaiserslautern wurde ein kombiniertes Wöhlermodell entwickelt, mit dem sich Wöhlerversuche beschreiben lassen, die im Übergang von Zeit- und Dauerfestigkeitsbereich durch ein (im doppelt-logarithmischen Sinne) lineares Gesetz nicht ausreichend abgebildet werden können. Um das zu erreichen, verwendet das Modell zwei Geraden mit einem variablen Abknickpunkt. Die Schätzung der Modellparameter erfolgt dabei durch das Maximum-Likelihood-Verfahren, um auch dem Einfluss zensierter Daten gerecht zu werden.

In der Praxis sind dabei Konfidenzintervalle für die einzelnen Parameter des Modells von besonderem Interesse. Es stellt sich die Frage, welches die effizienteste Methode zur Berechnung dieser ist. In dieser Arbeit werden verschiedene Methoden hierfür betrachtet und bezüglich ihrer Nutzungstauglichkeit hinsichtlich Genauigkeit und Geschwindigkeit bewertet.

Es zeigt sich, dass das Bootstrap-Verfahren, das Konfidenzintervalle aus Resamples konstruiert und dafür die Parameterschätzung so oft durchführen muss, wie Resamples erzeugt werden sollen, aufgrund der Komplexität des Modells mehrere Stunden bis Tage dauert. Das Verfahren lässt sich auch nicht dahingehend modifizieren, bei gleichbleibender Qualität durch eine vereinfachte Schätzung für jedes Resample deutlich schneller Ergebnisse zu liefern.

Aus dem Likelihood-Quotienten konstruierte Konfidenzintervalle, die auf der Invertierung des Likelihood-Quotienten-Tests basieren, sind nur für einen der beiden betrachteten Beispieldatensätze ausreichend genau, für den anderen deutlich zu groß.

Die aus der Fisher Information abgeleiteten Konfidenzintervalle, die zur Schätzung der Standardabweichung die asymptotische Normalität von Maximum-Likelihood-Schätzern ausnutzen, sind per Definition symmetrisch und weichen daher leicht von den Bootstrap-Konfidenzintervallen ab. Auf jeweils einer Seite stimmen die Intervallgrenzen aber fast überein und die Abweichungen sind so gering, dass die Konfidenzintervalle der Fisher Information eine effiziente Alternative bieten, deren Berechnung nur wenige Minuten dauert.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problemstellung	1
1.2	Methodik	2
1.3	Gliederung	2
1.4	Kurzfasit	2
2	Mathematische Grundlagen	3
2.1	Begriffsklärung und Notation	3
2.2	Zensierte Daten	3
2.3	Wahrscheinlichkeitsverteilungen	4
2.3.1	Normalverteilung	4
2.3.2	Weibull-Verteilung	4
2.3.3	Gumbel-Verteilung	5
2.3.4	Chi-Quadrat-Verteilung	6
2.4	Lineare Regression	6
2.5	Maximum-Likelihood-Verfahren	7
2.6	Konfidenzintervalle und -bereiche	8
2.6.1	Eindimensionale Konfidenzintervalle	8
2.6.2	Mehrdimensionale Konfidenzbereiche	8
2.6.3	Gemeinsame Konfidenzbereiche	8
3	Das Wöhlermodell	10
3.1	Problemstellung der Betriebsfestigkeit	10
3.2	Wöhlerversuch	10
3.3	Kombiniertes Wöhlermodell	12
3.3.1	Herleitung	12
3.3.2	Formalisierung	13
3.3.3	Anpassung für gleiche Streuung in Amplitudenrichtung	13
3.3.4	Modellierung einer Stichprobe	14
3.4	Beispieldatensätze	16
3.4.1	Beispiel A: Verdacht auf zwei Bereiche	16
3.4.2	Beispiel B: Fiktiver Beispieldatensatz	18
4	Parameterschätzung mit Maximum-Likelihood	21
4.1	Likelihood-Funktion für normalverteilte Lebensdauern	21
4.2	Likelihood-Funktion für Weibull-verteilte Lebensdauern	23
4.3	Schätzung der Modellparameter	23
4.3.1	Beispiel A	24
4.3.2	Beispiel B	25
5	Konfidenzintervalle mit Bootstrap	26
5.1	Bootstrap-Verfahren	26
5.2	Herleitung der Konfidenzintervalle	27
5.2.1	Bootstrap-Quantil-Methode	27

5.2.2	Bootstrap-Prinzip	27
5.3	Anwendung für Beispieldatensätze	27
5.3.1	Beispiel A	27
5.3.2	Beispiel B	28
5.4	Modifikationen zur Verringerung der Rechenzeit	29
5.4.1	Jackknife	30
5.4.2	Segmentierte lineare Regression	31
5.4.3	Zensierte Regression über Kaplan-Meier-Schätzer	32
6	Konfidenzintervalle mit Likelihood Quotient	33
6.1	Likelihood Quotient	33
6.2	Herleitung der Konfidenzintervalle	33
6.3	Anwendung für Beispieldatensätze	34
6.3.1	Beispiel A	34
6.3.2	Beispiel B	35
7	Konfidenzintervalle mit Fisher Information	36
7.1	Fisher Information	36
7.2	Herleitung der Konfidenzintervalle	36
7.3	Anwendung für Beispieldatensätze	37
7.3.1	Beispiel A	38
7.3.2	Beispiel B	38
8	Fazit	40
9	Ausblick	42
10	Literatur	43
Anhang		I
A Sätze und Beweise		I
B Partielle Ableitungen der Log-Likelihood-Funktion		III
C Quellcodes		VIII
D Abbildungsverzeichnis		XXIII
E Tabellenverzeichnis		XXIV
F Listing-Verzeichnis		XXV

1 Einleitung

1.1 Problemstellung

Für sicherheitsrelevante Bauteile ist ein Festigkeitsnachweis notwendig, damit ein ausfallsicherer Betrieb über die Nutzungsdauer gewährleistet werden kann. Dieser Nachweis ist für schwingend beanspruchte Bauteile gegen eine zyklische Beanspruchung zu erbringen und kann experimentell durch sogenannte Wöhlerversuche erbracht werden [16]. Dabei werden die baugleichen Versuchskörper bei unterschiedlicher, jeweils aber konstanter Lastamplitude solange beansprucht, bis ein definiertes Versagen eintritt oder eine festgelegte Anzahl an Schwingungen ohne Ausfall überstanden wurde. Die so erhaltenen Lebensdauern, die im zweiten Fall zensiert sind, da nur ein Mindestwert bekannt ist, werden in einem doppelt-logarithmischen Diagramm eingetragen. Dort gibt die an diese Daten angepasste Wöhlerlinie Aufschluss über den Zusammenhang zwischen Beanspruchung und Festigkeit.

Am Fraunhofer Institut für Techno- und Wirtschaftsmathematik in Kaiserslautern wurde ein kombiniertes Wöhlermodell entwickelt, mit dem sich durch eine stückweise lineare Wöhlerlinie auch Wöhlerversuche beschreiben lassen, die im unteren Bereich, dem Übergang vom Zeit- zum Dauerfestigkeitsbereich, durch das lineare Gesetz nicht ausreichend abgebildet werden können [13], wie beispielhaft in Abbildung 1 dargestellt.

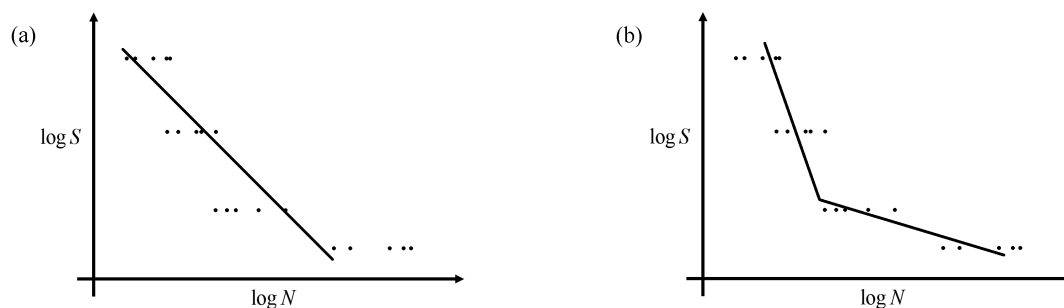


Abb. 1: Vergleich des einfachen Wöhlermodells mit nur einer Geraden (a) und des kombinierten Wöhlermodells mit zwei Steigungen (b) für eine fiktive Stichprobe.

Zur Berechnung von Konfidenzintervallen für die Modellparameter existieren verschiedene Methoden, die aufgrund der Komplexität des kombinierten Wöhlermodells teilweise sehr rechenintensiv sind. Es stellt sich die Frage, welches die effizienteste in diesem Zusammenhang ist. In dieser Arbeit werden ausgewählte Methoden betrachtet und bezüglich ihrer Nutzungstauglichkeit hinsichtlich Genauigkeit und Geschwindigkeit bewertet.

1.2 Methodik

Um die Nutzungstauglichkeit der Verfahren hinsichtlich Genauigkeit und Geschwindigkeit zu überprüfen, werden die Verfahren an zwei Beispieldatensätzen getestet. Dabei handelt es sich um einen realen Beispieldatensatz, für den die wahren Modellparameter unbekannt sind, und einen fiktiven Beispieldatensatz, der mithilfe von Zufallszahlen für vorgegebene, bekannte Modellparameter erstellt wurde. Es wird die Berechnungsdauer gemessen und jeweils sowohl die Größe als auch die Lage der Konfidenzintervalle im Vergleich zum realen beziehungsweise geschätzten Parameter betrachtet.

1.3 Gliederung

Zunächst werden in **Kapitel 2** die wichtigsten mathematischen Begriffe und Notationen eingeführt, und in **Kapitel 3** das der Arbeit zugrundeliegende Wählermodell erläutert sowie im Speziellen das kombinierte Wählermodell formalisiert. In **Kapitel 4** wird dargestellt, wie die einzelnen Parameter dieses Modells mit dem Maximum-Likelihood-Verfahren geschätzt werden können und welche geschätzten Parameter sich für die Beispieldatensätze ergeben.

Die folgenden Kapitel enthalten die zur Konstruktion von Konfidenzintervallen untersuchten Verfahren. Es wird jeweils die zugrundeliegende Theorie erläutert und für die Beispieldatensätze angewandt. In **Kapitel 5** wird das Bootstrap-Verfahren verwendet sowie Modifikationen zur Verringerung der Rechenzeit diskutiert. Als Alternative werden in **Kapitel 6** Konfidenzintervalle mithilfe des Likelihood Quotienten und in **Kapitel 7** mithilfe der Fisher Information konstruiert.

Zum Abschluss werden in **Kapitel 8** die Ergebnisse zusammengefasst und bewertet, und in **Kapitel 9** wird ein Ausblick über weiterführende Überlegungen gegeben.

1.4 Kurzfazit

Die aus dem Bootstrap-Verfahren bestimmten Konfidenzintervalle werden zwar als am genauesten betrachtet, ihre Berechnung dauert aber mehrere Stunden bis Tage. Mithilfe der Fisher Information lassen sich hingegen innerhalb kürzester Zeit Konfidenzintervalle bestimmen, die zwar eine Normalverteilung der Parameter annehmen und per Definition symmetrisch sind, aber vergleichbare Intervalle liefern und damit eine effiziente Alternative darstellen.

2 Mathematische Grundlagen

Zum besseren Verständnis der Arbeit werden die wichtigsten der verwendeten mathematischen Notationen und Begriffe kurz eingeführt. Für eine grundlegende Einführung in die mathematische Statistik wird auf [22] verwiesen.

2.1 Begriffsklärung und Notation

Definition 2.1.1 (Notationen)

In dieser Arbeit werden folgende Notationen verwendet:

- (a) $\ln(x)$ ist der Logarithmus zur Basis e , $\log(x)$ der Logarithmus zur Basis 10.
- (b) X, Y, Z, \dots bezeichnen Zufallsvariablen über einem nicht genauer betrachteten Wahrscheinlichkeitsraum (Ω, \mathcal{A}, P) .
- (c) F, G, H, \dots bezeichnen Wahrscheinlichkeitsverteilungen von Zufallsvariablen, die dazugehörigen Dichtefunktionen werden mit Kleinbuchstaben f, g, h, \dots benannt.
- (d) θ bezeichnet den (unbekannten) Parameter(-Vektor) einer Verteilung oder eines Modells und $\hat{\theta}$ einen Schätzer davon.
- (e) $\underline{X} = (X_1, \dots, X_n)$ bezeichnet eine Stichprobe der Größe n , wobei jedes X_i eine Beobachtung der Zufallsvariablen X ist.

2.2 Zensierte Daten

Ein zensierter Wert beschreibt eine unvollständige Beobachtung, bei der der exakte Wert des zu beobachtenden Ereignisses unbekannt ist. Der Wert ist *rechts zensiert*, wenn das Ereignis bis zum Ende der Beobachtung noch nicht aufgetreten ist. Dieser Fall liegt bei Prüfstandsversuchen zum Beispiel vor, wenn der Versuch aus Geduldsgründen vorzeitig abgebrochen wird oder das Bauteil wegen einer anderen Ausfallursache nicht weiter getestet werden kann. In der Betriebsfestigkeit werden zensierte Werte meist als *Durchläufer* bezeichnet. Mathematisch lässt sich das folgendermaßen formalisieren [3, S. 5]:

Definition 2.2.1 (Zensierter Wert)

Sei T_i der Ausfallzeitpunkt des i -ten Elements einer Stichprobe \underline{X} und c_i die Dauer der Beobachtung. Beobachtet wird dann $\underline{X} = ((x_i, d_i)_{i=1, \dots, n})$ mit

$$(x_i, d_i) = \begin{cases} (T_i, 1), & \text{falls } T_i \leq c_i \quad (\text{unzensiert}) \\ (c_i, 0), & \text{falls } T_i > c_i \quad (\text{zensiert}). \end{cases} \quad (2.1)$$

2.3 Wahrscheinlichkeitsverteilungen

2.3.1 Normalverteilung

Die Normalverteilung ist eine parametrische Wahrscheinlichkeitsverteilung mit Erwartungswert μ und Varianz σ^2 . Sie ist durch die Verteilungsfunktion (2.2) beziehungsweise die Dichtefunktion (2.3) gegeben [11, S. 267] und wird im Folgenden durch $\mathcal{N}(\mu, \sigma^2)$ abgekürzt. Für die Standardnormalverteilung $\mathcal{N}(0, 1)$ wird die Dichtefunktion mit $\phi(x)$ und die Verteilungsfunktion mit $\Phi(x)$ bezeichnet.

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} dt \quad (2.2)$$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (2.3)$$

Der zentrale Grenzwertsatz besagt, dass das arithmetische Mittel einer großen Anzahl beobachteter Zufallsvariablen mit beliebiger, gleicher Verteilung näherungsweise normalverteilt ist [4, S. 213ff]. Daher gilt die Normalverteilung als bedeutendste Wahrscheinlichkeitsverteilung und wird insbesondere dann häufig verwendet, wenn nichts über die wahre Verteilungsfunktion bekannt ist [4, S. 231f].

2.3.2 Weibull-Verteilung

Die Weibull-Verteilung ist eine parametrische Wahrscheinlichkeitsverteilung mit charakteristischer Lebensdauer $T > 0$ und Formparameter $b > 0$. Sie ist durch die Verteilungsfunktion (2.4) beziehungsweise die Dichtefunktion (2.5) gegeben [27] und wird im Folgenden durch $\mathcal{W}(T, b)$ abgekürzt.

$$F(x) = 1 - e^{-\left(\frac{x}{T}\right)^b}, \quad x \geq 0 \quad (2.4)$$

$$f(x) = \frac{b}{T} \left(\frac{x}{T}\right)^{b-1} e^{-\left(\frac{x}{T}\right)^b}, \quad x \geq 0 \quad (2.5)$$

$$h_F(x) = \frac{f(t)}{1 - F(t)} = \frac{b}{T} \left(\frac{x}{T}\right)^{b-1}. \quad (2.6)$$

Für die Ausfallrate zum Zeitpunkt x , die als Tendenz eines bevorstehenden Ausfalls interpretiert werden kann, ergibt sich (2.6). Mit $b < 1$ können fallende, mit $b = 1$ konstante und mit $b > 1$ steigende Ausfallraten modelliert werden [17, S. 230f]. Im in dieser Arbeit behandelten Gebiet der Betriebsfestigkeit stehen steigende Ausfallraten zur Modellierung von Ermüdungs- und Verschleißausfällen im Vordergrund, bei denen die Wahrscheinlichkeit eines Ausfalls mit steigender Betriebsdauer zunimmt.

Wegen der unterschiedlich modellierbaren Ausfallraten hat die Weibull-Verteilung einen großen Anwendungsbereich. Sie spielt eine wichtige Rolle als Typ III der Extremwertverteilungen und ergibt sich als Verteilung für den frühesten Ausfall mehrerer miteinander verknüpfter Weibull-verteilter Komponenten (engl. *weakest link theory* [27]).

2.3.3 Gumbel-Verteilung

Die Gumbel-Verteilung ist eine parametrische Wahrscheinlichkeitsverteilung mit Skalensparameter $\beta > 0$ und Lageparameter $\mu \in \mathbb{R}$ und gehört als Typ I zur Familie der Extremwertverteilungen [15]. Sie ist durch die Verteilungsfunktion (2.7) beziehungsweise die Dichtefunktion (2.8) gegeben [11, S. 138] und wird im Folgenden durch $\mathcal{G}(\beta, \mu)$ abgekürzt.

$$F(x) = e^{-e^{-\frac{1}{\beta}(x-\mu)}} \quad (2.7)$$

$$f(x) = \frac{1}{\beta} e^{-\frac{1}{\beta}(x-\mu)} e^{-e^{-\frac{1}{\beta}(x-\mu)}} \quad (2.8)$$

Die Gumbel-Verteilung ergibt sich nach dem Transformationssatz für Dichten als Verteilung des Logarithmus einer Weibull-verteilten Zufallsvariable (siehe A.0.2) und spielt daher bei der Betrachtung Weibull-verteilter Zufallsgrößen auf einer logarithmischen Skala eine wichtige Rolle.

2.3.4 Chi-Quadrat-Verteilung

Die Chi-Quadrat-Verteilung, auch χ^2 -Verteilung, ergibt sich als Wahrscheinlichkeitsverteilung der Summe der Quadrate von standardnormalverteilten Zufallsvariablen, also von $Z = X_1^2 + \dots + X_n^2$ mit $X_1, \dots, X_n \stackrel{uiv}{\sim} \mathcal{N}(0, 1)$. Sie ist durch die Dichtefunktion (2.9) gegeben, wobei n die Anzahl der Freiheitsgrade bezeichnet [11, S. 69]. Am meisten Verwendung findet die Chi-Quadrat-Verteilung im Rahmen von Hypothesentests und bei der Konstruktion von Konfidenzintervallen.

$$f_n(x) = \frac{2^{-\frac{n}{2}}}{\Gamma(\frac{n}{2})} x^{\frac{n}{2}-1} e^{-\frac{x}{2}}, \quad x > 0 \quad (2.9)$$

$$\text{mit der Gamma-Funktion } \Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt \quad (2.10)$$

2.4 Lineare Regression

Das Ziel der Regression ist es, eine abhängige Variable y , den *Regressand*, durch eine oder mehrere unabhängige Variablen X , die *Regressoren*, zu erklären, indem die *Regressionskoeffizienten* β bestimmt werden. Im Falle der linearen Regression wird ein linearer Zusammenhang zwischen diesen Größen angenommen. Für $Y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times k}$, $\beta \in \mathbb{R}^k$ und $\epsilon \in \mathbb{R}^n$ wird

$$Y = X\beta + \epsilon \quad (2.11)$$

betrachtet, wobei die Residuen ϵ_i unkorreliert sind, Erwartungswert 0 besitzen und gleiche Varianz σ^2 haben [6, S. 78f]. Zum Schätzen der Regressionskoeffizienten wird die Quadratsumme der Residuen mit der Methode der kleinsten Quadrate minimiert, sodass sich

$$\hat{\beta} = (X^T X)^{-1} X^T Y \quad (2.12)$$

als Schätzer für β ergibt. Unter der Annahme, dass $\epsilon_i \stackrel{uiv}{\sim} \mathcal{N}(0, \sigma^2)$ gilt, entspricht dieser Schätzer gerade dem Maximum-Likelihood-Schätzer [5]. Die Annahme normalverteilter Residuen ist außerdem notwendig, um einfache Signifikanztests konstruieren zu können. Sind die Residuen anders verteilt oder liegen zensierte Daten vor, lässt sich das Regres-

sionsmodell nicht mehr so einfach lösen. In dem Fall bietet beispielsweise das Maximum-Likelihood-Verfahren eine einfacher zu handhabende Alternative.

2.5 Maximum-Likelihood-Verfahren

Maximum Likelihood ist ein Schätzverfahren, das die Parameter einer bestimmten Wahrscheinlichkeitsverteilung so wählt, dass die beobachteten Daten unter der Annahme der gewählten Wahrscheinlichkeitsverteilung am plausibelsten erscheinen [22, S. 28f].

Definition 2.5.1 (Maximum-Likelihood-Schätzer)

Für eine konkrete Stichprobe \underline{X} einer Zufallsvariablen X mit stetiger Verteilung aus $\{P_\theta \mid \theta \in \Theta\}$ ist die Likelihood-Funktion definiert als $L(\theta \mid \underline{X}) = f_\theta(\underline{X})$ und der dazugehörige Schätzer von θ als $\hat{\theta} = \operatorname{argmax}_{\theta \in \Theta} L(\theta \mid \underline{X})$.

Wenn \underline{X} aus n unabhängigen, identisch verteilten Realisierungen besteht, gilt $L(\theta \mid \underline{X}) = \prod_{i=1}^n f_\theta(X_i)$. Um die Maximierung dieser Funktion zu vereinfachen, wird in der Regel die *Log-Likelihood-Funktion* (2.13) verwendet, die das gleiche Maximum besitzt.

$$\ell(\theta \mid \underline{X}) = \ln \left(\prod_{i=1}^n f_\theta(X_i) \right) = \sum_{i=1}^n \ln f_\theta(X_i) \quad (2.13)$$

Im Falle zensierter Daten muss die Formel (2.13) angepasst werden. Für rechts-zensierte Daten wird $P(X > X_i)$ anstelle von $P(X = X_i)$ betrachtet. Mit $I = \{1, \dots, n\}$, sowie $I_A = \{i \in I, d_i = 1\}$ und $I_D = \{i \in I, d_i = 0\}$ als jeweilige Indexmenge der Ausfälle beziehungsweise der Durchläufer ergibt sich

$$\ell(\theta \mid \underline{X}) = \sum_{i \in I_A} \ln f_\theta(X_i) + \sum_{i \in I_D} \ln (1 - F_\theta(X_i)). \quad (2.14)$$

Unter bestimmten Voraussetzungen sind Maximum-Likelihood-Schätzer konsistent und asymptotisch normal (siehe A.0.3 und A.0.4).

2.6 Konfidenzintervalle und -bereiche

2.6.1 Eindimensionale Konfidenzintervalle

Ein Konfidenzintervall ist ein zufälliges Intervall, welches mit bestimmter Wahrscheinlichkeit den wahren Parameter überdeckt und als Qualitätsmerkmal für einen Schätzer dienen kann, indem es dessen Varianz verdeutlicht [6, S. 139f].

Definition 2.6.1 (Konfidenzintervall)

Für eine Zufallsvariable X mit stetiger Verteilung aus $\{P_\theta \mid \theta \in \Theta \subset \mathbb{R}\}$ und $\alpha \in (0, 1)$ ist das zufällige Intervall $[u(X), o(X)] \subseteq \Theta$ ein $(1 - \alpha)$ -Konfidenzintervall, wenn

$$P_\theta(u(X) \leq \theta \leq o(X)) \geq 1 - \alpha \quad \forall \theta \in \Theta. \quad (2.15)$$

Die Irrtumswahrscheinlichkeit α wird häufig als 0.05 gewählt, sodass ein 95 %-Konfidenzintervall bestimmt wird. Dabei wird in der Regel ein symmetrisches Konfidenzintervall mit $P_\theta(u(X) \leq \theta) \geq 1 - \alpha/2$ und $P_\theta(\theta \geq o(X)) \geq 1 - \alpha/2$ konstruiert.

2.6.2 Mehrdimensionale Konfidenzbereiche

Die Definition 2.6.1 lässt sich auch auf den mehrdimensionalen Fall übertragen, indem die jeweiligen Parameter als Vektoren betrachtet werden. Außerdem lässt sich die Definition für einen Konfidenzbereich beliebiger Form anstelle eines Konfidenzintervalls (beziehungsweise eines Hyperquaders im mehrdimensionalen Fall) ausweiten.

Definition 2.6.2 (Konfidenzbereich)

Für eine Zufallsvariable X mit stetiger Verteilung aus $\{P_\theta \mid \theta \in \Theta \subset \mathbb{R}^d\}$ und $\alpha \in (0, 1)$ ist der zufällige Bereich $A(X) \subseteq \Theta$ ein $(1 - \alpha)$ -Konfidenzbereich, wenn

$$P_\theta(\theta \in A(X)) \geq 1 - \alpha \quad \forall \theta \in \Theta. \quad (2.16)$$

2.6.3 Gemeinsame Konfidenzbereiche

Der gemeinsame Konfidenzbereich für $\theta = (\theta_1, \dots, \theta_n)$ lässt sich nicht ohne Weiteres aus eindimensionalen Konfidenzintervallen der einzelnen Parameter θ_i bestimmen. Lediglich falls alle θ_i untereinander unabhängig sind, ist der Hyperquader $A(X)$, der sich als Produkt

$A(X) = A_1(X) \times \cdots \times A_n(X)$ der eindimensionalen $(1 - \alpha_i)$ -Konfidenzintervallen mit $1 - \alpha_i = \sqrt[n]{1 - \alpha}$ ergibt, ein gemeinsamer $(1 - \alpha)$ -Konfidenzbereich für θ [6, S. 146].

Im Falle abhängiger Parameter θ_i kann die *Bonferroni-Korrektur* verwendet werden, um einen gemeinsamen Konfidenzbereich zu konstruieren. Dabei wird die Bonferroni-Ungleichung ausgenutzt, sodass durch $\alpha_i = \alpha/n$ das Produkt der eindimensionalen Konfidenzintervalle einen gemeinsamen $(1 - \alpha)$ -Konfidenzbereich liefert, der aufgrund der groben Abschätzung der Bonferroni-Ungleichung allerdings sehr konservativ ist.

Wenn man versucht, einen exakteren gemeinsamen Konfidenzbereich anzugeben, ist es unklar, welche Form dieser hat. Häufig wird dafür eine Hyperkugel oder ein Hyperellipsoid verwendet, aber auch ein unregelmäßiger Körper kann ebenso ein entsprechender Konfidenzbereich sein. In Abbildung 2 sind vier unterschiedliche Bereiche eingetragen, die jeweils genau 90% der eingetragenen Datenpunkte enthalten.

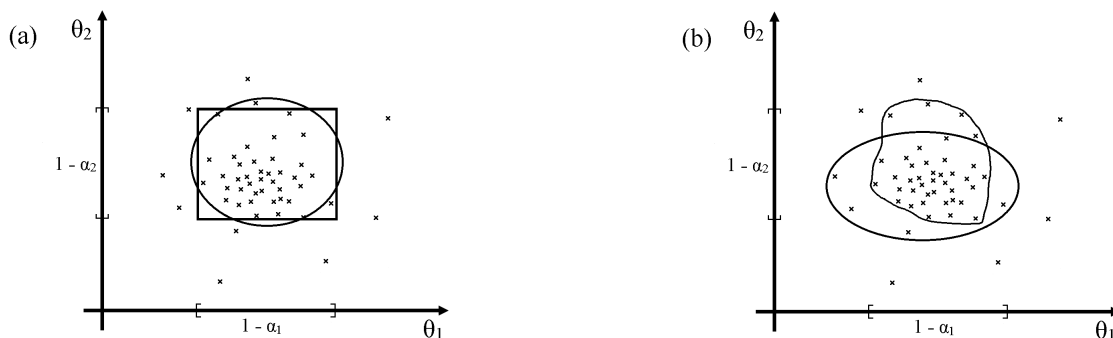


Abb. 2: Mögliche Formen von zweidimensionalen Konfidenzbereichen: Sowohl ein Rechteck oder ein Kreis (a) als auch eine Ellipse oder eine unregelmäßige Form (b) können einen entsprechenden Konfidenzbereich liefern. Alle vier enthalten in diesem Beispiel exakt 90% der eingetragenen Kreuze.

Im Rahmen dieser Arbeit werden aufgrund dieser Schwierigkeiten lediglich die Konfidenzintervalle einzelner Parameter betrachtet. Die Möglichkeiten der einzelnen Verfahren, gemeinsame Konfidenzbereiche zu liefern, werden im Ausblick in Kapitel 9 kurz erläutert.

3 Das Wöhlermodell

3.1 Problemstellung der Betriebsfestigkeit

Die Betriebsfestigkeit bewertet die Auslegung und Haltbarkeit von Bauteilen gegenüber ihrer mechanischen Beanspruchung und ergibt sich aus der Notwendigkeit, sowohl aus technischen, wirtschaftlichen als auch haftungsrechtlichen Gründen, einen ausfallsicheren Betrieb über die Nutzungsdauer mit hinreichender Sicherheit gewährleisten zu können. Ein Anriss oder Bruch von schwingbeanspruchten Bauteilen, der im Laufe der Zeit als Folge der schwingend einwirkenden Beanspruchung entsteht, kann verheerende Folgen haben, wie zum Beispiel 1998, als bei Eschede ein ICE entgleiste und 101 Menschen starben [16, S. 2]. Daher muss für die Bauteile ein Betriebsfestigkeitsnachweis erbracht werden, der mit statistisch begründeter Sicherheit die notwendige Festigkeit garantiert.

In einem vereinfachten Diagramm kann für ein Bauteil die zu ertragene Schwingspielzahl beziehungsweise Lebensdauer in einen Zusammenhang mit der Schwingbeanspruchung in Form der Amplitude der Spannungs-Dehnungs-Kurve gebracht werden. Dabei wird zwischen der Dauerfestigkeit und der Zeitfestigkeit unterschieden. Die Dauerfestigkeit beschreibt einen Grenzwert, bis zu dem eine Beanspruchung beliebig oft wiederholt werden kann, ohne zu einem Versagen zu führen, während die Zeitfestigkeit den Bereich beschreibt, in dem eine endliche Schwingspielzahl zu einem Versagen führt, das umso früher eintritt, je höher die Beanspruchung ist.

3.2 Wöhlerversuch

Der Wöhlerversuch beschreibt ein Verfahren zur experimentellen Ermittlung der Zeit- und Dauerfestigkeit von Bauteilen und ist in DIN 50 100 als Dauerschwing-Versuch genormt [16, S. 23]. Die Versuchskörper werden dabei schwingend, zumeist bei einer sinusförmigen Beanspruchung über die Zeit, bei gleicher Lastamplitude S und konstanter Ober- und Unterspannung belastet, wie in Abbildung 3 dargestellt.

Zur Ermittlung der sogenannten Wöhlerlinie werden mehrere baugleiche Versuchskörper auf unterschiedlichen Lasthorizonten beziehungsweise bei verschiedenen Lastamplituden bis zu ihrem jeweiligen Versagen oder bis zum Erreichen einer festgelegten maximalen Schwingspielzahl geprüft. Die erreichte Anzahl an Schwingspielen wird mit N bezeichnet.

Für die Darstellung der an die Ergebnisse angepassten Wöhlerlinie (engl. *SN curve*) existieren viele unterschiedliche Formeln. Wöhler verwendete in seiner ursprünglichen Forma-

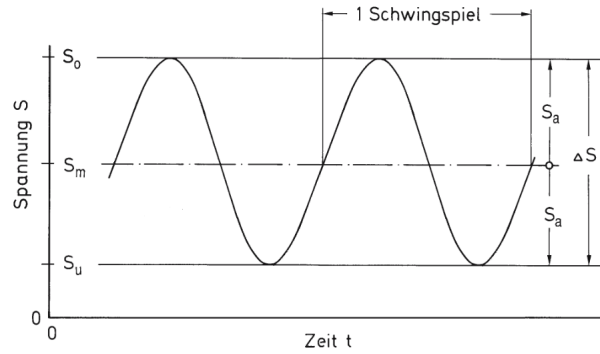


Abb. 3: Kennwerte zur Beschreibung eines Schwingenspiels bei Unter-, Mittel- und Oberspannungen S_u , S_m und S_o und Lastamplitude ΔS nach Haibach [16, S. 22, Abb. 2.1-1, angepasst].

lisierung im Jahre 1870 nur eine Logarithmierung der Lebensdauer:

$$\log N = a - b \cdot S. \tag{3.1}$$

Heutzutage hat sich hingegen die Formel von Basquin aus dem Jahre 1910 im doppelt-logarithmischen Netz durchgesetzt [16, S. 25]:

$$\log N = a - b \cdot \log S. \tag{3.2}$$

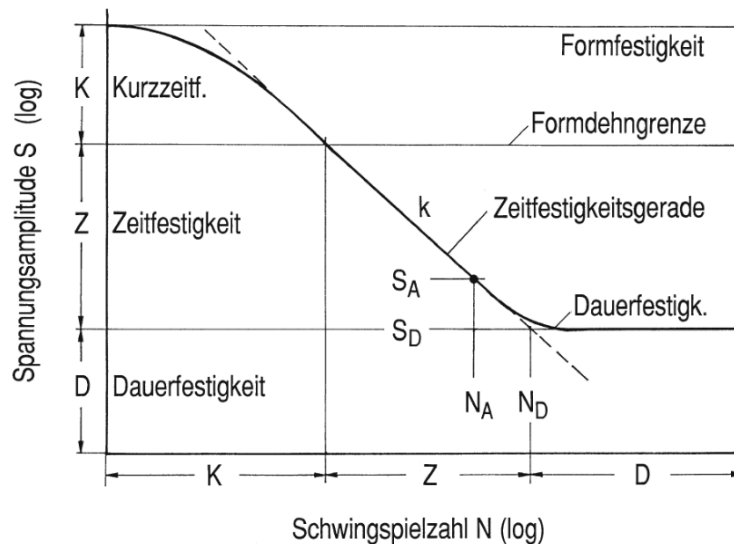


Abb. 4: Kennwerte zur Beschreibung einer Wöhlerlinie und die Bereiche Dauerfestigkeit (D), Zeitfestigkeit (Z) und Kurzzeitfestigkeit (K) nach Haibach [16, S. 26, Abb. 2.1-6].

Wenn S_C und N_C die jeweiligen Achsenwerte beim Abknickpunkt zwischen Zeit- und Dauerfestigkeit beschreiben, lässt sich die Formel zu

$$N = N_C \cdot (S/S_C)^{-k} \quad \text{für } S \geq S_C \quad (3.3)$$

umformen, wobei k die charakteristische Wöhlersteigung ist. In Abbildung 4 sind die Kennwerte zur Beschreibung der Wöhlerlinie nach Haibach [16, S. 26, Abb. 2.1-6] dargestellt. In der Regel wird dabei nur der Zeitfestigkeitsbereich betrachtet, in dem die tatsächliche Festigkeit eines Bauteils als linear betrachtet und durch die Zeitfestigkeitsgerade angenähert wird.

3.3 Kombiniertes Wöhlermodell

3.3.1 Herleitung

Da sich in der Praxis gezeigt hat, dass sich nicht alle Wöhlerversuche im Übergangsbereich zwischen Zeit- und Dauerfestigkeit durch das lineare Gesetz ausreichend beschreiben lassen, wurde am Fraunhofer Institut für Techno- und Wirtschaftsmathematik in Kaiserslautern ein kombiniertes Modell entwickelt [13]. Dieses stückweise lineare Modell wird durch zwei Geraden im Zeitfestigkeitsbereich beschrieben, wobei die untere der beiden im Übergang zur Dauerfestigkeit flacher verläuft, um die tatsächliche Festigkeit des Bauteils in diesem Bereich besser zu beschreiben.

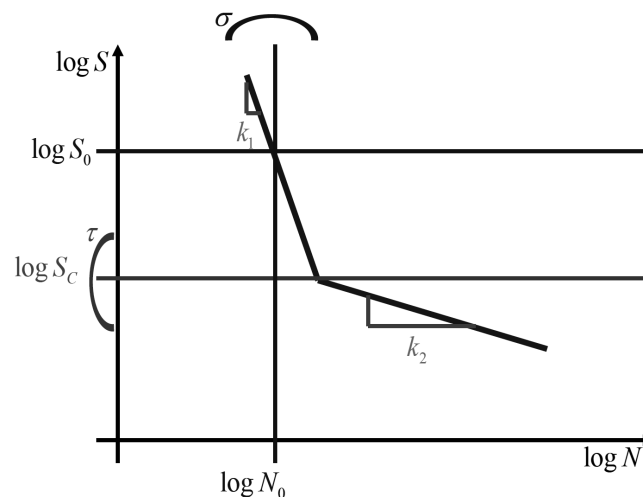


Abb. 5: Schematische Darstellung der Parameter im kombinierten Wöhlermodell: k_1 und k_2 sind die Steigungen der stückweise linearen Wöhlerlinie, die den Zusammenhang von Lastamplitude S und Schwingungszahl N abbildet, S_C ist der Abknickpunkt und N_0 bzw. S_0 ein beliebiger Referenzpunkt.

3.3.2 Formalisierung

Um das maximale Modell mathematisch zu formulieren, werden die folgenden sieben Parameter verwendet, wobei das Referenzniveau S_0 beliebig festgesetzt werden kann (typischerweise das höchste Lastniveau der betrachteten Stichprobe) und sich N_0 und σ jeweils auf die obere Wöhlerlinie beziehen. Alle Parameter müssen positiv sein und es wird davon ausgegangen, dass $k_2 > k_1$ gilt.

S_0	Referenzniveau	
N_0	mittlere Lebensdauer auf dem Referenzniveau	
σ	Standardabweichung der Log-Lebensdauer auf dem Referenzniveau	
S_C	mittlere Lage des Abknickpunkts im Lastbereich	(3.4)
τ	Standardabweichung des Abknickpunktes im Log-Lastbereich	
k_1	Steigung der oberen Wöhlerlinie	
k_2	Steigung der unteren Wöhlerlinie	

Durch dieses Modell wird eine Wöhlerlinie mit Abknickpunkt beschrieben, welcher für jedes Bauteil unterschiedlich sein kann. Das Modell einer einfachen Wöhlerlinie mit Dauerfestigkeitsbereich, wie es in der Literatur normalerweise beschrieben wird, lässt sich mit dem maximalen Modell durch den variablen Abknickpunkt und $k_2 = \infty$ als Grenzfall betrachten. Allgemein gilt

$$\log N = \begin{cases} \log N_0 + k_1(\log S_0 - \log S) & \text{falls } S_C \leq S \\ \log N_0 + k_1(\log S_0 - \log S_C) + k_2(\log S_C - \log S) & \text{falls } S_C > S. \end{cases} \quad (3.5)$$

3.3.3 Anpassung für gleiche Streuung in Amplitudenrichtung

Aus der Praxis ergibt sich die Forderung, dass die Streuung in Amplitudenrichtung über den gesamten Lebensdauerbereich gleich sein soll. Für die untere Wöhlerlinie ergibt sich die Streuung allerdings als Überlagerung der Streuung der Lebensdauer σ und der Streuung des Abknickpunktes τ . Dadurch ergibt sich als tatsächliche Streuung

$$\sqrt{\sigma^2 + (k_2 - k_1)^2 \tau^2}. \quad (3.6)$$

Unter der Voraussetzung, dass die Streuung in Amplitudenrichtung durchgängig gleich sein soll, lässt sich der Parameter τ für die tatsächliche Modellanpassung folgendermaßen einsparen:

$$\frac{\sigma}{k_1} = \frac{\sqrt{\sigma^2 + (k_2 - k_1)^2 \tau^2}}{k_2} \iff \tau = \frac{\sigma}{k_1} \sqrt{\frac{k_2 + k_1}{k_2 - k_1}}. \quad (3.7)$$

3.3.4 Modellierung einer Stichprobe

Für eine betrachtete Stichprobe $\underline{X} = (X_1, \dots, X_n)$ mit $X_i = (S_i, N_i, N_{0,i}, S_{C,i}, N_{C,i})$ werden folgende Notationen verwendet:

- n Anzahl der getesteten Bauteile
- i Index eines getesteten Bauteils, $i \in 1, \dots, n$
- S_i Testlast für Bauteil i
- N_i beobachtete Lebensdauer von Bauteil i (3.8)
- $N_{0,i}$ hypothetische Lebensdauer von Bauteil i auf dem Niveau S_0 bzgl. k_1
- $S_{C,i}$ Lage des Abknickpunkts für Bauteil i
- $N_{C,i}$ hypothetische Lebensdauer von Bauteil i auf dem Niveau $S_{C,i}$

$N_{0,i}$ ist dabei ein rein nomineller Wert, da die obere Wöhlerlinie auf dem Niveau S_0 nicht notwendig gilt. In der Regel sollte S_0 daher so groß gewählt werden, dass mit hoher Wahrscheinlichkeit $S_C < S_0$ gilt.

Die Abweichungen einzelner Beobachtungen werden mit ϵ und δ modelliert und als stochastisch unabhängig vorausgesetzt. Der Abknickpunkt S_C wird als normalverteilt angenommen, sodass sich für dessen Abweichungen normalverteilte Fehler ergeben:

$$\delta_i := \log S_{C,i} - \log S_C \sim \mathcal{N}(0, \tau^2). \quad (3.9)$$

Für die Lebensdauern werden zwei unterschiedliche Verteilungsannahmen betrachtet. Im Falle normalverteilter Lebensdauern liefern Maximum Likelihood und lineare Regression die gleichen Ergebnisse (ohne Beachtung von Durchläufern, siehe 2.5), was die Berechnungen deutlich vereinfacht. Dabei ergeben sich für die Schwingspielzahlen lognormalverteilte

Fehler:

$$\epsilon_i := \log N_{0,i} - \log N_0 \sim \mathcal{N}(0, \sigma^2). \quad (3.10)$$

Die Normalverteilung ist in der Regel aber unzureichend, um Lebensdauern zu modellieren, deren Ausfallwahrscheinlichkeit im Laufe der Zeit stetig zunimmt. Daher werden Weibull-verteilte Lebensdauern betrachtet, für die sich Gumbel-verteilte Fehler ergeben (siehe 2.3.3 bzw. A.0.2):

$$\epsilon_i := \log N_{0,i} - \log N_0 \sim \mathcal{G}(\beta, 0) \quad \text{mit } \sigma = \beta\pi/\sqrt{6}. \quad (3.11)$$

In beiden Fällen erhält man mit

$$\log N_{C,i} = \log N_0 + k_1(\log S_0 - \log S_{C,i}) \quad (3.12)$$

für $S_{C,i} \leq S_i$ beziehungsweise $\delta_i \leq \log S_i - \log S_C$

$$\begin{aligned} \log N_i &= \log N_{C,i} + k_1(\log S_{C,i} - \log S_i) + \epsilon_i \\ &= \log N_0 + k_1(\log S_0 - \log S_i) + \epsilon_i \\ &=: \mu_{i,1} + \epsilon_i \end{aligned} \quad (3.13)$$

und für $S_{C,i} > S_i$ beziehungsweise $\delta_i > \log S_i - \log S_C$

$$\begin{aligned} \log N_i &= \log N_{C,i} + k_2(\log S_{C,i} - \log S_i) + \epsilon_i \\ &= \log N_0 + k_1(\log S_0 - \log S_C) + k_2(\log S_C - \log S_i) + (k_2 - k_1)\delta_i + \epsilon_i \\ &=: \mu_{i,2} + (k_2 - k_1)\delta_i + \epsilon_i, \end{aligned} \quad (3.14)$$

wobei $\mu_{i,1}$ und $\mu_{i,2}$ den Erwartungswerten von $\log N_i$ bezüglich k_1 bzw. k_2 entsprechen.

3.4 Beispieldatensätze

Die unterschiedlichen Verfahren zur Berechnung von Konfidenzintervallen werden anhand zweier Beispieldatensätze vorgestellt und getestet. Dabei werden sowohl ein realer Datensatz, für den die wahren Modellparameter unbekannt sind, als auch ein fiktiver Datensatz, der für diese Arbeit aus vorgegebenen Modellparametern zufällig erstellt wurde, untersucht.

Für die Parameterschätzung wird im Folgenden durchgängig die Annahme normalverteilter Lebensdauern verwendet. Diese Konvention lässt die Betrachtung alternativer Verfahren zu, die auf linearer Regression basieren und daher (log-)normalverteilte Residuen verlangen. Die anderen Verfahren lassen sich ohne Weiteres für die Annahme Weibullverteilter Lebensdauern anpassen.

3.4.1 Beispiel A: Verdacht auf zwei Bereiche

Der erste Beispieldatensatz enthält reale Messwerte und liefert ein typisches Beispiel, bei dem zwei Bereiche mit unterschiedlichen Wöhlersteigungen vermutet werden. Die Stichprobe hat eine Größe von $n = 24$, wovon jeweils sechs Werte auf vier unterschiedlichen Lastniveaus liegen, und enthält auf dem untersten Lastniveau fünf zensierte Werte, bei denen die Beobachtung nach $3.75 \cdot 10^6$ Zyklen beendet wurde.

i	Last S_i [kN]	Lebensdauer N_i [Zyklen]	Zensurindex d_i
1	2720	$6.9000 \cdot 10^4$	1
2	2720	$7.3500 \cdot 10^4$	1
3	2720	$7.4250 \cdot 10^4$	1
4	2720	$7.6500 \cdot 10^4$	1
5	2720	$7.8000 \cdot 10^4$	1
6	2720	$1.0650 \cdot 10^5$	1
7	2400	$1.2375 \cdot 10^5$	1
8	2400	$1.4693 \cdot 10^5$	1
9	2400	$1.5225 \cdot 10^5$	1
10	2400	$1.6650 \cdot 10^5$	1
11	2400	$1.9575 \cdot 10^5$	1
12	2400	$2.1975 \cdot 10^5$	1
13	2080	$4.8825 \cdot 10^5$	1

i	Last S_i [kN]	Lebensdauer N_i [Zyklen]	Zensurindex d_i
14	2080	$5.4150 \cdot 10^5$	1
15	2080	$6.3525 \cdot 10^5$	1
16	2080	$1.0190 \cdot 10^6$	1
17	2080	$1.2098 \cdot 10^6$	1
18	2080	$1.5395 \cdot 10^6$	1
19	1760	$1.0058 \cdot 10^6$	1
20	1760	$3.7500 \cdot 10^6$	0
21	1760	$3.7500 \cdot 10^6$	0
22	1760	$3.7500 \cdot 10^6$	0
23	1760	$3.7500 \cdot 10^6$	0
24	1760	$3.7500 \cdot 10^6$	0

Tab. 1: Beispieldatensatz: Verdacht auf zwei Bereiche

In Abbildung 6 ist der Beispieldatensatz grafisch dargestellt. Besonders bei Berücksichtigung der Durchläufer auf dem untersten Lastniveau lässt sich vermuten, dass die Wöhlerlinie im Bereich in der unteren Hälfte des Diagramms deutlich flacher verläuft.

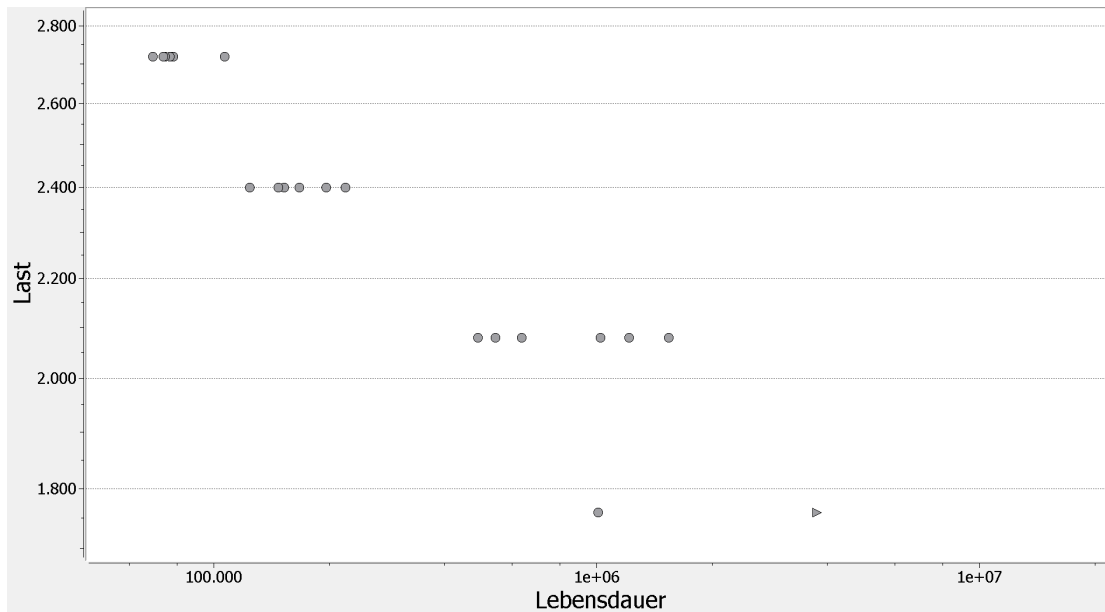


Abb. 6: Grafische Darstellung des Beispieldatensatzes „Verdacht auf zwei Bereiche“ - Die Durchläufer sind als Dreieck eingetragen und liegen alle übereinander.

3.4.2 Beispiel B: Fiktiver Beispieldatensatz

Der zweite Beispieldatensatz ist ein rein fiktiver Datensatz, der mit Hilfe von Zufallszahlen für beliebig gewählte Parametern erstellt wurde, um die berechneten Konfidenzintervalle mit den in diesem Fall bekannten wahren Parametern vergleichen zu können. Darüber hinaus ist für diesen Datensatz garantiert, dass das kombinierte Wöhlermodell den Daten tatsächlich zugrunde liegt.

Es wurde ein Datensatz mit je sechs Prüflingen auf acht unterschiedlichen Lastniveaus (130, 150, 170, 190, 210, 230, 250, 270) erstellt. Der Stichprobenumfang beträgt somit $n = 48$. Die Wöhlersteigungen wurden als $k_1 = 5$ und $k_2 = 10$, das Referenzniveau als $S_0 = 270$ mit $N_0 = 10^5$, der Abknickpunkt als $S_C = 200$ und die Streuung als $\sigma = 0.1$ festgelegt. Gemäß 3.3.3 ergibt sich daraus $\tau = 0.034641$. Bei der so generierten Stichprobe wurden alle Werte oberhalb von $2 \cdot 10^7$ zensiert.

Mit Hilfe von Matlab (siehe Listing 1) wurde nach diesen Vorgaben die in Tabelle 2 dargestellte Stichprobe zufällig erzeugt. In dieser liegen auf dem untersten Lastniveau vier zensierte Werte vor.

i	Last S_i [kN]	Lebensdauer N_i [Zyklen]	Zensurindex d_i
1	270	$9.6097 \cdot 10^4$	1
2	270	$7.5707 \cdot 10^4$	1
3	270	$9.3387 \cdot 10^4$	1
4	270	$4.7511 \cdot 10^4$	1
5	270	$7.7858 \cdot 10^4$	1
6	270	$7.2004 \cdot 10^4$	1
7	250	$1.1633 \cdot 10^5$	1
8	250	$1.3989 \cdot 10^5$	1
9	250	$1.3633 \cdot 10^5$	1
10	250	$2.2991 \cdot 10^5$	1
11	250	$1.2881 \cdot 10^5$	1
12	250	$1.3871 \cdot 10^5$	1
13	230	$1.5533 \cdot 10^5$	1
14	230	$1.9973 \cdot 10^5$	1
15	230	$1.6383 \cdot 10^5$	1
16	230	$2.2450 \cdot 10^5$	1
17	230	$2.7132 \cdot 10^5$	1

i	Last S_i [kN]	Lebensdauer N_i [Zyklen]	Zensurindex d_i
18	230	$2.4468 \cdot 10^5$	1
19	210	$5.0151 \cdot 10^5$	1
20	210	$2.4136 \cdot 10^5$	1
21	210	$5.8496 \cdot 10^5$	1
22	210	$5.6328 \cdot 10^5$	1
23	210	$3.6122 \cdot 10^5$	1
24	210	$2.8861 \cdot 10^5$	1
25	190	$1.2316 \cdot 10^6$	1
26	190	$5.0558 \cdot 10^5$	1
27	190	$5.2008 \cdot 10^5$	1
28	190	$7.1070 \cdot 10^5$	1
29	190	$4.2124 \cdot 10^5$	1
30	190	$4.8874 \cdot 10^5$	1
31	170	$2.1297 \cdot 10^6$	1
32	170	$2.1568 \cdot 10^6$	1
33	170	$1.3529 \cdot 10^6$	1
34	170	$3.4848 \cdot 10^6$	1
35	170	$3.4819 \cdot 10^6$	1
36	170	$2.7479 \cdot 10^6$	1
37	150	$1.0497 \cdot 10^7$	1
38	150	$6.7714 \cdot 10^6$	1
39	150	$9.7771 \cdot 10^6$	1
40	150	$8.1443 \cdot 10^6$	1
41	150	$6.9537 \cdot 10^6$	1
42	150	$6.9095 \cdot 10^6$	1
43	130	$2.0000 \cdot 10^7$	0
44	130	$2.0000 \cdot 10^7$	0
45	130	$1.9408 \cdot 10^7$	1
46	130	$1.4424 \cdot 10^7$	1
47	130	$2.0000 \cdot 10^7$	0
48	130	$2.0000 \cdot 10^7$	0

Tab. 2: Fiktiver Datensatz mit normal-verteilten Abweichungen für bekannte Parameter ($k_1 = 5$, $k_2 = 10$, $N_0 = 10^5$, $S_C = 200$, $\sigma = 0.1$, $\tau = 0.034641$)

In Abbildung 7 ist der Datensatz grafisch dargestellt. Im Gegensatz zum ersten Beispiel sind deutlich mehr Daten vorhanden und die Streuung im unteren Bereich ist etwas geringer. Auch für diesen Datensatz lässt sich gut erkennen, dass eine mögliche Ausgleichsgerade für Lasten unterhalb von 200 kN deutlich flacher verläuft, wobei der Unterschied der beiden Steigungen etwas kleiner ist als beim ersten Beispieldatensatz.

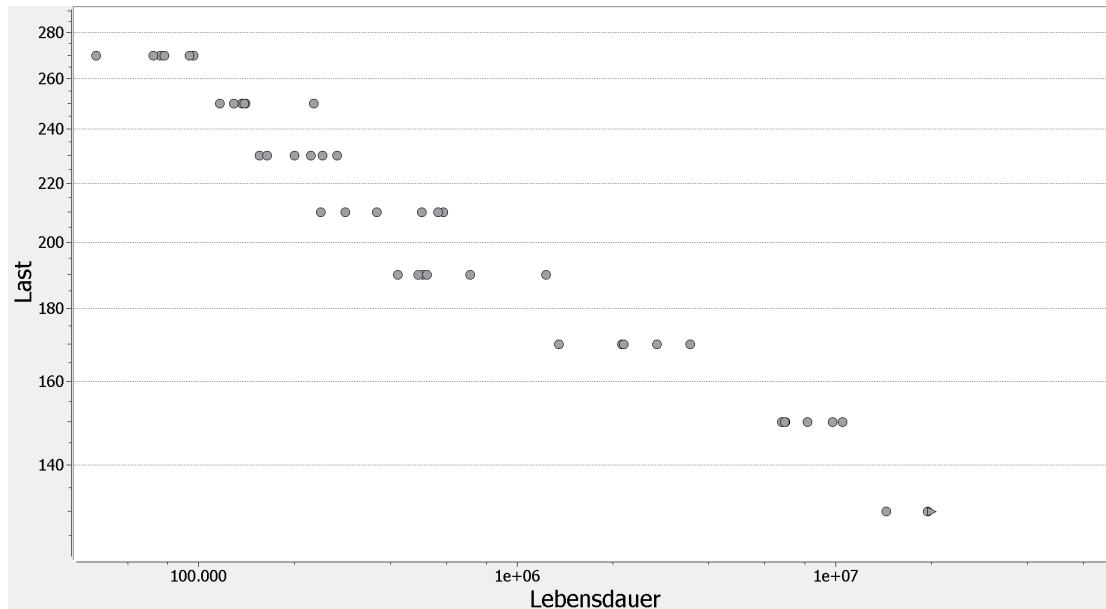


Abb. 7: Grafische Darstellung des Fiktiven Beispieldatensatzes - Die Durchläufer sind als Dreieck eingetragen und liegen alle übereinander.

4 Parameterschätzung mit Maximum-Likelihood

Zur Anwendung des Maximum-Likelihood-Verfahrens ist es notwendig, die Likelihood-Funktion des zugrundeliegenden Modells zu bestimmen. Diese ergibt sich nach (2.14) als

$$\ell(\theta | \underline{X}) = \sum_{i \in I_A} \ln f_\theta(n | X_i) + \sum_{i \in I_D} \ln (1 - F_\theta(n | X_i)). \quad (4.1)$$

Im Folgenden wird jeweils auf die Angabe der Abhängigkeit von X_i verzichtet, also zum Beispiel $f_\theta(n)$ anstelle von $f_\theta(n|X_i)$ geschrieben.

4.1 Likelihood-Funktion für normalverteilte Lebensdauern

Für eine Stichprobe des kombinierten Wöhlermodells (siehe 3.3.4) ergibt sich die Verteilungsfunktion wegen $\delta_i \sim \mathcal{N}(0, \tau^2)$ als

$$\begin{aligned} F_\theta(n) &= P(\log N_i \leq n) \\ &= P(\log N_i \leq n | \delta_i \leq \log S_i - \log S_C) \cdot P(\delta_i \leq \log S_i - \log S_C) \\ &\quad + P(\log N_i \leq n | \delta_i > \log S_i - \log S_C) \cdot P(\delta_i > \log S_i - \log S_C) \\ &= P(\epsilon_i \leq n - \mu_{i,1}) \cdot \Phi\left(\frac{\log S_i - \log S_C}{\tau}\right) \\ &\quad + A(n) \cdot \left(1 - \Phi\left(\frac{\log S_i - \log S_C}{\tau}\right)\right) \end{aligned} \quad (4.2)$$

mit

$$\begin{aligned} A(n) &= P(\log N_i \leq n | \delta_i > \log S_i - \log S_C) \\ &= P(\mu_{i,2} + (k_2 - k_1)\delta_i + \epsilon_i \leq n | \delta_i > \log S_i - \log S_C). \end{aligned} \quad (4.3)$$

Für normalverteilte Lebensdauern mit $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ gilt in (4.2)

$$P(\epsilon_i \leq n - \mu_{i,1}) = \Phi\left(\frac{n - \mu_{i,1}}{\sigma}\right) \quad (4.4)$$

und $A(n)$ ergibt sich als Integral der Faltung der Dichten f_1 und f_2

$$f_1(n) = \frac{1}{\sigma} \phi\left(\frac{n - \mu_{i,2}}{\sigma}\right) \quad (4.5)$$

$$\begin{aligned} f_2(n) &= \frac{1}{(k_2 - k_1)\tau \left(1 - \Phi\left(\frac{(k_2 - k_1)(\log S_i - \log S_C)}{(k_2 - k_1)\tau}\right)\right)} \phi\left(\frac{n}{(k_2 - k_1)\tau}\right) \mathbb{1}_{((k_2 - k_1)(\log S_i - \log S_C), \infty)}(n) \\ &= \frac{1}{(k_2 - k_1)\tau \left(1 - \Phi\left(\frac{\log S_i - \log S_C}{\tau}\right)\right)} \phi\left(\frac{n}{(k_2 - k_1)\tau}\right) \mathbb{1}_{((k_2 - k_1)(\log S_i - \log S_C), \infty)}(n) \end{aligned} \quad (4.6)$$

als

$$\begin{aligned} A(n) &= \int_{-\infty}^n \int_{-\infty}^{\infty} f_1(u - v) f_2(v) \, dv \, du \\ &= \frac{1}{1 - \Phi\left(\frac{\log S_i - \log S_C}{\tau}\right)} \\ &\quad \cdot \int_{-\infty}^n \int_{(k_2 - k_1)(\log S_i - \log S_C)}^{\infty} \frac{1}{\sigma} \phi\left(\frac{u - v - \mu_{i,2}}{\sigma}\right) \frac{1}{(k_2 - k_1)\tau} \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) \, dv \, du \\ &= \frac{1}{1 - \Phi\left(\frac{\log S_i - \log S_C}{\tau}\right)} \\ &\quad \cdot \int_{(k_2 - k_1)(\log S_i - \log S_C)}^{\infty} \Phi\left(\frac{n - v - \mu_{i,2}}{\sigma}\right) \frac{1}{(k_2 - k_1)\tau} \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) \, dv. \end{aligned} \quad (4.7)$$

Durch Einsetzen von (4.7) in (4.2) und Kürzen ergeben sich

$$\begin{aligned} F_{\theta}(n) &= \Phi\left(\frac{n - \mu_{i,1}}{\sigma}\right) \Phi\left(\frac{\log S_i - \log S_C}{\tau}\right) \\ &\quad + \int_{(k_2 - k_1)(\log S_i - \log S_C)}^{\infty} \Phi\left(\frac{n - v - \mu_{i,2}}{\sigma}\right) \frac{1}{(k_2 - k_1)\tau} \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) \, dv \quad \text{und} \end{aligned} \quad (4.8)$$

$$\begin{aligned} f_{\theta}(n) &= \frac{d}{dn} F_{\theta}(n) = \frac{1}{\sigma} \phi\left(\frac{n - \mu_{i,1}}{\sigma}\right) \Phi\left(\frac{\log S_i - \log S_C}{\tau}\right) \\ &\quad + \int_{(k_2 - k_1)(\log S_i - \log S_C)}^{\infty} \frac{1}{\sigma} \phi\left(\frac{n - v - \mu_{i,2}}{\sigma}\right) \frac{1}{(k_2 - k_1)\tau} \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) \, dv. \end{aligned} \quad (4.9)$$

4.2 Likelihood-Funktion für Weibull-verteilte Lebensdauern

Für Weibull-verteilte Lebensdauern mit $\epsilon_i \sim \mathcal{G}(\beta, 0)$ mit $\beta = \sigma\sqrt{6}/\pi$ gelten (4.2), (4.3) und (4.6) unverändert. Für die von ϵ_i abhängigen Gleichungen ergeben sich hingegen

$$P(\epsilon_i \leq n - \mu_{i,1}) = e^{-e^{-\frac{1}{\beta}(n-\mu_{i,1})}}, \quad f_1(n) = \frac{1}{\beta} e^{-\frac{1}{\beta}(n-\mu_{i,1})} e^{-e^{-\frac{1}{\beta}(n-\mu_{i,1})}} \quad \text{und} \quad (4.10)$$

$$A(n) = \frac{1}{1 - \Phi\left(\frac{\log S_i - \log S_C}{\tau}\right)} \cdot \int_{(k_2-k_1)(\log S_i - \log S_C)}^{\infty} e^{-e^{-1/\beta(n-v-\mu_{i,2})}} \frac{1}{(k_2 - k_1)\tau} \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) dv. \quad (4.11)$$

Durch Einsetzen von (4.11) in (4.2) und Kürzen ergeben sich

$$F_\theta(n) = e^{-e^{-\frac{1}{\beta}(n-\mu_{i,1})}} \Phi\left(\frac{\log S_i - \log S_C}{\tau}\right) + \int_{(k_2-k_1)(\log S_i - \log S_C)}^{\infty} e^{-e^{-\frac{1}{\beta}(n-v-\mu_{i,2})}} \frac{1}{(k_2 - k_1)\tau} \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) dv \quad \text{und} \quad (4.12)$$

$$f_\theta(n) = \frac{d}{dn} F_\theta(n) = \frac{1}{\beta} e^{-\frac{1}{\beta}(n-\mu_{i,1})} e^{-e^{-\frac{1}{\beta}(n-\mu_{i,1})}} \Phi\left(\frac{\log S_i - \log S_C}{\tau}\right) + \int_{(k_2-k_1)(\log S_i - \log S_C)}^{\infty} \frac{1}{\beta} e^{-\frac{1}{\beta}(n-v-\mu_{i,2})} e^{-e^{-\frac{1}{\beta}(n-v-\mu_{i,2})}} \frac{1}{(k_2 - k_1)\tau} \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) dv. \quad (4.13)$$

4.3 Schätzung der Modellparameter

Mit der Log-Likelihood-Funktion (4.1) lassen sich für das vorgestellte Wählermodell aus einer gegebenen Stichprobe die Modellparameter N_0 , σ , S_C , τ , k_1 und k_2 durch das Maximum-Likelihood-Verfahren schätzen. Zur Maximierung muss dabei auf numerische Methoden zurückgegriffen werden, da keine analytische Lösung bekannt ist.

Die Implementierung innerhalb des am Fraunhofer ITWM entwickelten Software-Pakets *Jurojin*¹ verwendet dabei aus der *Ipopt*-Bibliothek² ein nichtlineares Optimierungsverfah-

¹<http://www.itwm.fraunhofer.de/jurojin>

²<https://projects.coin-or.org/Ipopt>

ren mit Nebenbedingungen (Innere-Punkte-Verfahren). Für dieses werden für alle Parameter plausible untere und obere Grenzen verwendet. Die Messungen der Berechnungsdauer wurden auf einem Windows Server 2008 (64-bit, 2.80 GHz, 32 GB RAM) im Mehrbenutzerbetrieb vorgenommen.

In der Praxis haben sich aufgrund der Komplexität der Likelihood-Funktion Schwierigkeiten bezüglich der numerischen Stabilität und der Konvergenzrate gezeigt. Aus diesen Gründen wurde die Implementierung des Verfahrens am Fraunhofer ITWM weitergehend verfeinert. So wird die Streuung σ iterativ solange vergrößert, bis die Optimierung einen Wert deutlich unterhalb der aktuellen Schranke liefert, und der Abknickpunkt S_C wird auf einem ausreichend feinen Raster fixiert und innerhalb der Optimierung als Konstante betrachtet [13].

4.3.1 Beispiel A

Für den Beispieldatensatz „Verdacht auf zwei Bereiche“ liefert die Schätzung der Modellparameter die in Tabelle 3 dargestellten Werte, wobei sich τ wie in 3.3.3 erwähnt aus σ ableitet und sich N_0 auf das oberste Lastniveau $S_0 = 2720$ bezieht.

k_1	$\log N_0$	σ	S_C	τ	k_2
5.35763	4.9045	0.10152	2225.88	0.02871	13.6215

Tab. 3: Geschätzte Modellparameter für den Beispieldatensatz „Verdacht auf zwei Bereiche“

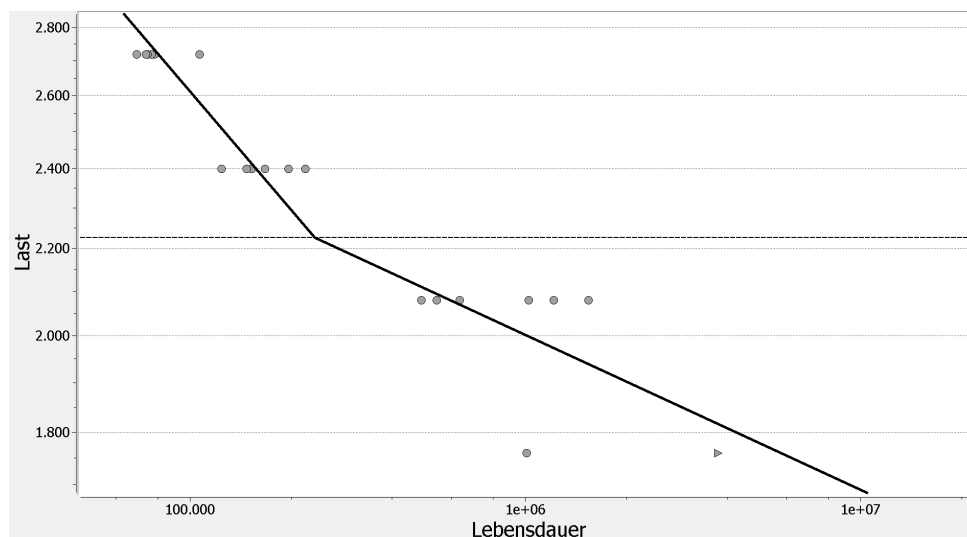


Abb. 8: Grafische Darstellung der geschätzten Wöhlerlinie und des zugrundeliegenden Datensatzes „Verdacht auf zwei Bereiche“. Ausfälle sind als Kreis, Durchläufer als Dreieck dargestellt, wobei die fünf Durchläufer alle übereinander liegen.

Die geschätzte Wöhlerlinie ist in Abbildung 8 dargestellt. Wie erwartet knickt die Wöhlerlinie zwischen den beiden mittleren Lastniveaus ab und verläuft im unteren Bereich deutlich flacher. Dabei ist gut zu erkennen, dass das Modell die zensierten Werte berücksichtigt hat, da die Wöhlerlinie auf dem niedrigsten Lastniveau deutlich rechts von den beobachteten Durchläufern liegt.

4.3.2 Beispiel B

Für den fiktiven Beispieldatensatz liefert die Schätzung die in Tabelle 4 dargestellten Parameter. In diesem Fall führen die Durchläufer auf dem untersten Lastniveau dazu, dass k_2 merklich unterhalb des wahren Wertes liegt, während die anderen Parameter nicht deutlich vom jeweils wahren Wert abweichen.

	k_1	$\log N_0$	σ	S_C	τ	k_2
Schätzung	5.46486	4.9216	0.10246	212.76	0.03854	8.8531
wahrer Wert	5.00	5.00	0.100	200.00	0.03464	10.00

Tab. 4: Geschätzte Modellparameter für den Fiktiven Beispieldatensatz

Die resultierende Wöhlerlinie ist in Abbildung 9 zusammen mit dem Datensatz dargestellt. Da die Stichprobe recht groß ist, passt die Wöhlerlinie sehr gut zu den Daten. Die zu den wahren Parametern gehörende Wöhlerlinie verläuft bei etwas niedriger liegendem Abknickpunkt im oberen Bereich etwas steiler, im unteren Bereich etwas flacher.

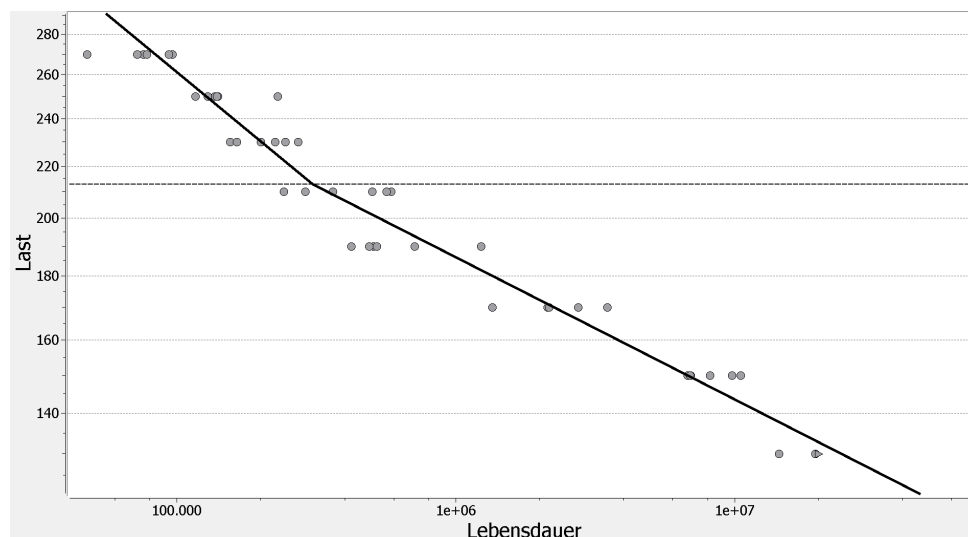


Abb. 9: Grafische Darstellung der geschätzten Wöhlerlinie und des zugrundeliegenden fiktiven Datensatzes. Ausfälle sind als Kreis, Durchläufer als Dreieck dargestellt, wobei die vier Durchläufer alle übereinander liegen.

5 Konfidenzintervalle mit Bootstrap

Um die Zuverlässigkeit der geschätzten Parameter angeben zu können, sind Konfidenzintervalle für die einzelnen Parameter gesucht. Im Idealfall würde man unzählige Stichproben erheben und könnte daraus die Verteilung der Parameter bestimmen und damit Konfidenzintervalle angeben. In der Praxis steht aber nur eine einzige Stichprobe limitierten Umfangs zur Verfügung.

Eine Möglichkeit, aus dieser einzelnen Stichprobe eine Information über die Verteilung der Parameter zu erhalten, bietet das Bootstrap-Verfahren. Beim (nichtparametrischen) Bootstrap-Verfahren werden dazu, vereinfacht gesagt, B neue Stichproben durch Ziehen mit Zurücklegen aus der ursprünglichen Stichprobe erzeugt. Aus den insgesamt B Bootstrap-Stichproben werden insgesamt B Schätzwerte bestimmt, mithilfe derer etwas über die Verteilung des Schätzers gesagt werden kann.

5.1 Bootstrap-Verfahren

Sei $\underline{X} = (X_1, \dots, X_n)$ eine Stichprobe mit $X_1, \dots, X_n \stackrel{iid}{\sim} F$ und $T_F(X)$ eine Statistik mit $T_F \sim G$, wobei F und G unbekannt sind. Um G zu approximieren, wird die empirische Verteilung \hat{G} der Schätzer $T_i^* := T_{F^*}(\underline{X}_i^*)$ bestimmt, wobei $\underline{X}_i^* = (X_{i,1}^*, \dots, X_{i,n}^*)$ mit $X_{i,j}^* \sim F^* \approx F$ die sogenannten Bootstrap-Stichproben sind. Beim nichtparametrischen Bootstrap-Verfahren wird dafür die empirische Verteilungsfunktion $F^* := \hat{F}$ verwendet. [22, S. 96]

Die Monte-Carlo-Methode kann verwendet werden, um aus F^* insgesamt B Bootstrap-Stichproben $\underline{X}_1^*, \dots, \underline{X}_B^*$ zu erzeugen und ihre Schätzwerte $T_i^* = T_{F^*}(\underline{X}_i^*)$ zu berechnen. Aus dem Gesetz der großen Zahlen ergibt sich, dass der Erwartungswert von T über das arithmetische Mittel geschätzt werden kann:

$$E[T] \approx E[T^*] = \frac{1}{B} \sum_{i=1}^B T_i^* \quad (5.1)$$

Im Falle des kombinierten Wählermodells werden also für jede Bootstrap-Stichprobe \underline{X}_i^* die Modellparameter mittels Maximum Likelihood geschätzt und anschließend die empirische Verteilung der geschätzten Parameter betrachtet. Aus dieser lassen sich Konfidenzintervalle für den Schätzer bestimmen.

5.2 Herleitung der Konfidenzintervalle

5.2.1 Bootstrap-Quantil-Methode

Die Bootstrap-Quantil-Methode verwendet die geordneten Schätzwerte $T_{(1)}^*, \dots, T_{(B)}^*$, um ein approximatives Konfidenzintervall für den Schätzer anzugeben. Im allgemeinen Fall erhält man durch die Bestimmung der beiden Quantile $q_{1-\alpha/2}^*$ und $q_{\alpha/2}^*$ aus der empirischen Verteilungsfunktion \hat{F}_{T^*} das folgende zweiseitige α -Konfidenzintervall für T^* , welches das entsprechende Konfidenzintervall für T approximiert [10, S. 170]:

$$\left[q_{1-\alpha/2}^*, q_{\alpha/2}^* \right] \quad (5.2)$$

5.2.2 Bootstrap-Prinzip

Eine weitere Möglichkeit bietet das Bootstrap-Prinzip, das ebenfalls die Quantile der Bootstrap-Schätzwerte T^* nutzt. Um das Konfidenzintervall unabhängiger von den speziellen Eigenschaften der geschätzten Verteilung zu machen, wird der Schätzwert \hat{T} der Originalstichprobe als Pivot zur Korrektur verwendet [7, S. 194]:

$$\left[2\hat{T} - q_{\alpha/2}^*, 2\hat{T} - q_{1-\alpha/2}^* \right] \quad (5.3)$$

Das Bootstrap-Prinzip wird oft in der Literatur bevorzugt [12], kann aber (ohne zusätzliche Korrektur durch Logarithmieren) dazu führen, dass für strikt positive Parameter eine negative untere Grenze bestimmt wird. Daher ist im Einzelfall zu prüfen, welche Methode zur Schätzung der Konfidenzintervalle besser geeignet ist.

Da sich in der Anwendung keine Probleme bezüglich der strikt positiven Parameter ergeben haben, wurde im Folgenden das Bootstrap-Prinzip verwendet. Auf die Rechenzeit hat die Wahl der konkreten Bootstrap-Methode in diesem Fall ohnehin keinen Einfluss.

5.3 Anwendung für Beispieldatensätze

5.3.1 Beispiel A

Tabelle 5 enthält die für den Datensatz „Verdacht auf zwei Bereiche“ mit dem Bootstrap-Prinzip berechneten 95%-Konfidenzintervalle für unterschiedlich viele Bootstrap-Stichpro-

ben. Die Berechnung hat dabei für $B = 200$ insgesamt $16.512s$ (ca. 4 1/2 Stunden) und für $B = 2000$ insgesamt $161.448s$ (ca. 45 Stunden) gedauert. Die Unterschiede zwischen den Ergebnissen sind dabei trotz zehnmal mehr Bootstrap-Stichproben und damit auch einer zehnmal so langen Berechnungsdauer sehr gering.

	k_1	$\log N_0$	σ	S_C	k_2
B = 200	[2.57, 5.51]	[4.88, 5.02]	[0.086, 0.161]	[2158, 2372]	[12.77, 15.42]
B = 2000	[2.60, 5.34]	[4.86, 5.01]	[0.087, 0.156]	[2158, 2356]	[13.05, 15.59]

Tab. 5: 95%-Konfidenzintervalle für Modellparameter des Datensatzes „Verdacht auf zwei Bereiche“ für unterschiedlich viele Bootstrap-Stichproben ($B = 2000$ bzw. $B = 200$).

Dabei fällt auf, dass die Konfidenzintervalle nicht symmetrisch sind, insbesondere für die Steigungen der Wählerlinie und den Abknickpunkt, deren jeweilige Schätzwerte eher am Rand der Intervalle liegen ($\hat{k}_1 = 5.38$, $\hat{S}_C = 2225$ und $\hat{k}_2 = 13.62$). Dieser Umstand lässt sich beim Blick auf die grafische Darstellung der ursprünglichen Schätzung (vgl. Abbildung 8) dadurch erklären, dass je nach Zusammensetzung der erzeugten Bootstrap-Stichprobe die erste Wählerlinie etwas flacher und dadurch der Abknickpunkt niedriger und die zweite Wählerlinie etwas steiler geschätzt wird, nicht aber umgekehrt.

5.3.2 Beispiel B

Für den Datensatz „Fiktiver Beispieldatensatz“ ergeben sich die in Tabelle 6 angegebenen 95%-Konfidenzintervalle. Für $B = 2000$ dauert die Berechnung $88.884s$ (ca. 24 1/2 Stunden), für $B = 200$ mit $8.916s$ (ca. 2 1/2 Stunden) und für $B = 5000$ mit $224.532s$ (ca. 62 1/2 Stunden) entsprechend kürzer beziehungsweise länger. Bei einer Stichprobengröße von $n = 48$ im Vergleich zu $n = 24$ bei Beispiel A ist die Optimierung innerhalb des Maximum-Likelihood-Verfahrens etwa doppelt so schnell und die Konfidenzintervalle sind deutlich kleiner. Ebenso wie bei Beispiel A sind die Konfidenzintervalle nicht symmetrisch.

	k_1	$\log N_0$	σ	S_C	k_2
B = 200	[4.96, 5.46]	[4.91, 4.99]	[0.102, 0.137]	[189, 218]	[8.85, 9.73]
B = 2000	[4.99, 5.46]	[4.91, 5.00]	[0.102, 0.138]	[189, 218]	[8.85, 9.75]
B = 5000	[5.01, 5.46]	[4.91, 5.00]	[0.102, 0.139]	[189, 218]	[8.85, 9.74]
θ_i enthalten	ja / ja / nein	nein / ja / ja	nein	ja	nein

Tab. 6: 95%-Konfidenzintervalle für Modellparameter des fiktiven Beispieldatensatzes für unterschiedlich viele Bootstrap-Stichproben. Maximal drei der wahren Parameter sind im jeweiligen Konfidenzintervall enthalten.

Da für diesen fiktiven Datensatz die wahren Parameter, mit denen die Stichprobe zufällig erzeugt wurde, bekannt sind, lässt sich überprüfen, ob der jeweils wahre Parameter innerhalb des Konfidenzintervalls liegt. Das ist nur für $\sigma = 0.1$ und $k_2 = 10$ nicht der Fall. Allerdings liegen $k_1 = 5$ und $\log N_0 = 5$ jeweils sehr dicht am Rand des Konfidenzintervalls. Das bedeutet allerdings nicht, dass das Bootstrap-Verfahren schlecht ist, da ein Konfidenzintervall nicht zwangsläufig den wahren Parameter überdecken muss. In diesem Fall lässt der Datensatz lediglich keine bessere Schätzung zu.

5.4 Modifikationen zur Verringerung der Rechenzeit

Der Nachteil des Bootstrap-Verfahrens für das kombinierte Wählermodell ist, dass die einzelne Parameterschätzung, die innerhalb jeder Bootstrap-Schleife einmal und damit insgesamt B -mal durchgeführt werden muss, aufgrund der Komplexität der Log-Likelihood-Funktion sehr lange dauert. Dies führt, je nach Geschwindigkeit des verwendeten Computers und der Anzahl gewünschter Bootstrap-Stichproben, zu einem Zeitaufwand von insgesamt mehreren Stunden bis mehreren Tagen. Um den Zeitaufwand zu verringern, wird zunächst versucht, das Bootstrap-Verfahren zu modifizieren.

Eine Modifikation beziehungsweise ein Spezialfall des Bootstrap-Verfahrens ist *Jackknife*, das lediglich so viele Parameterschätzungen verwendet, wie Werte in der originalen Stichprobe vorliegen, und dadurch entsprechend schneller ist.

Eine andere Möglichkeit, das Bootstrap-Verfahren zu beschleunigen, könnte es sein, das Maximum-Likelihood-Verfahren zur Parameterschätzung innerhalb der Bootstrap-Schleife durch eine einfachere und dadurch schnellere, wenn auch voraussichtlich ungenauere Parameterschätzung wie die lineare Regression zu ersetzen. Für die Parameterschätzung selbst wurden solche Verfahren bisher nicht betrachtet, da im Gegensatz zu diesen das Maximum-Likelihood-Verfahren problemlos mit beliebig komplexen Modellen umgehen kann. Sowohl für die Berücksichtigung zensierter Daten als auch für segmentierte lineare Modelle existieren Verfahren, um das entsprechende Regressionsmodell zu lösen.

Da das nichtparametrische Bootstrap nicht auf den originalen Schätzwerten $\hat{\theta}_i$ basiert (im Gegensatz zum parametrischen Bootstrap, das diese Werte zum Erzeugen der Bootstrap-Stichproben verwendet), ist es theoretisch möglich, das Maximum-Likelihood-Verfahren zur Schätzung von $\hat{\theta}$ und ein anderes Verfahren zur Schätzung der Konfidenzintervalle zu verwenden, ohne gegen die prinzipielle Vorgehensweise beim Bootstrap-Verfahren zu verstoßen. Es ist aber davon auszugehen, dass die Ergebnisse in dem Fall nicht ganz so präzise wie bei der Verwendung von Maximum Likelihood sind.

5.4.1 Jackknife

Das *Jackknife*, in diesem Fall das *delete-1-Jackknife*, ist sowohl ein Vorläufer als auch ein Spezialfall des Bootstrap-Verfahrens. Es lässt sich ebenfalls zur Schätzung von Konfidenzintervallen anwenden [25], wenngleich es normalerweise hauptsächlich zur Schätzung des Bias verwendet wird. Es basiert auf der Idee, die geänderten Stichproben zu betrachten, bei denen jeweils ein Wert der originalen Stichprobe entfernt wurde.

Definition 5.4.1 (Jackknife)

Sei $\underline{X} = (X_1, \dots, X_n)$ eine Stichprobe und $\hat{\theta} = \theta(\underline{X})$ ein Schätzer. Dann sind

$$\underline{X}_{(i)} = (X_1, X_2, \dots, X_{i-1}, X_{i+1}, \dots, X_n) \quad (5.4)$$

für $i = 1, \dots, n$ die Jackknife-Stichproben. [10, S. 141f]

Mit $\hat{\theta}_{(i)} = \theta(\underline{X}_{(i)})$ und $\hat{\theta}_{(\cdot)} = \sum_{i=1}^n \hat{\theta}_{(i)}/n$ lässt sich die Varianz als

$$\hat{\sigma}_{\hat{\theta}}^2 = \frac{n-1}{n} \sum_{i=1}^n (\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)})^2 \quad (5.5)$$

schätzen und daraus das Konfidenzintervall

$$\left[\hat{\theta} - u_{\frac{1+\alpha}{2}} \sqrt{\frac{1}{n} \hat{\sigma}_{\hat{\theta}}^2}, \hat{\theta} + u_{\frac{1+\alpha}{2}} \sqrt{\frac{1}{n} \hat{\sigma}_{\hat{\theta}}^2} \right] \quad (5.6)$$

bestimmen, wobei u_{α} das α -Quantil der Standardnormalverteilung bezeichnet.

Um den durch das Jackknife geschätzten Bias zu berücksichtigen, wird teilweise die Verwendung der Pseudowerte

$$ps_{(i)} = n\hat{\theta} - (n-1)\hat{\theta}_{(i)} \quad (5.7)$$

anstelle von $\hat{\theta}_{(i)}$ und $ps_{(\cdot)} = \sum_{i=1}^n ps_{(i)}/n$ anstelle von $\hat{\theta}_{(\cdot)}$ in (5.5) und (5.6) empfohlen [23]. Für die Beispieldatensätze wurden beide Varianten getestet. Da die Parameterschätzung nur $n = 24$ -mal beziehungsweise $n = 48$ -mal wiederholt werden muss, ist die Berechnungsdauer mit 1.986s (ca. 1/2 Stunde) und 3.223s (ca. 1 Stunde) für die Datensätze deutlich geringer.

In Tabelle 7 sind die 95%-Konfidenzintervalle des Jackknife dargestellt, sowohl für die

Verwendung von $\hat{\theta}_{(i)}$ als auch von $ps_{(i)}$. Für $\hat{\theta}_{(i)}$ sind diese allerdings deutlich kleiner als die Bootstrap-Konfidenzintervalle. Das liegt daran, dass die Stichproben von Wählerversuchen in der Regel mehrere Versuche auf gleichen Lastniveaus enthalten und dadurch der Einfluss eines einzelnen Wertes auf die Parameterschätzung sehr gering ist. Dadurch unterscheiden sich die geschätzten Parameter kaum, wenn jeweils ein Wert aus der Stichprobe entfernt wird.

Auch die Bias-Korrektur bei Verwendung von $ps_{(i)}$ liefert in diesem Fall nicht das gewünschte Resultat. Besonders deutlich wird dies beim Abknickpunkt S_C , für den das Konfidenzintervall [1844, 1931] berechnet wird, obwohl der Maximum-Likelihood-Schätzwert bei 2226 liegt.

	k_1	$\log N_0$	σ	S_C	k_2
$\hat{\theta}_{(i)}$	[5.34, 5.37]	[4.86, 4.94]	[0.097, 0.106]	[2182, 2269]	[13.22, 14.02]
$ps_{(i)}$	[5.31, 5.34]	[4.72, 4.80]	[0.102, 0.111]	[1844, 1931]	[13.85, 14.65]

Tab. 7: 95%-Konfidenzintervalle für die Modellparameter durch Jackknife bei Verwendung von $\hat{\theta}_{(i)}$ und $ps_{(i)}$

Für den größeren, fiktiven Beispieldatensatz sind die geschätzten Parameter sogar für alle Jackknife-Stichproben gleich denen der originalen Stichprobe, da der Einfluss eines einzelnen Wertes auf die Parameterschätzung des Datensatzes zu gering ist. Um dies zu verhindern, könnte ein *delete-k-Jackknife* verwendet werden, bei dem jeweils k Werte aus der Stichprobe entfernt werden, was letztlich aber je nach gewähltem k wieder eine deutlich höhere Berechnungsdauer aufweisen würde und dementsprechend nicht als schnellere Alternative in Frage kommt.

Die Anzahl der zu betrachtenden Jackknife-Stichproben ergibt sich dabei als $\binom{n}{k}$, was bei $n = 24$ bereits für $k = 2$ eine Betrachtung von 276 Stichproben bedeutet, während das Bootstrap-Verfahren bereits für $B = 200$ gute Ergebnisse liefert. Die Verwendung von Jackknife ist daher für das kombinierte Wählermodell nicht zu empfehlen.

5.4.2 Segmentierte lineare Regression

Standardmodelle der linearen Regression behandeln in der Regel nur eine Regressionsgerade und nicht, wie es beim kombinierten Wählermodell der Fall ist, eine stückweise lineare Funktion. Ein solches Modell führt im Allgemeinen auf eine lineare Regression mit Nebenbedingungen zurück. Analog zur Definition in Abschnitt 2.4 wird hierbei für ein Modell mit zwei Geraden

$$Y = \begin{cases} X\beta_1 + \epsilon, & x \leq c \\ X\beta_2 + \epsilon, & x > c \end{cases} \quad \text{bzw.} \quad E[Y|X] = \begin{cases} f_1(X, \beta_1), & x \leq c \\ f_2(X, \beta_2), & x > c \end{cases} \quad (5.8)$$

betrachtet. Dabei ist die Nebenbedingung

$$f_1(c, \beta_1) = f_2(c, \beta_2) \quad (5.9)$$

zu berücksichtigen, um die Stetigkeit der Wöhlerlinie zu garantieren.

Hudson hat zur Lösung hiervon in 1966 eine Methode vorgestellt, die alle zulässigen Partitionen der Daten in die zwei Bereiche bestimmt und anschließend jeweils eine Regression mit explizit gegebenen Nebenbedingungen rechnet [18]. Außerdem stellt Lerman in [19] eine weitere Methode zur Lösung mittels Gittersuche vor. Lineare Regression lässt sich also prinzipiell auch für das kombinierte Wöhlermodell mit zwei Steigungen anwenden.

5.4.3 Zensierte Regression über Kaplan-Meier-Schätzer

Das größere Problem stellen für die Regression zensierte Daten dar, da die exakte Lage dieser Werte und damit der zu minimierende Abstand zur Ausgleichsgeraden unbekannt ist. Buckley and James präsentieren in [2] eine Möglichkeit, diese Werte bei linearer Regression zu berücksichtigen, indem der Kaplan-Meier-Schätzer verwendet wird. Für das einfachste Modell $y_i = \alpha + \beta x_i + \epsilon_i$ werden die Parameter so geschätzt, sodass $\hat{\alpha}$ und $\hat{\beta}$ gerade $\int \epsilon^2 d\hat{F}_{\alpha, \beta}(\epsilon)$ minimieren, wobei

$$\hat{F}_{\alpha, \beta}(\epsilon) = 1 - \prod_{i, e_i(\alpha, \beta) \leq \epsilon} \left(\frac{n-i}{n-i-1} \right)^{\delta_i} \quad \text{und} \quad e_i(\alpha, \beta) = y_i - \alpha - \beta x_i. \quad (5.10)$$

Die Berücksichtigung zensierter Daten ist bei linearer Regression also ebenfalls möglich. Die vorgestellte Methode benötigt aber letztlich umfangreiche numerische Methoden und hat nicht immer eine eindeutige Lösung, wodurch sie innerhalb des Bootstrap-Verfahrens ungeeignet für eine einfachere, schnellere Schätzung der Modellparameter ist. Daher wurde die Idee, das Bootstrap-Verfahren durch die Verwendung linearer Regression anstelle des Maximum-Likelihood-Verfahrens zu beschleunigen, wieder verworfen.

6 Konfidenzintervalle mit Likelihood Quotient

Der Likelihood-Quotienten-Test ist ein bekannter Hypothesentest für Maximum-Likelihood-Schätzer, der auf dem asymptotischen Verhalten der Test-Statistik des Likelihood Quotienten basiert, die gerade Chi-Quadrat-verteilt ist. Durch Invertierung des Tests lassen sich Konfidenzintervalle für die geschätzten Parameter bestimmen.

6.1 Likelihood Quotient

Definition 6.1.1 (Likelihood-Quotienten-Test)

Sei $\mathcal{L}(X) = P_\theta$ mit $\theta \in \Theta \subset \mathbb{R}^d$, $L(\theta|\underline{X})$ die Likelihood- bzw. $\ell(\theta|\underline{X})$ die Log-Likelihood-Funktion, $\theta_0 \in \Theta$ und $\hat{\theta}$ ein Maximum-Likelihood-Schätzer von θ . Dann ist der (asymptotische) Likelihood-Quotienten-Test für die Nullhypothese $H_0 : \theta = \theta_0$ gegen $H_1 : \theta \neq \theta_0$ für die Likelihood Test-Statistik

$$\lambda_n(\underline{X}) = \frac{L(\theta_0|\underline{X})}{\sup_{\theta \in \Theta} L(\theta|\underline{X})} = \frac{L(\theta_0|\underline{X})}{L(\hat{\theta}|\underline{X})} \quad (6.1)$$

durch den Annahmebereich

$$C_0 = \{x : -2 \ln \lambda_n(\underline{X}) = 2[\ell(\hat{\theta}|\underline{X}) - \ell(\theta_0|\underline{X})] \leq q_{d,\alpha}\} \quad (6.2)$$

gegeben, wobei $q_{d,\alpha}$ das α -Quantil der χ^2 -Verteilung mit d Freiheitsgraden ist.

In manchen Fällen lässt sich die exakte Verteilung des Likelihood-Quotienten unter der Hypothese bestimmt werden, andernfalls wird der kritische Wert des Annahmebereiches aus der χ^2 -Verteilung bestimmt. Dieses asymptotische Verhalten von $-2 \ln \lambda_n(\underline{X})$ folgt aus der asymptotische Normalität des Maximum-Likelihood-Schätzers (siehe auch A.0.4) [22, S. 74].

6.2 Herleitung der Konfidenzintervalle

Durch Invertieren des vorgestellten Hypothesentests lässt sich ein Konfidenzintervall konstruieren. Der in (6.2) definierte Annahmebereich gibt den Bereich an, in dem die Nullhypothese $H_0 : \theta = \theta_0$ nicht verworfen werden kann. Daraus ergibt sich, dass

$$P_\theta(u(\underline{X}) \leq \theta \leq o(\underline{X})) \geq \alpha \quad (6.3)$$

für alle $\theta \in \Theta$ gilt, wenn $\theta^* =: u(\underline{X})$ beziehungsweise $\theta^* =: o(\underline{X})$ gerade maximal beziehungsweise minimal gewählt werden, sodass $2[\ell(\hat{\theta}|\underline{X}) - \ell(\theta^*|\underline{X})] \leq q_{d,(1+\alpha)/2}$ gilt. Also ist für θ^* mit $\theta_i^* = \hat{\theta}_i$ für alle $i \neq j$

$$\left[\operatorname{argmax}_{\theta_j^*, \theta_j^* < \hat{\theta}_j} \ell(\hat{\theta}|\underline{X}) - \ell(\theta^*|\underline{X}) \leq \frac{1}{2}q_{1,(1+\alpha)/2}, \operatorname{argmin}_{\theta_j^*, \theta_j^* > \hat{\theta}_j} \ell(\hat{\theta}|\underline{X}) - \ell(\theta^*|\underline{X}) \leq \frac{1}{2}q_{1,(1+\alpha)/2} \right] \quad (6.4)$$

ein approximatives α -Konfidenzintervall für θ_j [21].

Es ist zu berücksichtigen, dass in diesem Fall jeweils nur θ_j als Parameter von Interesse betrachtet wird und sich das Konfidenzintervall darauf bezieht, dass die anderen Parameter auf ihren ursprünglichen Schätzwerten fixiert sind. Dies ist bildlich so vorzustellen, dass die durch $q_{d,\alpha}$ definierte (Hyper-)Schnittebene der Log-Likelihood-Funktion als Rechteck angenommen und auf die Achse des Parameters θ_j projiziert wird. Ein möglicher Algorithmus zur Berücksichtigung der weiteren Modellparameter θ_i als variabel wird zum Beispiel in [26] vorgestellt, ist aber sehr rechenintensiv und wird daher in diesem Zusammenhang nicht als schnellere Alternative zum Bootstrap-Verfahren in Betracht gezogen.

6.3 Anwendung für Beispieldatensätze

Um das Konfidenzintervall eines Parameters θ_j zu bestimmen, werden die Lösungen von $\ell(\hat{\theta}|\underline{X}) - \ell(\theta^*|\underline{X}) \leq \frac{1}{2}q_{1,(1+\alpha)/2}$ bezüglich θ_j^* mit dem Bisektions-Verfahren numerisch approximiert werden (siehe Anhang C, Listing 20).

6.3.1 Beispiel A

Die Tabelle 8 enthält die für den Beispieldatensatz „Verdacht auf zwei Bereiche“ berechneten 95%-Konfidenzintervalle, deren Berechnung nur 5s dauert, womit das Verfahren im Vergleich zum Bootstrap-Verfahren sehr schnell ist. Allerdings sind die Konfidenzintervalle im Vergleich zu denen des Bootstrap-Verfahrens recht ungenau. Insbesondere für die untere Wählersteigung k_2 als auch für den Abknickpunkt S_C sind die berechneten Konfidenzintervalle deutlich zu groß.

k_1	$\log N_0$	σ	S_C	k_2
[3.75, 6.63]	[4.81, 4.98]	[0.071, 0.150]	[2003, 2377]	[10.90, 20.48]

Tab. 8: 95%-Konfidenzintervalle für die Modellparameter des Datensatzes „Verdacht auf zwei Bereiche“. Insbesondere für die Wählersteigung k_2 ist das berechnete Konfidenzintervall deutlich größer als beim Bootstrap-Verfahren.

6.3.2 Beispiel B

Die Tabelle 9 enthält die für den fiktiven Beispieldatensatz berechneten 95%-Konfidenzintervalle, deren Berechnung bei doppeltem Stichprobenumfang mit 10s doppelt solange wie bei Beispiel A dauert. Für diesen fiktiven Datensatz liefert der Likelihood Quotient insgesamt deutlich bessere Ergebnisse als für den realen Datensatz. Wie beim Bootstrap-Verfahren werden drei der wahren Parameter von den Konfidenzintervallen überdeckt. Die Intervalle der Wählersteigungen k_1 und k_2 sind aber verglichen mit den Bootstrap-Konfidenzintervallen etwas zu groß.

	k_1	$\log N_0$	σ	S_C	k_2
	[4.37, 5.94]	[4.88, 4.96]	[0.082, 0.134]	[191, 224]	[7.97, 9.66]
θ_i enthalten	ja	nein	ja	ja	nein

Tab. 9: 95%-Konfidenzintervalle für die Modellparameter des fiktiven Beispieldatensatzes. Wie beim Bootstrap-Verfahren sind drei Parameter im entsprechenden Konfidenzintervall enthalten, aber die Konfidenzintervalle sind etwas größer.

7 Konfidenzintervalle mit Fisher Information

Eine Möglichkeit zur Bestimmung von approximativen Konfidenzintervallen für Maximum-Likelihood-Schätzer bietet die Fisher-Information, die sich aus den zweiten Ableitungen der Log-Likelihood-Funktion ergibt. Die Idee dafür basiert auf der asymptotischen Normalität von Maximum-Likelihood-Schätzern. Die so aus der Fisher Information abgeleiteten Konfidenzintervalle werden im Englischen auch als *Fisher Matrix Confidence Bounds* bezeichnet [20].

7.1 Fisher Information

Definition 7.1.1 (Fisher Information)

Für $\mathcal{L}(X) = P_\theta$ mit $\theta \in \Theta \subset \mathbb{R}^m$, $f_\theta(x)$ stetig partiell differenzierbar bzgl. θ für fast alle x , ist die (erwartete) Fisher Information definiert als

$$I(\theta) := \left(E_\theta \left[- \frac{\partial^2}{\partial \theta_i \partial \theta_j} \ln f_\theta(X) \right] \right)_{i,j=1,\dots,m} \quad (7.1)$$

und die beobachtete Fisher Information als

$$J_n(\theta|\underline{X}) := \left(- \sum_{i=1}^n \frac{\partial^2}{\partial \theta_j \partial \theta_k} \ln f_\theta(X_i) \right)_{j,k=1,\dots,m}. \quad (7.2)$$

Nach dem Gesetz der großen Zahlen konvergiert der Term auf der rechten Seite von (7.2) im Mittel gegen den Erwartungswert, sodass $J_n(\theta|\underline{X}) \xrightarrow{p} nI(\theta)$ gilt. Für einen konsistenten Schätzer $\hat{\theta}$ gilt zusätzlich $\hat{\theta} \xrightarrow{p} \theta$, sodass die beobachtete Fisher Information als Approximation für die (erwartete) Fisher Information verwendet werden kann [9]:

$$J_n(\hat{\theta}|\underline{X}) \xrightarrow{p} nI(\theta). \quad (7.3)$$

7.2 Herleitung der Konfidenzintervalle

Nach A.0.4 sind Maximum-Likelihood-Schätzer unter gewissen Voraussetzungen asymptotisch normal mit Konvergenzrate $a_n = \sqrt{n}$, wobei die Kovarianzmatrix dem Inversen der Fisher Information entspricht. Dadurch ist

$$\left[\hat{\theta}_i - \frac{u_{\frac{1+\alpha}{2}}}{\sqrt{n}} \sqrt{(I(\hat{\theta})^{-1})_{i,i}}, \hat{\theta}_i + \frac{u_{\frac{1+\alpha}{2}}}{\sqrt{n}} \sqrt{(I(\hat{\theta})^{-1})_{i,i}} \right] \quad (7.4)$$

ein approximatives α -Konfidenzintervall für θ_i , wobei u_α das α -Quantil der Standardnormalverteilung bezeichnet [22, Prop. 4.4.10]. Für die beobachtete Fisher Information gilt analog [14]:

$$\left[\hat{\theta}_i - u_{\frac{1+\alpha}{2}} \sqrt{(J_n(\hat{\theta}|\underline{X})^{-1})_{i,i}}, \hat{\theta}_i + u_{\frac{1+\alpha}{2}} \sqrt{(J_n(\hat{\theta}|\underline{X})^{-1})_{i,i}} \right]. \quad (7.5)$$

7.3 Anwendung für Beispieldatensätze

Um die Konfidenzintervalle bestimmen zu können, ist die Fisher Information des Maximum-Likelihood-Schätzers zu berechnen. Im Falle des maximalen Wählermodells müssen dafür nicht nur die partiellen Ableitungen zweiter Ordnung von f_θ sondern im Falle von Durchläufern auch von F_θ bestimmt werden, da für die Log-Likelihood-Funktion (2.14) gilt:

$$\begin{aligned} \frac{\partial}{\partial \theta_j} \ell(\theta | \underline{X}) &= \sum_{i \in I_A} \frac{\frac{\partial^2}{\partial \theta_j^2} f_\theta(X_i)}{f_\theta(X_i)} + \sum_{i \in I_D} \frac{\frac{\partial^2}{\partial \theta_j^2} F_\theta(X_i)}{F_\theta(X_i) - 1} \\ \frac{\partial^2}{\partial \theta_j \partial \theta_k} \ell(\theta | \underline{X}) &= \sum_{i \in I_A} \frac{\left(\frac{\partial^2}{\partial \theta_j \partial \theta_k} f_\theta(X_i) \right) f_\theta(X_i) - \left(\frac{\partial}{\partial \theta_j} f_\theta(X_i) \right) \left(\frac{\partial}{\partial \theta_k} f_\theta(X_i) \right)}{\left(f_\theta(X_i) \right)^2} \\ &\quad + \sum_{i \in I_D} \frac{\left(\frac{\partial^2}{\partial \theta_j \partial \theta_k} F_\theta(X_i) \right) \left(F_\theta(X_i) - 1 \right) - \left(\frac{\partial}{\partial \theta_j} F_\theta(X_i) \right) \left(\frac{\partial}{\partial \theta_k} F_\theta(X_i) \right)}{\left(F_\theta(X_i) - 1 \right)^2} \end{aligned} \quad (7.6)$$

Für die rechnerische Bestimmung der Fisher Information wurden die partiellen Ableitungen erster Ordnung exakt bestimmt und die partiellen Ableitungen zweiter Ordnung numerisch approximiert (siehe Anhang B).

Die Berechnungen erfolgten der Einfachheit halber mit Matlab, ohne die Funktionen bezüglich der Rechengeschwindigkeit zu optimieren (siehe Anhang C, Listing 3). Da das Verfahren keine Maximierung benötigt und insbesondere nicht wie das Bootstrap-Verfahren eine große Anzahl an Wiederholungen verwendet, ist die notwendige Berechnungsdauer ohnehin im Bereich von wenigen Minuten.

7.3.1 Beispiel A

Die Tabelle 10 enthält die geschätzten Konfidenzintervalle für den Beispieldatensatz „Verdacht auf zwei Bereiche“, deren Berechnung 55s dauert. Da diese Konfidenzintervalle per Definition symmetrisch sind, unterscheiden sie sich (von $\log N_0$ abgesehen) von den nicht-symmetrischen Bootstrap-Konfidenzintervallen. Für σ (0.156 vs 0.162), S_C (2425 vs 2356) und k_2 (12.90 vs 13.05) ist jeweils eine Grenze der Intervalle sehr ähnlich, während die jeweils andere entsprechend größer beziehungsweise kleiner ist. Das Konfidenzintervall für k_1 hingegen weicht beidseitig von dem per Bootstrap bestimmten (2.60 bis 5.34) ab. Damit ist eine größere, unter Umständen auch zu große Überdeckungsrate der Konfidenzintervalle zu erwarten.

k_1	$\log N_0$	σ	S_C	k_2
[4.11, 6.61]	[4.84, 4.96]	[0.042, 0.162]	[2027, 2425]	[12.90, 14.34]

Tab. 10: 95%-Konfidenzintervalle für die Modellparameter des Beispieldatensatzes „Verdacht auf zwei Bereiche“, die per Definition symmetrisch sind und insgesamt den Bootstrap-Konfidenzintervallen sehr nahe kommen.

7.3.2 Beispiel B

Die Tabelle 11 enthält die geschätzten Konfidenzintervalle für den fiktiven Beispieldatensatz, deren Berechnung bei doppeltem Stichprobenumfang mit 111s doppelt solange wie bei Beispiel A dauert. Auch für diesen Datensatz ist das aus der Fisher Information berechnete Konfidenzintervall für $\log N_0$ nahezu identisch mit dem des Bootstrap-Verfahrens und die weiteren Konfidenzintervalle stimmen auf jeweils einer Seite fast überein (k_1 : 5.02 vs 4.99, S_C : 192 vs 189, k_2 : 9.52 vs 9.74), nur das Konfidenzintervall für σ ist deutlich verschoben. Im Gegensatz zum Bootstrap-Verfahren werden die wahren Parameter von $\log N_0$ und k_1 nicht mehr durch das Konfidenzintervall überdeckt, dafür der von σ .

	k_1	$\log N_0$	σ	S_C	k_2
	[5.02, 5.91]	[4.88, 4.96]	[0.080, 0.125]	[192, 233]	[8.20, 9.51]
θ_i enthalten	nein	nein	ja	ja	nein

Tab. 11: 95%-Konfidenzintervalle für die Modellparameter des fiktiven Beispieldatensatzes. Nur zwei Konfidenzintervalle überdecken den wahren Parameter, bei den anderen liegt der jeweils wahre Parameter nur knapp außerhalb.

Insgesamt sind für diesen Beispieldatensatz die Unterschiede zwischen den Bootstrap-Konfidenzintervallen und den aus der Fisher Information konstruierten sehr klein. Für den

realen Datensatz „Verdacht auf zwei Bereiche“ sind die Unterschiede bei halb so großem Stichprobenumfang etwas größer. Es lässt sich vermuten, dass sich dieser Umstand proportional verhält und die aus der Fisher Information konstruierten Konfidenzintervalle umso besser sind, je größer der Stichprobenumfang.

8 Fazit

Das Bootstrap-Verfahren ist ein bewährtes Verfahren zur Berechnung von Konfidenzintervallen, insbesondere im Fall komplexer Schätzer, für die keine exakten Konfidenzintervalle bekannt sind. Es lässt sich auch für das kombinierte Wählermodell ohne besonderen Programmieraufwand implementieren und liefert die Konfidenzintervalle mit der besten Überdeckungsrate, obwohl diese Konfidenzintervalle für den fiktiven Beispieldatensatz nur drei der fünf wahren Parameter überdecken.

Die Anwendung des Bootstrap-Verfahrens dauert je nach Anzahl an Bootstrap-Stichproben allerdings mehrere Stunden bis Tage, da die rechenintensive Maximum-Likelihood-Schätzung so häufig wiederholt werden muss, wie Bootstrap-Stichproben erzeugt werden. Aufgrund der Komplexität des Modells lässt es sich nicht ohne Weiteres mit vereinfachten Mitteln wie linearer Regression beschleunigen. Wenn eine kürzere Berechnungsdauer gewünscht ist, muss ein alternatives Verfahren in Betracht gezogen werden.

Eine schnellere Alternative bieten Verfahren, die das asymptotische Verhalten der Maximum-Likelihood-Schätzer ausnutzen und daraus entsprechende Konfidenzintervalle ableiten. Sowohl die aus dem Likelihood Quotienten als auch die aus der Fisher Information konstruierten Konfidenzintervalle lassen sich innerhalb weniger Sekunden beziehungsweise Minuten berechnen. Die Tabellen 12 und 13 zeigen die berechneten Konfidenzintervalle der beiden betrachteten Beispieldatensätze für die drei unterschiedlichen Verfahren.

	k_1	$\log N_0$	σ	S_C	k_2
Bootstrap	[2.60, 5.34]	[4.86, 5.01]	[0.087, 0.156]	[2158, 2356]	[13.05, 15.59]
Likelihood	[3.75, 6.63]	[4.81, 4.98]	[0.071, 0.150]	[2003, 2377]	[10.90, 20.48]
Fisher	[4.11, 6.61]	[4.84, 4.96]	[0.042, 0.162]	[2027, 2425]	[12.90, 14.34]
Schätzwert	5.35763	4.9045	0.10152	2225.88	13.6215

Tab. 12: 95%-Konfidenzintervalle aus dem Bootstrap-Verfahrens ($B = 2000$), dem Likelihood Quotienten und der Fisher Information für die Modellparameter des Datensatzes „Verdacht auf zwei Bereiche“

Insbesondere für den realen Beispieldatensatz „Verdacht auf zwei Bereiche“ liefert der Likelihood Quotient allerdings keine zufriedenstellenden Ergebnisse, da die Konfidenzintervalle teilweise deutlich zu groß sind. Nur für den fiktiven Beispieldatensatz sind die Ergebnisse genau genug.

Die Fisher Information hingegen liefert bei beiden Beispieldatensätzen Konfidenzintervalle, die denen des Bootstrap-Verfahrens sehr nahe kommen. Im Gegensatz zu jenen sind sie per Definition symmetrisch, stimmen aber auf jeweils einer Seite mit ihnen nahezu

überein.

	k_1	$\log N_0$	σ	S_C	k_2
Bootstrap	[4.99, 5.46]	[4.91, 5.00]	[0.102, 0.138]	[189, 218]	[8.85, 9.75]
Likelihood	[4.37, 5.94]	[4.88, 4.96]	[0.082, 0.134]	[191, 224]	[7.97, 9.66]
Fisher	[5.02, 5.91]	[4.88, 4.96]	[0.080, 0.125]	[192, 233]	[8.20, 9.51]
wahrer Wert	5.00	5.00	0.100	200	10.00
Schätzwert	5.46486	4.9216	0.10246	212.76	8.8531

Tab. 13: 95%-Konfidenzintervalle aus dem Bootstrap-Verfahrens ($B = 2000$), dem Likelihood Quotienten und der Fisher Information für die Modellparameter des fiktiven Beispieldatensatzes

In Anwendungsfällen, wo die Berechnungsdauer eine wichtige Rolle spielt, bieten die aus der Fisher Information berechneten Konfidenzintervalle somit eine ausreichend genaue und sehr schnelle Alternative und können ohne Bedenken verwendet werden. Wenn allerdings eine Berechnungsdauer von einigen Stunden kein Problem darstellt, sollte das Bootstrap Verfahren bevorzugt werden, das bereits bei einer vergleichsweise geringen Anzahl von beispielsweise 200 Bootstrap-Stichproben bessere Ergebnisse liefert.

Verfahren	Intervalle	Rechendauer	Bewertung der Intervalle
Bootstrap	nicht symmetrisch	Stunden bis Tage	sehr gut
Likelihood	nicht symmetrisch	wenige Sekunden	befriedigend
Fisher	symmetrisch	wenige Minuten	gut

Tab. 14: Bewertung der unterschiedlichen Verfahren zur Berechnung von Konfidenzintervallen für die Parameter des kombinierten Wählermodells

9 Ausblick

Die Konstruktion von Konfidenzintervallen aus der Fisher Information, wie sie in Kapitel 7 vorgestellt wird, lässt sich hinsichtlich ihrer Implementierung optimieren. Bisher wurden lediglich die partiellen Ableitungen erster Ordnung exakt bestimmt und die zweiter Ordnung numerisch approximiert. Wenn auch die partiellen Ableitungen erster Ordnung numerisch bestimmt werden, sind insbesondere die Konfidenzintervalle der Wählersteigungen k_1 und k_2 etwas größer (siehe Anhang B, Tabelle 15). Daraus lässt sich vermuten, dass die Konfidenzintervalle bei exakter Bestimmung der partiellen Ableitung erster und zweiter Ordnung noch etwas genauer wären. Darüber hinaus lässt sich die Implementierung auch bezüglich der Rechenzeit optimieren, wobei diese für die betrachteten Beispieldatensätze bei nur einer beziehungsweise zwei Minuten liegt. Die verwendeten Funktionen und Methoden (siehe Anhang C, Listing 3) wurden nicht hinsichtlich der Matrizenrechnung von Matlab optimiert.

Außer den Konfidenzintervallen einzelner Parameter können aber auch die gemeinsamen Konfidenzbereiche aller Parameter von großem Interesse sein. Wie in 2.6.3 dargestellt, ist die Konstruktion solcher Bereiche nicht ganz einfach, da die exakte Form unbekannt ist. Für die Interpretation des Modells genügt es aber, einen gemeinsamen Vorhersagebereich im Wählerdiagramm in Form von Konfidenzlinien zu berechnen und einzutragen.

Mit dem Bootstrap-Verfahren lassen sich ohne Probleme entsprechende Vorhersagebereiche bestimmen. Dafür muss auf beliebigen Lastniveaus für jede Bootstrap-Stichprobe die für die aus der Stichprobe geschätzten Modellparameter theoretische Lebensdauer ermittelt werden. Für die so erhaltenen Lebensdauern lassen sich auf jedem der gewählten Lastniveaus die empirischen Quantile angeben und so im Diagramm (bei einer ausreichend großen Anzahl an betrachteten Lastniveaus) als Konfidenzlinie eintragen. Die dafür notwendige Rechendauer bleibt für das Bootstrap-Verfahren gleich groß.

Beim Fisher-Verfahren lassen sich aus der berechneten Kovarianzmatrix unter Annahme der asymptotischen Normalität der Schätzer mit Hilfe einer Monte-Carlo-Simulation auf ähnliche Weise Vorhersagebereiche aufstellen. Der hierfür notwendige rechnerische Aufwand ist größer als bei der Bestimmung der einzelnen Konfidenzintervalle, bleibt aber im Minutenbereich.

10 Literatur

- [1] ABRAMOWITZ, M. ; STEGUN, I.A.: *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*. 10. Auflage. US Departemend of Commerce, 1972
- [2] BUCKLEY, J. ; JAMES, I.: Linear Regression with Censored Data. In: *Biometrika* Bd. 66. Biometrika Trust, 1979. – S. 429-436
- [3] COX, D. R. ; OAKES, D.: Analysis of Survival Data. In: *Monographs on Statistics and Applied Probability*. Chapman & Hall, 1984
- [4] CRAMÉR, H.: *Mathematical Methods of Statistics*. Asia Publishing House, 1946
- [5] CROOT, E.: *Maximum likelihood estimators and least squares*. http://people.math.gatech.edu/~ecroot/3225/maximum_likelihood.pdf, aufgerufen am 23.07.2015, 2010
- [6] CZADO, C. ; SCHMIDT, T.: *Mathematische Statistik*. Springer-Verlag, 2011
- [7] DAVISON, A. C. ; HINKLEY, D. V.: *Bootstrap methods and their application*. Cambridge University Press : Cambridge Series in Statistical and Probabilistic Mathematics, 1997
- [8] DE LEVIE, R.: An improved numerical approximation for the first derivative. In: *J. Chem. Sci.* Bd. 121. Indian Academy of Sciences, 2009. – S. 935-950
- [9] EFRON, B. ; HINKLEY, D. V.: Assessing the Accuracy of the Maximum Likelihood Estimator: Observed Versus Expected Fisher Information. In: *Biometrika* Bd. 65. Biometrika Trust, 1978. – S. 457-482
- [10] EFRON, B. ; TIBSHIRANI, R. J.: An Introduction to the Bootstrap. In: *Monographs on Statistics and Applied Probability* 57. Chapman & Hall, 1994
- [11] EVERITT, B. S.: *The Cambridge Dictionary of Statistics*. 2. Auflage. Cambridge University Press, 2002
- [12] FETH, S.: *Partially Passed Component Counting for Evaluating Reliability*, TU Kaiserslautern, Diss., 2009
- [13] FRAUNHOFER ITWM: *Entwicklung und Evaluierung statistischer Algorithmen zur flexiblen Anpassung von Wählerlinienmodellen*. 2011. – Interner Projektbericht
- [14] GEYER, C. J.: *Fisher Information and Confidence Intervals Using Maximum Like-*

- likelihood*. 2007. – University of Minnesota, <http://www.stat.umn.edu/geyer/s06/5102/notes/fish.pdf>, aufgerufen am 04.08.2015
- [15] GUMBEL, E. J.: Les valeurs extrêmes des distributions statistiques. In: *Annales de l'I. H. P.* Bd. 5. Numdam, 1935. – S. 115-158
- [16] HAIBACH, E.: *Betriebsfestigkeit*. 3. Auflage. Springer-Verlag, 2006
- [17] HARTUNG, J.: *Statistik*. 14. Auflage. Oldenbourg-Verlag, 2005
- [18] HUDSON, D. J.: Fitting Segmented Curves Whose Join Points Have to Be Estimated. In: *Journal of the American Statistical Association* Bd. 61. Wiley, 1966. – S. 1079-1129
- [19] LERMAN, P. M.: Fitting Segmented Regression Models by Grid Search. In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* Bd. 29. Wiley, 1980. – S. 77-84
- [20] NELSON, W.: *Applied Life Data Analysis*. John Wiley & Sons, Inc., 1982
- [21] PATTERSON, D.: *Profile Likelihood Confidence Intervals for GLM's*. 2014. – University of Montana, <http://www.math.umt.edu/patterson/ProfileLikelihoodCI.pdf>, aufgerufen am 21.08.2015
- [22] REDENBACH, C.: *Mathematical Statistics*. 2013. – TU Kaiserslautern, http://www.mathematik.uni-kl.de/fileadmin/AGs/stat/Lectures-Literatures/Mathematical_Statistics/Statistics.pdf, aufgerufen am 23.07.2015
- [23] SAWYER, S.: *Resampling Data: Using a Statistical Jackknife*. 2005. – Washington University, <http://www.math.wustl.edu/~sawyer/handouts/Jackknife.pdf>, aufgerufen am 23.07.2015
- [24] SPALL, J. C.: Monte Carlo Computation of the Fisher Information Matrix in Non-standard Settings. In: *Journal of Computational and Graphical Statistics* Bd. 14. 2005. – S. 889-909
- [25] TUKEY, J. W.: Bias and confidence in not quite large samples. In: *Annals of Mathematical Statistics* 29 (1958), S. 614
- [26] VENZON, D. J. ; MOOLGAVKAR, S. H.: A Method for Computing Profile-Likelihood-Based Confidence Intervals. In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* Bd. 37. Wiley, 1988. – S. 87-94
- [27] WEIBULL, W.: A statistical distribution function of wide applicability. In: *Journal of Applied Mechanics* 18 (1951), September. – S. 293 - 297

Anhang

A Sätze und Beweise

Transformationsatz für Dichten^[6, S. 5]

Satz A.0.1 Sei $X = (X_1, \dots, X_n)^T$ eine Zufallsvariable mit Werten in \mathbb{R} und Dichtefunktion $f_X : \mathbb{R} \rightarrow \mathbb{R}_+$ und $h : \mathbb{R} \rightarrow \mathbb{R}$ stetig differenzierbar mit stetig differenzierbarer Umkehrabbildung h^{-1} sowie $J_{h^{-1}}$ ihre Jacobi-Matrix. Dann gilt für die Dichte f_Y von $Y = h(X)$

$$f_Y(y) = f_X(h^{-1}(y)) | \det J_{h^{-1}}(y) |. \quad (\text{A.1})$$

Log-Transformation der Weibull-Verteilung

Satz A.0.2 Sei X eine Zufallsvariable mit $X \sim \mathcal{W}(b, T)$ und $Y := -\log X$. Dann gilt $Y \sim \mathcal{G}(\beta, \mu)$ mit $\beta = 1/b$ und $\mu = \log T$.

Beweis. Für die Transformation $Y := -\log X$ ergibt sich $h(x) = -\log x$ mit $h^{-1}(x) = e^{-x}$ und $J_{h^{-1}}(x) = -e^{-x}$. Nach A.0.1 gilt dann

$$\begin{aligned} f_Y(x) &= f_X(h^{-1}(x))e^{-x} = f_X(e^{-x})e^{-x} \\ &= \frac{b}{T} \left(\frac{e^{-x}}{T}\right)^{b-1} e^{-\left(\frac{e^{-x}}{T}\right)^b} e^{-x} \\ &= \frac{b}{T^b} e^{-xb} e^{-\left(\frac{e^{-x}}{T}\right)^b} \\ &= \frac{1}{\beta} e^{-\frac{1}{\beta}(x-\mu)} e^{-e^{-\frac{1}{\beta}(x-\mu)}} \end{aligned} \quad (\text{A.2})$$

□

Konsistenz von Maximum-Likelihood-Schätzern^[22, S. 33]

Satz A.0.3 Seien X_1, \dots, X_n unabhängig identisch verteilt mit $\mathcal{L}(X_i) = P_{\theta_0}$ mit $P_{\theta_0} \in \{P_{\theta} : \theta \in \Theta\}$, sei f_{θ} die Dichte von P_{θ} und gelte

(i) $\Theta \subset \mathbb{R}$ ist kompakt,

- (ii) $\text{supp } f_\theta = \overline{\{x : f_\theta(x) > 0\}}$ ist unabhängig von θ ,
 (iii) f_θ ist stetig differenzierbar in θ und $\frac{d}{d\theta} f_\theta(x)$ erfüllt

$$\left| \frac{d}{d\theta} f_\theta(x) \right| \leq L(x) f_\theta(x) \quad \forall x \in \mathbb{R}, \theta \in \Theta \quad (\text{A.3})$$

mit $L(x) > 0$,

- (iv) $E_{\theta_0}[L(X_1)] < \infty$,
 (v) $E_{\theta_0}[\log f_\theta(X_1)]$ existiert für alle θ und hat ein eindeutiges globales Maximum in θ_0 .

Wenn der Maximum-Likelihood-Schätzer $\hat{\theta}_n$ existiert, ist er konsistent.

Asymptotische Normalität von Maximum-Likelihood-Schätzern^[22, S. 36]

Satz A.0.4 Sei zusätzlich zu den Bedingungen aus A.0.3 folgendes erfüllt:

- (vi) f_θ ist zweimal stetig differenzierbar bezüglich θ ,
 (vii) $\frac{d^2}{d\theta^2} \log f_\theta(x)$ ist Lipschitz-stetig in θ , d.h.

$$\left| \frac{d^2}{d\theta^2} \log f_\theta(x) - \frac{d^2}{d\eta^2} \log f_\eta(x) \right| \leq H(x) |\theta - \eta| \quad (\text{A.4})$$

mit $E[H(X_1)] < \infty$,

- (viii) $0 < I(P_{\theta_0}) < \infty$.

Dann gilt

$$\sqrt{n}(\hat{\theta}_n - \theta_0) \xrightarrow[n \rightarrow \infty]{L} \mathcal{N}\left(0, \frac{1}{I(P_{\theta_0})}\right). \quad (\text{A.5})$$

Leibnizregel für Parameterintegrale^[1, S. 11]

Satz A.0.5 Für stetig differenzierbare Funktionen a , b und f gilt nach Leibniz:

$$\frac{d}{dx} \left(\int_{a(x)}^{b(x)} f(t, x) dt \right) = \int_{a(x)}^{b(x)} dx f(t, x) dt + f(b(x), x) \frac{d}{dx} b(x) - f(a(x), x) \frac{d}{dx} a(x) \quad (\text{A.6})$$

B Partielle Ableitungen der Log-Likelihood-Funktion

Im Folgenden werden die partiellen Ableitungen erster Ordnung von F_θ (4.8) und f_θ (4.9) angegeben, die für die Berechnung der Fisher Information (vgl. (7.6)) notwendig sind.

Zur Bestimmung der Ableitungen sei an die Leibnizregel für Parameterintegrale (A.0.5) erinnert. Außerdem wurden die folgenden Ableitungen verwendet:

$$\begin{aligned}\frac{\partial}{\partial x}\Phi(f(x)) &= \left(\frac{\partial}{\partial x}f(x)\right)\phi(f(x)) \\ \frac{\partial}{\partial x}\phi(f(x)) &= \left(\frac{\partial}{\partial x}f(x)\right)(-f(x))\phi(f(x))\end{aligned}\tag{B.1}$$

$$\begin{aligned}\frac{\partial}{\partial k_1}k_1\sqrt{\frac{k_2 - k_1}{k_2 + k_1}} &= \frac{k_2^2 - k_1k_2 - k_1^2}{(k_2 + k_1)^2\sqrt{\frac{k_2 - k_1}{k_2 + k_1}}} \\ \frac{\partial}{\partial k_2}k_1\sqrt{\frac{k_2 - k_1}{k_2 + k_1}} &= \frac{k_1^2}{(k_2 + k_1)^2\sqrt{\frac{k_2 - k_1}{k_2 + k_1}}}\end{aligned}\tag{B.2}$$

$$\begin{aligned}\frac{\partial}{\partial k_1}\frac{1}{\sigma\sqrt{(k_2/k_1)^2 - 1}} &= \frac{k_2^2}{\sigma k_1^3((k_2/k_1)^2 - 1)^{3/2}} \\ \frac{\partial}{\partial k_2}\frac{1}{\sigma\sqrt{(k_2/k_1)^2 - 1}} &= \frac{k_2}{\sigma k_1^2((k_2/k_1)^2 - 1)^{3/2}}\end{aligned}\tag{B.3}$$

Desweiteren werden folgenden Variablen zur verkürzten Schreibweise verwendet, die beim Bilden der partiellen Ableitungen entsprechend berücksichtigt werden müssen.

$$\begin{aligned}\mu_{i,1} &= \log N_0 + k_1(\log S_0 - \log S_i) \\ \mu_{i,2} &= \log N_0 + k_1(\log S_0 - \log S_C) + k_2(\log S_C - \log S_i) \\ a_i &= (k_2 - k_1)(\log S_i - \log S_C) \\ \tau &= \frac{\sigma}{k_1}\sqrt{\frac{k_2 + k_1}{k_2 - k_1}}\end{aligned}\tag{B.4}$$

Partielle Ableitungen erster Ordnung von F_θ

$$\begin{aligned} \frac{\partial}{\partial N_0} F_\theta(n) = & -\frac{1}{\sigma N_0 \ln(10)} \left[\phi\left(\frac{n - \mu_{i,1}}{\sigma}\right) \Phi\left(\frac{\log S_i - \log S_C}{\tau}\right) \right. \\ & \left. + \int_{a_i}^{\infty} \phi\left(\frac{n - v - \mu_{i,2}}{\sigma}\right) \frac{1}{(k_2 - k_1)\tau} \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) dv \right] \end{aligned} \quad (\text{B.5})$$

$$\begin{aligned} \frac{\partial}{\partial \sigma} F_\theta(n) = & \frac{\mu_{i,1} - n}{\sigma^2} \phi\left(\frac{n - \mu_{i,1}}{\sigma}\right) \Phi\left(\frac{\log S_i - \log S_C}{\tau}\right) \\ & + \Phi\left(\frac{n - \mu_{i,1}}{\sigma}\right) \left(\frac{\log S_C - \log S_i}{\sigma\tau}\right) \phi\left(\frac{\log S_i - \log S_C}{\tau}\right) \\ & + \int_{a_i}^{\infty} \frac{\mu_{i,2} + v - n}{\sigma^2} \phi\left(\frac{n - v - \mu_{i,2}}{\sigma}\right) \frac{1}{(k_2 - k_1)\tau} \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) \\ & + \Phi\left(\frac{n - v - \mu_{i,2}}{\sigma}\right) \frac{1}{\sigma(k_2 - k_1)\tau} \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) \left(\frac{v^2}{(k_2 - k_1)^2\tau^2} - 1\right) dv \end{aligned} \quad (\text{B.6})$$

$$\begin{aligned} \frac{\partial}{\partial S_C} F_\theta(n) = & \Phi\left(\frac{n - \mu_{i,1}}{\sigma}\right) \frac{-1}{\sigma S_C \ln(10)} \phi\left(\frac{\log S_i - \log S_C}{\tau}\right) \\ & + \int_{a_i}^{\infty} \frac{-1}{\sigma S_C \ln(10)\tau} \phi\left(\frac{n - v - \mu_{i,2}}{\sigma}\right) \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) dv \\ & + \frac{-1}{S_C \ln(10)\tau} \Phi\left(\frac{n - a_i - \mu_{i,2}}{\sigma}\right) \phi\left(\frac{a_i}{(k_2 - k_1)\tau}\right) \end{aligned} \quad (\text{B.7})$$

$$\begin{aligned} \frac{\partial}{\partial k_1} F_\theta(n) = & \frac{\log S_i - \log S_0}{\sigma} \phi\left(\frac{n - \mu_{i,1}}{\sigma}\right) \Phi\left(\frac{\log S_i - \log S_C}{\tau}\right) \\ & + \Phi\left(\frac{n - \mu_{i,1}}{\sigma}\right) \frac{\log S_i - \log S_C}{\sigma} \frac{k_2^2 - k_1 k_2 - k_1^2}{(k_2 + k_1)^2 \sqrt{\frac{k_2 - k_1}{k_2 + k_1}}} \phi\left(\frac{\log S_i - \log S_C}{\tau}\right) \\ & + \int_{a_i}^{\infty} \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) \left[\frac{\log S_C - \log S_0}{\sigma(k_2 - k_1)\tau} \phi\left(\frac{n - v - \mu_{i,2}}{\sigma}\right) \right. \\ & \left. + \Phi\left(\frac{n - v - \mu_{i,2}}{\sigma}\right) \frac{k_2^2}{\sigma k_1^3 ((k_2/k_1)^2 - 1)^{3/2}} \left(1 - \frac{v^2}{(k_2 - k_1)^2\tau^2}\right) \right] dv \\ & + \frac{\log S_C - \log S_i}{(k_2 - k_1)\tau} \Phi\left(\frac{n - a_i - \mu_{i,2}}{\sigma}\right) \phi\left(\frac{a_i}{(k_2 - k_1)\tau}\right) \end{aligned} \quad (\text{B.8})$$

$$\begin{aligned}
\frac{\partial}{\partial k_2} F_\theta(n) = & \Phi\left(\frac{n - \mu_{i,1}}{\sigma}\right) \frac{\log S_i - \log S_C}{\sigma} \frac{k_1^2}{(k_2 + k_1)^2 \sqrt{\frac{k_2 - k_1}{k_2 + k_1}}} \phi\left(\frac{\log S_i - \log S_C}{\tau}\right) \\
& + \int_{a_i}^{\infty} \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) \left[\frac{\log S_i - \log S_C}{\sigma(k_2 - k_1)\tau} \phi\left(\frac{n - v - \mu_{i,2}}{\sigma}\right) \right. \\
& + \left. \Phi\left(\frac{n - v - \mu_{i,2}}{\sigma}\right) \frac{k_2}{\sigma k_1^2 ((k_2/k_1)^2 - 1)^{3/2}} \left(1 - \frac{v^2}{(k_2 - k_1)^2 \tau^2}\right) \right] dv \\
& + \frac{\log S_i - \log S_C}{(k_2 - k_1)\tau} \Phi\left(\frac{n - a_i - \mu_{i,2}}{\sigma}\right) \phi\left(\frac{a_i}{(k_2 - k_1)\tau}\right)
\end{aligned} \tag{B.9}$$

Partielle Ableitungen erster Ordnung von f_θ

$$\begin{aligned}
\frac{\partial}{\partial N_0} f_\theta(n) = & - \frac{1}{\sigma^2 N_0 \ln(10)} \left[\frac{\mu_{i,1} - n}{\sigma} \phi\left(\frac{n - \mu_{i,1}}{\sigma}\right) \Phi\left(\frac{\log S_i - \log S_C}{\tau}\right) \right. \\
& \left. + \int_{a_i}^{\infty} \frac{\mu_{i,2} + v - n}{\sigma(k_2 - k_1)\tau} \phi\left(\frac{n - v - \mu_{i,2}}{\sigma}\right) \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) dv \right]
\end{aligned} \tag{B.10}$$

$$\begin{aligned}
\frac{\partial}{\partial \sigma} f_\theta(n) = & \frac{(\mu_{i,1} - n)^2 - \sigma^2}{\sigma^4} \phi\left(\frac{n - \mu_{i,1}}{\sigma}\right) \Phi\left(\frac{\log S_i - \log S_C}{\tau}\right) \\
& + \frac{1}{\sigma} \phi\left(\frac{n - \mu_{i,1}}{\sigma}\right) \left(\frac{\log S_C - \log S_i}{\sigma\tau}\right) \phi\left(\frac{\log S_i - \log S_C}{\tau}\right) \\
& + \int_{a_i}^{\infty} \frac{(\mu_{i,2} + v - n)^2 - \sigma^2}{\sigma^4} \phi\left(\frac{n - v - \mu_{i,2}}{\sigma}\right) \frac{1}{(k_2 - k_1)\tau} \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) \\
& + \phi\left(\frac{n - v - \mu_{i,2}}{\sigma}\right) \frac{1}{\sigma^2(k_2 - k_1)\tau} \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) \left(\frac{v^2}{(k_2 - k_1)^2 \tau^2} - 1\right) dv
\end{aligned} \tag{B.11}$$

$$\begin{aligned}
\frac{\partial}{\partial S_C} f_\theta(n) = & \phi\left(\frac{n - \mu_{i,1}}{\sigma}\right) \frac{-1}{\sigma^2 S_C \ln(10)} \phi\left(\frac{\log S_i - \log S_C}{\tau}\right) \\
& + \int_{a_i}^{\infty} \frac{n - v - \mu_{i,2}}{\sigma^2 S_C \ln(10)\tau} \phi\left(\frac{n - v - \mu_{i,2}}{\sigma}\right) \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) dv \\
& + \frac{-1}{\sigma S_C \ln(10)\tau} \phi\left(\frac{n - a_i - \mu_{i,2}}{\sigma}\right) \phi\left(\frac{a_i}{(k_2 - k_1)\tau}\right)
\end{aligned} \tag{B.12}$$

$$\begin{aligned}
\frac{\partial}{\partial k_1} f_{\theta}(n) &= \frac{(\log S_i - \log S_0)(\mu_{i,1} - n)}{\sigma^3} \phi\left(\frac{n - \mu_{i,1}}{\sigma}\right) \Phi\left(\frac{\log S_i - \log S_C}{\tau}\right) \\
&+ \phi\left(\frac{n - \mu_{i,1}}{\sigma}\right) \frac{\log S_i - \log S_C}{\sigma^2} \frac{k_2^2 - k_1 k_2 - k_1^2}{(k_2 + k_1)^2 \sqrt{\frac{k_2 - k_1}{k_2 + k_1}}} \phi\left(\frac{\log S_i - \log S_C}{\tau}\right) \\
&+ \int_{a_i}^{\infty} \phi\left(\frac{n - v - \mu_{i,2}}{\sigma}\right) \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) \left[\frac{\log S_C - \log S_0}{\sigma^2 (k_2 - k_1)\tau} \right. \\
&+ \left. \frac{k_2^2}{\sigma^2 k_1^3 ((k_2/k_1)^2 - 1)^{3/2}} \left(1 - \frac{v^2}{(k_2 - k_1)^2 \tau^2}\right) \right] dv \\
&+ \frac{\log S_C - \log S_i}{\sigma (k_2 - k_1)\tau} \phi\left(\frac{n - a_i - \mu_{i,2}}{\sigma}\right) \phi\left(\frac{a_i}{(k_2 - k_1)\tau}\right)
\end{aligned} \tag{B.13}$$

$$\begin{aligned}
\frac{\partial}{\partial k_2} f_{\theta}(n) &= \phi\left(\frac{n - \mu_{i,1}}{\sigma}\right) \frac{\log S_i - \log S_C}{\sigma^2} \frac{k_1^2}{(k_2 + k_1)^2 \sqrt{\frac{k_2 - k_1}{k_2 + k_1}}} \phi\left(\frac{\log S_i - \log S_C}{\tau}\right) \\
&+ \int_{a_i}^{\infty} \phi\left(\frac{n - v - \mu_{i,2}}{\sigma}\right) \phi\left(\frac{v}{(k_2 - k_1)\tau}\right) \left[\frac{(\log S_i - \log S_C)(\mu_{i,2} + v - n)}{\sigma^3 (k_2 - k_1)\tau} \right. \\
&+ \left. \frac{k_2}{\sigma^2 k_1^2 ((k_2/k_1)^2 - 1)^{3/2}} \left(1 - \frac{v^2}{(k_2 - k_1)^2 \tau^2}\right) \right] dv \\
&+ \frac{\log S_i - \log S_C}{\sigma (k_2 - k_1)\tau} \phi\left(\frac{n - a_i - \mu_{i,2}}{\sigma}\right) \phi\left(\frac{a_i}{(k_2 - k_1)\tau}\right)
\end{aligned} \tag{B.14}$$

Naherungsformel fur partiellen Ableitungen zweiter Ordnung

Zur Approximation der partiellen Ableitungen zweiter Ordnung aus den bekannten partiellen Ableitungen erster Ordnung wird die Naherungsformel (B.15) mit $h = 0.0001 \cdot \hat{\theta}_i$ verwendet, wie in [24] empfohlen. Fur die Berechnung in einer endgultigen Software wird empfohlen, auf komplexere Algorithmen zuruckzugreifen (vgl. [8]).

$$\frac{\partial}{\partial x} f(x) \approx \frac{-f(x + 2h) + 8f(x + h) - 8f(x - h) + f(x - 2h)}{12h} \tag{B.15}$$

Konfidenzintervalle bei numerischer Fisher Information

Um einschätzen zu können, wie groß der Einfluss der Approximation der partiellen Ableitungen zweiter Ordnung ist, wurden auch Konfidenzintervalle bei vollständiger numerischer Approximation der Fisher Information bestimmt. Dafür wurde (B.15) für die partiellen Ableitungen erster Ordnung verwendet, (B.16) für die partiellen Ableitungen zweiter Ordnung.

$$\frac{\partial}{\partial x \partial y} f(x, y) \approx \frac{f(x+h, y+k) - f(x+h, y-k) - f(x-h, y+k) + f(x-h, y-k)}{4hk} \quad (\text{B.16})$$

Im Vergleich zu den Ergebnissen bei exakter Bestimmung der partiellen Ableitungen erster Ordnung sind für den realen Datensatz aus Beispiel A die Konfidenzintervalle für σ und S_C etwas kleiner und für k_2 deutlich größer. Für den fiktiven Beispieldatensatz aus Beispiel B sind die Konfidenzintervalle beider Wählersteigungen k_1 und k_2 etwas größer. Damit zeigt sich, dass für den Praxiseinsatz auch die partiellen Ableitungen zweiter Ordnung exakt bestimmt werden sollten, um den numerischen Fehler möglichst klein zu halten.

	k_1	$\log N_0$	σ	S_C	k_2
Beispiel A	[4.47, 6.24]	[4.81, 4.99]	[0.077, 0.126]	[2115, 2337]	[10.42, 16.82]
Beispiel B	[4.69, 6.24]	[4.86, 4.97]	[0.078, 0.127]	[192, 234]	[7.55, 10.16]

Tab. 15: 95%-Konfidenzintervalle für die Modellparameter bei vollständiger numerischer Approximation der Fisher Information

C Quellcodes

createExample.m

```

1 %% Define variables
2 k1 = 5;
3 k2 = 10;
4 logN0 = log10(10^5);
5 logS0 = log10(270);
6 logSC = log10(200);
7 limit = 2*10^7;
8 sigma = 0.1;
9 tau = sigma / k1 * sqrt( (k2 + k1) / (k2 - k1) );
10 levels = 130:20:270;
11 samplesPerLevel = 6;
12 samples = samplesPerLevel * length(levels);
13 logS = log10(sort(repmat(levels, 1, samplesPerLevel), 'descend'));
14
15 %% Create sample
16 delta = normrnd(0, tau, 1, samples);
17 epsilon = normrnd(0, sigma, 1, samples);
18 indexK1 = (delta <= logS - logSC);
19 indexK2 = (delta > logS - logSC);
20 logN = zeros(1, samples);
21 logN(indexK1) = logN0 + k1 * (logS0 - logS(indexK1)) + epsilon(indexK1);
22 logN(indexK2) = logN0 + k1 * (logS0 - logSC) + k2 * (logSC - logS(
    indexK2)) + (k2 - k1) * delta(indexK2) + epsilon(indexK2);
23 N = 10.^logN;
24
25 %% Censor sample
26 X = min(N, limit);
27 C = (N >= limit);

```

Listing 1: createExample.m - Erzeugung des fiktiven Beispieldatensatzes

data.m

```

1 VerdachtAufZweiBereiche = [ ...
2     2720,    69000, 1; 2720,    73500, 1; 2720,    74250, 1; ...
3     2720,    76500, 1; 2720,    78000, 1; 2720,   106500, 1; ...
4     2400,   123750, 1; 2400,   146925, 1; 2400,   152250, 1; ...
5     2400,   166500, 1; 2400,   195750, 1; 2400,   219750, 1; ...
6     2080,   488250, 1; 2080,   541500, 1; 2080,   635250, 1; ...

```

```

7   2080, 1018500, 1; 2080, 1209750, 1; 2080, 1539450, 1; ...
8   1760, 1005750, 1; 1760, 3750000, 0; 1760, 3750000, 0; ...
9   1760, 3750000, 0; 1760, 3750000, 0; 1760, 3750000, 0 ...
10  ];
11
12  VerdachtAufZweiBereicheThetaHat = struct( 'k1', 5.35763, 'N0', 80253, '
      sigma', 0.10152, 'SC', 2225.88, 'tau', 0.02871, 'k2', 13.6215 );
13
14  FiktiverBeispieldatensatz = [ ...
15     270,     96097, 1; 270,     75707, 1; 270,     93387, 1; ...
16     270,     47511, 1; 270,     77858, 1; 270,     72004, 1; ...
17     250,    116325, 1; 250,    139892, 1; 250,    136328, 1; ...
18     250,    229910, 1; 250,    128808, 1; 250,    138712, 1; ...
19     230,    155327, 1; 230,    199729, 1; 230,    163826, 1; ...
20     230,    224496, 1; 230,    271324, 1; 230,    244684, 1; ...
21     210,    501508, 1; 210,    241362, 1; 210,    584963, 1; ...
22     210,    563283, 1; 210,    361221, 1; 210,    288607, 1; ...
23     190,   1231600, 1; 190,    505575, 1; 190,    520080, 1; ...
24     190,    710704, 1; 190,    421239, 1; 190,    488737, 1; ...
25     170,   2129694, 1; 170,   2156811, 1; 170,   1352907, 1; ...
26     170,   3484762, 1; 170,   3481865, 1; 170,   2747902, 1; ...
27     150,  10497185, 1; 150,   6771357, 1; 150,   9777079, 1; ...
28     150,   8144300, 1; 150,   6953737, 1; 150,   6909458, 1; ...
29     130,  20000000, 0; 130,  20000000, 0; 130,  19407585, 1; ...
30     130,  14423533, 1; 130,  20000000, 0; 130,  20000000, 0 ...
31  ];
32
33  FiktiverBeispieldatensatzThetaHat = struct( 'k1', 5.46486, 'N0', 83481,
      'sigma', 0.10246, 'SC', 212.76, 'tau', 0.03854, 'k2', 8.8531 );

```

Listing 2: data.m - Daten der beiden Beispieldatensätze

ci.m

```

1  function [ c ] = ci( data, theta, alpha )
2      if nargin < 3
3          alpha = 0.95;
4      end
5
6      inverse = (fisherMatrix(data, thetaHat, 1e-4))^-1;
7      u = norminv( (1 + alpha) / 2 );
8      theta = [theta.N0; theta.sigma; theta.SC; theta.k1; theta.k2];
9      c = [hat - u .* sqrt(diag(inverse)), hat + u .* sqrt(diag(inverse))];
10 end

```

Listing 3: ci.m - Berechnung der Konfidenzintervalle aus Fisher Information

fisherMatrix.m

```

1 function [matrix] = fisherMatrix(data, tHat, partialH)
2     global S N
3
4     S = data(:,1);
5     N = data(:,2);
6     d = data(:,3);
7     t = [tHat.N0, tHat.sigma, tHat.SC, tHat.k1, tHat.k2];
8
9     d0F = @(i) [dN0_F(t,i), dsigma_F(t,i), dSC_F(t,i), dk1_F(t,i), dk2_F(t,i)];
10    d0f = @(i) [dN0_dF(t,i), dsigma_dF(t,i), dSC_dF(t,i), dk1_dF(t,i), dk2_dF(t
    ,i)];
11
12    ddF = @(i) [d0j(@dN0_F,1,i), d0j(@dsigma_F,1,i), d0j(@dSC_F,1,i), d0j(
    @dk1_F,1,i), d0j(@dk2_F,1,i); d0j(@dN0_F,2,i), d0j(@dsigma_F,2,i),
    d0j(@dSC_F,2,i), d0j(@dk1_F,2,i), d0j(@dk2_F,2,i); d0j(@dN0_F,3,i),
    d0j(@dsigma_F,3,i), d0j(@dSC_F,3,i), d0j(@dk1_F,3,i), d0j(@dk2_F
    ,3,i); d0j(@dN0_F,4,i), d0j(@dsigma_F,4,i), d0j(@dSC_F,4,i), d0j(
    @dk1_F,4,i), d0j(@dk2_F,4,i); d0j(@dN0_F,5,i), d0j(@dsigma_F,5,i),
    d0j(@dSC_F,5,i), d0j(@dk1_F,5,i), d0j(@dk2_F,5,i)];
13
14    ddf = @(i) [d0j(@dN0_dF,1,i), d0j(@dsigma_dF,1,i), d0j(@dSC_dF,1,i),
    d0j(@dk1_dF,1,i), d0j(@dk2_dF,1,i); d0j(@dN0_dF,2,i), d0j(
    @dsigma_dF,2,i), d0j(@dSC_dF,2,i), d0j(@dk1_dF,2,i), d0j(@dk2_dF,2,
    i); d0j(@dN0_dF,3,i), d0j(@dsigma_dF,3,i), d0j(@dSC_dF,3,i), d0j(
    @dk1_dF,3,i), d0j(@dk2_dF,3,i); d0j(@dN0_dF,4,i), d0j(@dsigma_dF,4,
    i), d0j(@dSC_dF,4,i), d0j(@dk1_dF,4,i), d0j(@dk2_dF,4,i); d0j(
    @dN0_dF,5,i), d0j(@dsigma_dF,5,i), d0j(@dSC_dF,5,i), d0j(@dk1_dF,5,
    i), d0j(@dk2_dF,5,i)];
15
16    matrix = -[d0jd0k_1(1,1), d0jd0k_1(1,2), d0jd0k_1(1,3), d0jd0k_1(1,4),
    d0jd0k_1(1,5); d0jd0k_1(2,1), d0jd0k_1(2,2), d0jd0k_1(2,3),
    d0jd0k_1(2,4), d0jd0k_1(2,5); d0jd0k_1(3,1), d0jd0k_1(3,2),
    d0jd0k_1(3,3), d0jd0k_1(3,4), d0jd0k_1(3,5); d0jd0k_1(4,1),
    d0jd0k_1(4,2), d0jd0k_1(4,3), d0jd0k_1(4,4), d0jd0k_1(4,5);
    d0jd0k_1(5,1), d0jd0k_1(5,2), d0jd0k_1(5,3), d0jd0k_1(5,4),
    d0jd0k_1(5,5)];
17
18    function [y] = d0jd0k_1(j, k)

```

```

19     y = 0;
20
21     for i = find(d == 1)
22         ddfi = ddf(i);
23         dfi = d0f(i);
24         y = y + (ddfi(j,k)*dF(theta,i)-dfi(j)*dfi(k))/(dF(theta,i))^2;
25     end
26
27     for i = find(d == 0)
28         ddFi = ddF(i);
29         dFi = d0F(i);
30         y = y + (ddFi(j,k)*(F(theta,i)-1)-dFi(j)*dFi(k))/(F(theta,i)-1)^2;
31     end
32 end
33
34 function [y] = partial(f, x, h)
35     if nargin < 3
36         h = partialH*x;
37     end
38
39     y = (- f(x+2*h) + 8*f(x+h) - 8*f(x-h) + f(x-2*h)) / (12*h);
40 end
41
42 function [y] = d0j(f, j, i)
43     y = partial(@(x) f(newTheta(j, x), i), theta(j));
44 end
45
46 function [t] = newTheta(j, x)
47     t = theta;
48     t(j) = x;
49 end
50 end

```

Listing 4: fisherMatrix.m - Berechnung der beobachteten Fisher Information

param.m

```

1 function [S0,N0,sigma,SC,k1,k2,n,Si,mu1,mu2,a,tau] = param(theta,i)
2     global S N
3
4     S0 = S(1);
5     N0 = theta(1);
6     sigma = theta(2);
7     SC = theta(3);

```

```

8   k1 = theta(4);
9   k2 = theta(5);
10  n = log10(N(i));
11  Si = S(i);
12  mu1 = log10(theta(1)) + theta(4) * (log10(S0) - log10(S(i)));
13  mu2 = log10(theta(1)) + theta(4) * (log10(S0) - log10(theta(3))) +
      theta(5) * (log10(theta(3)) - log10(S(i)));
14  a = (theta(5) - theta(4)) * (log10(S(i)) - log10(theta(3)));
15  tau = theta(2)/theta(4)*((theta(5)+theta(4))/(theta(5)-theta(4)))^0.5;
16  end

```

Listing 5: param.m - Berechnung der verwendeten Variablen

Phi.m

```

1  function [y] = Phi(x)
2    y = normcdf(x);
3  end

```

Listing 6: Phi.m - Berechnung von $\Phi(x)$

dPhi.m

```

1  function [y] = dPhi(x)
2    y = normpdf(x);
3  end

```

Listing 7: dPhi.m - Berechnung von $\phi(x)$

F.m

```

1  function [y] = F(theta, i)
2    [S0, N0, sigma, SC, k1, k2, n, Si, mu1, mu2, a, tau] = param(theta, i);
3    y = Phi( (n - mu1)/sigma ) * Phi( (log10(Si) - log10(SC))/tau ) + ...
4    integral(@(v) ( Phi( (n - v - mu2) ./ sigma) .* 1 ./ ( (k2 - k1) .*
      tau ) .* dPhi( v ./ ( (k2 - k1) .* tau ) ) ), a, Inf);
5  end

```

Listing 8: F.m - Berechnung von $F_{\theta}(n)$

dF.m

```

1 function [y] = dF(theta, i)
2   [S0, N0, sigma, SC, k1, k2, n, Si, mu1, mu2, a, tau] = param(theta, i);
3   y = 1/sigma * dPhi((n-mu1)/sigma) * Phi((log10(Si)-log10(SC))/tau) + ...
4     integral(@(v) ( 1 ./ sigma .* dPhi( (n - v - mu2) ./ sigma) .* 1 ./
        ( (k2 - k1) .* tau ) .* dPhi( v ./ ( (k2 - k1) .* tau) ) ), a,
        Inf);
5 end

```

Listing 9: dF.m - Berechnung von $f_{\theta}(n)$ **dN0_F.m**

```

1 function [y] = dN0_F(theta, i)
2   [S0, N0, sigma, SC, k1, k2, n, Si, mu1, mu2, a, tau] = param(theta, i);
3   y = - 1 / (sigma * N0 * log(10)) * ( ...
4     dPhi( (n - mu1)/sigma ) * Phi( (log10(Si) - log10(SC))/tau ) + ...
5     integral(@(v) ( dPhi( (n - v - mu2) ./ sigma) .* 1 ./ ( (k2 - k1) .*
        tau ) .* dPhi( v ./ ( (k2 - k1) .* tau) ) ), a, Inf) );
6 end

```

Listing 10: dN0_F.m - Partielle Ableitung erster Ordnung: $F_{\theta}(n)$ nach N_0 **dsigma_F.m**

```

1 function [y] = dsigma_F(theta, i)
2   [S0, N0, sigma, SC, k1, k2, n, Si, mu1, mu2, a, tau] = param(theta, i);
3   y = (mu1 - n) / sigma^2 * dPhi( (n - mu1) / sigma ) * Phi( (log10(Si)
4     - log10(SC)) / tau ) + ...
5     Phi( (n - mu1) / sigma ) * (log10(SC) - log10(Si)) / (sigma * tau) *
6     dPhi( (log10(Si) - log10(SC)) / tau ) + ...
7     integral(@(v) ( ...
8       1 ./ ((k2 - k1) .* tau) .* dPhi( v ./ ( (k2 - k1) .* tau) ) .* (...
9       (mu2 + v - n) ./ sigma.^2 .* dPhi( (n - v - mu2) ./ sigma) + ...
10      1 ./ sigma .* Phi( (n - v - mu2) ./ sigma) .* (v.^2 / ((k2 - k1)
        .^2 .* tau.^2) - 1) ...
        ) ), a, Inf);
10 end

```

Listing 11: dsigma_F.m - Partielle Ableitung erster Ordnung: $F_{\theta}(n)$ nach σ

dSC_F.m

```

1 function [y] = dSC_F(theta, i)
2   [S0, N0, sigma, SC, k1, k2, n, Si, mu1, mu2, a, tau] = param(theta, i);
3   y = - 1 / (sigma * SC * log(10)) * ( ...
4     Phi( (n - mu1)/sigma ) * dPhi( (log10(Si) - log10(SC))/tau ) + ...
5     integral(@(v) ( ...
6       1 ./ tau .* dPhi( (n - v - mu2) ./ sigma) .* dPhi( v ./ ( (k2 - k1)
7         ) .* tau) ) ...
8     ), a, Inf) + ...
9     1/tau * Phi( (n - a - mu2)/sigma) * dPhi( a / ( (k2 - k1)*tau) );
10  end

```

Listing 12: dSC_F.m - Partielle Ableitung erster Ordnung: $F_\theta(n)$ nach S_C

dk1_F.m

```

1 function [y] = dk1_F(theta, i)
2   [S0, N0, sigma, SC, k1, k2, n, Si, mu1, mu2, a, tau] = param(theta, i);
3   y = (log10(Si) - log10(S0)) / sigma * dPhi( (n - mu1) / sigma ) * Phi(
4     (log10(Si) - log10(SC)) / tau ) + ...
5     (log10(Si) - log10(SC)) / sigma * Phi( (n - mu1) / sigma ) * dPhi( (
6     log10(Si) - log10(SC)) / tau ) * ...
7     (k2^2 - k1*k2 - k1^2) / ((k2+k1)^2 * sqrt((k2-k1)/(k2+k1))) ...
8     + integral(@(v) ( dPhi( v ./ ( (k2 - k1) .* tau) ) .* ( ...
9       (log10(SC) - log10(S0)) ./ (sigma .* (k2 - k1) .* tau) .* dPhi
10      ( (n - v - mu2) ./ sigma) + ...
11      Phi( (n - v - mu2) ./ sigma) .* k2.^2 / (sigma .* k1.^3 .* ((
12      k2./k1).^2 - 1) .^(3/2)) .* ...
13      (1 - v.^2 / ( (k2 - k1).^2 .* tau.^2) ) ) ), a, Inf) ...
14     + (log10(SC) - log10(Si)) / ((k2 - k1) * tau) * Phi( (n - a - mu2) /
15     sigma) * dPhi( a / ( (k2 - k1) * tau) );
16  end

```

Listing 13: dk1_F.m - Partielle Ableitung erster Ordnung: $F_\theta(n)$ nach k_1

dk2_F.m

```

1 function [y] = dk2_F(theta, i)
2   [S0, N0, sigma, SC, k1, k2, n, Si, mu1, mu2, a, tau] = param(theta, i);
3   y = Phi( (n - mu1) / sigma ) * (log10(Si) - log10(SC)) / sigma * ...

```

```

4      k1^2 / ((k2 + k1)^2 * sqrt((k2-k1)/(k2+k1))) * dPhi( (log10(Si) -
      log10(SC)) / tau) ...
5      + integral(@(v) ( dPhi( v ./ ( (k2 - k1) .* tau) ) .* ( ...
6          (log10(Si) - log10(SC)) ./ (sigma .* (k2 - k1) .* tau) .* dPhi
          ( (n - v - mu2) ./ sigma) + ...
7          Phi( (n - v - mu2) ./ sigma) .* k2 / (sigma .* k1.^2 .* ((k2./
          k1).^2 - 1) .^(3/2)) .* ...
8          (1 - v.^2 / ( (k2 - k1).^2 .* tau.^2) ) ) ), a, Inf) ...
9      + (log10(Si) - log10(SC)) / ((k2 - k1) * tau) * Phi( (n - a - mu2) /
      sigma) * dPhi( a / ( (k2 - k1) * tau) );
10 end

```

Listing 14: dk2_F.m - Partielle Ableitung erster Ordnung: $F_{\theta}(n)$ nach k_2

dN0_dF.m

```

1 function [y] = dN0_dF(theta, i)
2 [S0, N0, sigma, SC, k1, k2, n, Si, mu1, mu2, a, tau] = param(theta, i);
3 y = - 1 / (sigma^2 * N0 * log(10)) * ( ...
4     (mu1 - n) / sigma * dPhi( (n - mu1) / sigma ) * Phi( (log10(Si) -
      log10(SC)) / tau ) + ...
5     integral(@(v) ( (mu2 + v - n) ./ ( sigma .* (k2 - k1) .* tau ) .*
      dPhi( (n - v - mu2) ./ sigma) .* dPhi( v ./ ( (k2 - k1) .* tau) )
      ), a, Inf) );
6 end

```

Listing 15: dN0_dF.m - Partielle Ableitung erster Ordnung: $f_{\theta}(n)$ nach N_0

dsigma_dF.m

```

1 function [y] = dsigma_dF(theta, i)
2 [S0, N0, sigma, SC, k1, k2, n, Si, mu1, mu2, a, tau] = param(theta, i);
3 y = ((mu1 - n)^2 - sigma^2) / sigma^4 * dPhi( (n - mu1) / sigma ) *
      Phi( (log10(Si) - log10(SC)) / tau ) + ...
4     dPhi( (n - mu1) / sigma ) * (log10(SC) - log10(Si)) / (sigma^2 * tau
      ) * dPhi( (log10(Si) - log10(SC)) / tau ) + ...
5     integral(@(v) ( ( ((mu2+v-n).^2 - sigma.^2) ./ ( sigma.^4 .* (k2-k1)
      .* tau ) + 1 ./ (sigma.^2 .* (k2-k1) .* tau) .* (v.^2 / ((k2-k1)
      .^2 .* tau.^2) - 1) ) .* dPhi( (n-v-mu2)./sigma) .* dPhi( v ./ (
      (k2-k1).*tau) ) ), a, Inf);
6 end

```

Listing 16: dsigma_dF.m - Partielle Ableitung erster Ordnung: $f_{\theta}(n)$ nach σ

dSC_dF.m

```

1 function [y] = dSC_dF(theta, i)
2   [S0, N0, sigma, SC, k1, k2, n, Si, mu1, mu2, a, tau] = param(theta, i);
3   y = 1 / (sigma * SC * log(10)) * ( ...
4     -1/sigma*dPhi((n - mu1)/sigma) * dPhi((log10(Si)-log10(SC))/tau) + ...
5     integral(@(v) ( (n - v - mu2) ./ (sigma .* tau) .* dPhi( (n - v - mu2)
6       ) ./ sigma) .* dPhi( v ./ ( (k2 - k1) .* tau) ) ), a, Inf) - 1/
7     tau * dPhi( (n-a-mu2)/sigma) * dPhi( a/( (k2-k1)*tau) ) );
8 end

```

Listing 17: dSC_dF.m - Partielle Ableitung erster Ordnung: $f_{\theta}(n)$ nach S_C

dk1_dF.m

```

1 function [y] = dk1_dF(theta, i)
2   [S0, N0, sigma, SC, k1, k2, n, Si, mu1, mu2, a, tau] = param(theta, i);
3   y = (log10(Si) - log10(S0)) * (mu1 - n) / sigma^3 * ...
4     dPhi( (n-mu1)/sigma ) * Phi( (log10(Si)-log10(SC))/tau ) + ...
5     (log10(Si) - log10(SC)) / sigma^2 * dPhi( (n - mu1) / sigma ) * dPhi
6     ( (log10(Si) - log10(SC)) / tau ) * ...
7     (k2^2 - k1*k2 - k1^2) / ((k2+k1)^2 * sqrt((k2-k1)/(k2+k1))) ...
8     + integral(@(v) ( ...
9       dPhi( v ./ ( (k2-k1).*tau) ) .* dPhi( (n-v-mu2)./sigma) .* ( ...
10        (log10(SC) - log10(S0)) ./ (sigma.^2 .* (k2 - k1) .* tau) + ...
11        k2.^2 / (sigma.^2 .* k1.^3 .* ((k2./k1).^2 - 1) .^(3/2)) .* ...
12        (1 - v.^2 / ( (k2 - k1).^2 .* tau.^2) ) ), a, Inf) + ...
13     (log10(SC) - log10(Si)) / (sigma * (k2 - k1) * tau) * dPhi( (n - a -
14       mu2) / sigma) * dPhi( a / ( (k2 - k1) * tau) );
15 end

```

Listing 18: dk1_dF.m - Partielle Ableitung erster Ordnung: $f_{\theta}(n)$ nach k_1

dk2_dF.m

```

1 function [y] = dk2_dF(theta, i)
2   [S0, N0, sigma, SC, k1, k2, n, Si, mu1, mu2, a, tau] = param(theta, i);
3   y = dPhi( (n-mu1)/sigma ) * (log10(Si)-log10(SC)) / sigma^2 * ...
4     k1^2 / ((k2 + k1)^2 * sqrt((k2-k1)/(k2+k1))) * dPhi( (log10(Si) -
5     log10(SC)) / tau) ...
6     + integral(@(v) ( ...
7       dPhi((n-v-mu2)./sigma) .* dPhi(v ./ ((k2-k1).*tau)) .* ( ...

```

```

7      (log10(Si) - log10(SC)) .* (mu2 + v - n) ./ (sigma.^3 .* (k2 -
8          k1) .* tau) + ...
9      k2 / (sigma.^2 .* k1.^2 .* ((k2./k1).^2 - 1) .^(3/2)) .* ...
10     (1 - v.^2 / ( (k2 - k1).^2 .* tau.^2) ) ), a, Inf) + ...
11     (log10(Si) - log10(SC)) / (sigma * (k2 - k1) * tau) * dPhi( (n - a -
12         mu2) / sigma) * dPhi( a / ( (k2 - k1) * tau) );
13 end

```

Listing 19: dk2_dF.m - Partielle Ableitung erster Ordnung: $f_{\theta}(n)$ nach k_2

likelihoodCI.m

```

1 function [ sortedCI ] = profileLikelihood( data, thetaHat, alpha )
2     global S N
3
4     if nargin < 3
5         alpha = 0.95;
6     end
7
8     S = data(:,1);
9     N = data(:,2);
10    d = data(:,3);
11    defaultStepFactor = 0.1;
12    epsFactor = 1e-8;
13    eps = epsFactor;
14    maxIterations = 1000;
15    quantile = chi2inv( (1 + alpha) / 2, 1 );
16
17    theta = [thetaHat.N0, thetaHat.sigma, thetaHat.SC, thetaHat.k1,
18            thetaHat.k2];
19    sortedCI = NaN(5,2);
20    L = ell(theta);
21
22    for j = 1:5
23        f = @(x) 2*(L - ellThetaJ(theta, j, x));
24        sortedCI(j,:) = [lowerBound(f, j), upperBound(f, j)];
25    end
26
27    function [xm] = lowerBound(f, j)
28        eps = epsFactor * theta(j);
29        x_min = theta(j) * (1 - defaultStepFactor);
30
31        while f(x_min) < quantile
32            x_min = x_min - defaultStepFactor * theta(j);

```

```
32     end
33
34     xm = bisection(f, x_min, theta(j), -1);
35 end
36
37 function [xm] = upperBound(f, j)
38     eps = epsFactor * theta(j);
39     x_max = theta(j) * (1 + defaultStepFactor);
40
41     while f(x_max) < quantile
42         x_max = x_max + defaultStepFactor * theta(j);
43     end
44
45     xm = bisection(f, theta(j), x_max, 1);
46 end
47
48 function [xm] = bisection(f, x1, x2, dir)
49     for i = 1:maxIterations
50         xm = (x1 + x2) / 2;
51         ym = f(xm);
52
53         if abs(quantile - ym) < eps || abs(x1 - x2) < eps
54             return;
55         end
56
57         if ym < quantile && dir == 1
58             x1 = xm;
59         else
60             x2 = xm;
61         end
62     end
63 end
64
65 function [y] = ellThetaJ(theta, j, x)
66     theta(j) = x;
67     y = ell(theta);
68 end
69
70 function [y] = ell(theta)
71     y = 0;
72
73     for i = find(d == 1)'
74         y = y + log( dF(theta, i) );
75     end
76
```

```

77     for i = find(d == 0)
78         y = y + log( 1 - F(theta, i) );
79     end
80 end
81 end

```

Listing 20: likelihoodCI.m - Berechnung der Likelihood-Konfidenzintervalle

Woehler.cpp

```

1 void CWoehler::completeModella(CProbability conf) {
2     checkSample();
3
4     CSurvivalDataFlyweight dataFlyweight(*(Sample->getSurvivalData()));
5
6     ModellaEstimator = new CModellaEstimator();
7     ModellaEstimator->setSurvivalData(&dataFlyweight);
8     ModellaEstimator->setConfidenceIntervalSimulationSize(CIResamples);
9     ModellaEstimator->setConfidenceIntervalMethod(CIMethod);
10    ModellaEstimator->complete(conf);
11
12    readWoehlerParameterFromModellaEstimator();
13    ModellaEstimator->getCIs()->display();
14    ModellaEstimator->getBootstrapCIs()->display();
15 }

```

Listing 21: Woehler.cpp - Relevanter Ausschnitt der Wählermodell-Klasse in Jurojin

ModellITWMEstimator.cpp

```

1 void CModellITWMEstimator::complete(CProbability conf) {
2     setConstParameters();
3     setVectorParameters();
4     setSpecialParameters();
5     prepareMaximizationFunction();
6     startLoopForLogSC();
7     readEstimatedParameters();
8     prepareBootstrapVectors();
9     computeConfidenceInterval(conf);
10 }
11
12 void CModellITWMEstimator::computeConfidenceIntervalsFromBootstrapValues(
    double const &conf) {

```

```
13 ConfidenceIntervals.clear();
14 ConfidenceIntervals.resize( BootstrapParameters.size() );
15 BootstrapConfidenceIntervals.clear();
16 BootstrapConfidenceIntervals.resize( BootstrapParameters.size() );
17
18 for(size_t i = 0; i != BootstrapParameters.size(); i++) {
19     CSampleQuantile quants;
20     quants.insertSequence( BootstrapParameters[i].begin(),
21         BootstrapParameters[i].end() );
22
21     ConfidenceIntervals[i] = quants.computeConfidenceInterval(conf);
22     BootstrapConfidenceIntervals[i] = quants.bootstrapInterval(conf,
23         BootstrapValues[i]);
24 }
25 }
26
27 void CModelITWMEstimator::
28     computeJackknifeConfidenceIntervalsFromBootstrapValues(double const &
29     conf) {
30     ConfidenceIntervals.clear();
31     ConfidenceIntervals.resize( BootstrapParameters.size() );
32     BootstrapConfidenceIntervals.clear();
33     BootstrapConfidenceIntervals.resize( BootstrapParameters.size() );
34
35     CConfidenceInterval emptyInterval(0, 0);
36     CNormalDistribution F(0, 1);
37     double confFactor = F.quantile( 1 - (1 - conf)/2 );
38
39     for(size_t i = 0; i != BootstrapParameters.size(); i++) {
40         unsigned int n = BootstrapParameters[i].size();
41         double avg = 0;
42         double sE = 0;
43         double ps = 0;
44         vector<double> psi(n, 0.0);
45
46         for(size_t j = 0; j != n; j++) {
47             psi[i] = BootstrapValues[i] + (n - 1) * (BootstrapValues[i] -
48                 BootstrapParameters[i][j]);
49             ps += psi[i];
50
51             avg += BootstrapParameters[i][j];
52         }
53
54         ps /= (double)n;
55         avg /= (double)n;
```

```

53
54     for(size_t j = 0; j != n; j++)
55         sE += pow(avg - BootstrapParameters[i][j], 2);
56
57     sE = sqrt(((double)(n-1) / (double)(n) * sE);
58
59     ConfidenceIntervals[i] = CConfidenceInterval(ps - confFactor * sE,
60         ps + confFactor * sE);
61     BootstrapConfidenceIntervals[i] = CConfidenceInterval(
62         BootstrapValues[i] - confFactor * sE, BootstrapValues[i] +
        confFactor * sE);
61     }
62 }

```

Listing 22: ModellTWMEstimator.cpp - Relevanter Ausschnitt der abstrakten Klasse zum Schätzen von Modellparametern in Jurojin

Modell1aEstimator.cpp

```

1 void CModell1aEstimator::computeConfidenceInterval(CProbability conf) {
2     if( CIMethod == "Jackknife" )
3         computeConfidenceIntervalWithJackknife(conf);
4     else
5         computeConfidenceIntervalWithBootstrap(conf);
6 }
7
8 void CModell1aEstimator::computeConfidenceIntervalWithBootstrap(
9     CProbability conf) {
10     CNormalDistribution F(0, std::pow(Sigma,2));
11
12     CRandomNumberGenerator generator;
13     generator.setNoOfSamples(CISimulationSize);
14     generator.setSampleSize(CSampleSize(SurvivalData->getSampleSize()));
15     generator.setDistribution(&F);
16     generator.complete();
17     vector<double> eps = generator.getSurvivalData()->getAllLifetimes();
18     vector<string> groupValues = SurvivalData->getAllGroupValues();
19     vector<double> loadHorizons = SurvivalData->getGroupValuesAsDouble();
20     unsigned int loadHorizonsSize = loadHorizons.size();
21
22     for(unsigned int mcStep = 0; mcStep < CISimulationSize; ++mcStep) {
23         CSurvivalDataFlyweight loopSample;
24         loopSample.setSampleSize( loadHorizonsSize );

```

```

25     for(unsigned int i = 0; i < loadHorizonsSize; ++i) {
26         double predictiveValue = std::pow( 10, std::log10(
                getPredictiveValueFromLoad(loadHorizons.at(i))) + eps.at(mcStep
                *loadHorizonsSize + i) );
27         loopSample.setLifetimeValueAt(i, predictiveValue);
28         loopSample.setCensorValueAt(i, CSurvivalData::FAILURE);
29         loopSample.setGroupValueAt(i, groupValues.at(i));
30     }
31
32     findWoehlerModelForLoopSample(loopSample);
33 }
34
35 computeConfidenceIntervalsFromBootstrapValues( conf.getValue() );
36 }
37
38 void CModellaEstimator::computeConfidenceIntervalWithJackknife(
        CProbability conf) {
39     unsigned int sampleSize = SurvivalData->getSampleSize();
40     for(unsigned int i = 0; i < sampleSize; ++i) {
41         CSurvivalDataFlyweight loopSample(SurvivalData);
42         loopSample.removeRow(i);
43         findWoehlerModelForLoopSample(loopSample);
44     }
45
46     computeJackknifeConfidenceIntervalsFromBootstrapValues(conf.getValue());
47 }
48
49 void CModellaEstimator::findWoehlerModelForLoopSample(ISurvivalData &
        sample) {
50     CModellaEstimator model;
51     model.setSurvivalData(&sample);
52     model.setDefaultStartParameters(MinimumParameters);
53     model.complete();
54
55     BootstrapParameters[0].push_back( model.getSlope1() );
56     BootstrapParameters[1].push_back( model.getLogN0() );
57     BootstrapParameters[2].push_back( model.getSigma() );
58     BootstrapParameters[3].push_back( model.getSC() );
59     BootstrapParameters[4].push_back( model.getEnduranceSTD() );
60     BootstrapParameters[5].push_back( model.getLogAbknick() );
61     BootstrapParameters[6].push_back( model.getSlope2() );
62 }

```

Listing 23: ModellaEstimator.cpp - Relevanter Ausschnitt der Klasse zum Schätzen der Parameter des kombinierten Modells in Jurojin

D Abbildungsverzeichnis

Abb. 1	Beispiel einer Wöhlerlinie für eine Stichprobe	1
Abb. 2	Mögliche Formen zweidimensionaler Konfidenzbereiche	9
Abb. 3	Kennwerte eines Schwingspiels nach Haibach	11
Abb. 4	Kennwerte einer Wöhlerlinie nach Haibach	11
Abb. 5	Schematische Darstellung des kombinierten Wöhlermodells	12
Abb. 6	Grafische Darstellung des Datensatzes Verdacht auf zwei Bereiche	17
Abb. 7	Grafische Darstellung des Fiktiven Beispieldatensatzes	20
Abb. 8	Verdacht auf zwei Bereiche: geschätzte Wöhlerlinie	24
Abb. 9	Fiktiver Datensatz: geschätzte Wöhlerlinie	25

E Tabellenverzeichnis

Tab. 1	Beispieldatensatz: Verdacht auf zwei Bereiche	17
Tab. 2	Fiktiver Datensatz mit bekannten Parametern	19
Tab. 3	Verdacht auf zwei Bereiche: Geschätzte Modellparameter	24
Tab. 4	Fiktiver Datensatz: Geschätzte Modellparameter	25
Tab. 5	Verdacht auf zwei Bereiche: Bootstrap-Konfidenzintervalle	28
Tab. 6	Fiktiver Beispieldatensatz: Bootstrap-Konfidenzintervalle	28
Tab. 7	Verdacht auf zwei Bereiche: Jackknife-Konfidenzintervalle für . .	31
Tab. 8	Verdacht auf zwei Bereiche: Likelihood-Konfidenzintervalle . . .	35
Tab. 9	Fiktiver Beispieldatensatz: Likelihood-Konfidenzintervalle	35
Tab. 10	Verdacht auf zwei Bereiche: Fisher-Konfidenzintervalle	38
Tab. 11	Fiktiver Beispieldatensatz: Fisher-Konfidenzintervalle	38
Tab. 12	Verdacht auf zwei Bereiche: Vergleich der Konfidenzintervalle . .	40
Tab. 13	Fiktiver Beispieldatensatz: Vergleich der Konfidenzintervalle . . .	41
Tab. 14	Bewertung der unterschiedlichen Verfahren	41
Tab. 15	Konfidenzintervalle bei numerischer Bestimmung der Fisher Infor- mation	VII

F Listing-Verzeichnis

Lst. 1	createExample.m - Erzeugung des fiktiven Beispieldatensatzes	VIII
Lst. 2	data.m - Daten der beiden Beispieldatensätze	VIII
Lst. 3	ci.m - Berechnung der Konfidenzintervalle aus Fisher Information	IX
Lst. 4	fisherMatrix.m - Berechnung der beobachteten Fisher Information	X
Lst. 5	param.m - Berechnung der verwendeten Variablen	XI
Lst. 6	Phi.m - Berechnung von $\Phi(x)$	XII
Lst. 7	dPhi.m - Berechnung von $\phi(x)$	XII
Lst. 8	F.m - Berechnung von $F_\theta(n)$	XII
Lst. 9	dF.m - Berechnung von $f_\theta(n)$	XIII
Lst. 10	dN0_F.m - Partielle Ableitung erster Ordnung: $F_\theta(n)$ nach N_0	XIII
Lst. 11	dsigma_F.m - Partielle Ableitung erster Ordnung: $F_\theta(n)$ nach σ	XIII
Lst. 12	dSC_F.m - Partielle Ableitung erster Ordnung: $F_\theta(n)$ nach S_C	XIV
Lst. 13	dk1_F.m - Partielle Ableitung erster Ordnung: $F_\theta(n)$ nach k_1	XIV
Lst. 14	dk2_F.m - Partielle Ableitung erster Ordnung: $F_\theta(n)$ nach k_2	XIV
Lst. 15	dN0_dF.m - Partielle Ableitung erster Ordnung: $f_\theta(n)$ nach N_0	XV
Lst. 16	dsigma_dF.m - Partielle Ableitung erster Ordnung: $f_\theta(n)$ nach σ	XV
Lst. 17	dSC_dF.m - Partielle Ableitung erster Ordnung: $f_\theta(n)$ nach S_C	XVI
Lst. 18	dk1_dF.m - Partielle Ableitung erster Ordnung: $f_\theta(n)$ nach k_1	XVI
Lst. 19	dk2_dF.m - Partielle Ableitung erster Ordnung: $f_\theta(n)$ nach k_2	XVI
Lst. 20	likelihoodCI.m - Berechnung der Likelihood-Konfidenzintervalle	XVII
Lst. 21	Woehler.cpp - Relevanter Ausschnitt der Wählermodell-Klasse in Jurojin	XIX
Lst. 22	ModellITWMEstimator.cpp - Relevanter Ausschnitt der abstrakten Klasse zum Schätzen von Modellparametern in Jurojin	XIX
Lst. 23	ModellaEstimator.cpp - Relevanter Ausschnitt der Klasse zum Schätzen der Parameter des kombinierten Modells in Jurojin	XXI

Erklärung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Die Stellen der Masterarbeit, die andere Quellen im Wortlaut oder dem Sinn nach entnommen wurden, sind durch Angaben der Herkunft kenntlich gemacht.

Kaiserslautern, den 11. September 2015

(Hannes Christiansen)