universität wien

DIPLOMARBEIT

Titel der Diplomarbeit

„Ein Verfahren zur Itemkalibrierung mittels adaptiver Itemauswahl“

Verfasser

Robert Emprechtinger

Angestrebter akademischer Grad

Magister der Naturwissenschaften (Mag. rer. nat.)

Wien, 2015

Studienkennzahl lt. Studienblatt:	A 298
Studienrichtung lt. Studienblatt:	Psychologie
Betreuerin / Betreuer:	Univ.-Prof. Dr. Mag. Klaus D. Kubinger

Danksagung

Dank geht an Dipl.-Ing. Florian Reisecker, der mich intensiv bei der Entwicklung des Algorithmus und allgemeinen Fragen bezüglich Softwareentwicklung beraten hat.

Dank geht an meine Lebensgefährtin Daniela Rasl, die mir wertvolle Anregungen und Verbesserungsvorschläge für den vorliegenden Text zukommen hat lassen.

Dank geht an meinen mittlerweile dreieinhalbjährigen Sohn Leonas, der mich oft zu Pausen gezwungen hat und hoffentlich einmal Verständnis dafür entwickelt, dass ich im vergangenen Jahr leider viel zu wenig Zeit für ihn aufbringen konnte.

Dank geht an meinen Kollegen Marius Stehle, der mit seinen Anregungen mehrere Unklarheiten beseitigen konnte.

Besonderer Dank gebührt Ing. Mag. Andreas Pfaffel, der mit seinen ausführlichen Rückmeldungen bezüglich statistischer Fragen und der Hilfestellung im Verfassen von wissenschaftlichen Texten erheblich zu dem Endergebnis dieser Arbeit beigetragen hat sowie Univ.-Prof. Dr. Dr. Christiane Spiel für die Beratung zu dieser Arbeit.

Erwähnen möchte ich auch Univ.-Prof. Dr. Mag. Klaus D. Kubinger, der die formale Betreuung dieser Arbeit übernommen hat.

Abstract Deutsch

Ein wichtiger Schritt bei der Entwicklung neuer psychologisch-diagnostischer Testverfahren ist die als Itemkalibrierung bezeichnete Prüfung der Verrechnungsfairness der konstruierten Items. Ein Problem bei der Itemkalibrierung ist die zufällige Itemvorgabe (zufällig bedeutet in diesem Kontext, dass jedes Item stets die selbe Wahrscheinlichkeit hat, vorgegeben zu werden). Dementsprechend erhalten Testpersonen, bei denen sich bereits aufgrund ihres bisherigen Antwortverhaltens zum Beispiel eine besonders hohe Fähigkeit erkennen lassen würde, auch weiterhin (zufällig) sehr leichte Items zur Bearbeitung. Das Problem hierbei ist, dass die Information, die über die vorgegeben Items in der Testkalibrierung gewonnen werden könnte, nicht optimal ausgeschöpft wird. Deshalb wurde im Rahmen dieser Arbeit ein Algorithmus zur Itemkalibrierung mittels adaptiver Itemauswahl im Rasch-Modell entwickelt. Dieser ist in der Lage ein Verfahren mit gleichverteilten Itemparametern und standardnormalverteilten Personenparameter zu kalibrieren. Außerdem führt der Algorithmus bereits im Rahmen der Testkalibrierung die Modellprüfung auf Gültigkeit des Rasch-Modells mittels Fischer und Scheiblechner *z*-Test durch. Bei der Itemauswahl berücksichtigt der Algorithmus nicht nur die Item- und Personenparameter, sondern auch die bereits vorhandene Information zu jedem Item. Mittels einer Simulationsstudie wurde die Genauigkeit und Präzision nach abgeschlossener Kalibrierung sowie die Präzision zu unterschiedlichen Kalibrierungszeitpunkten erhoben und jene der adaptiven Kalibrierungsstrategie mit zufälliger Itemauswahl verglichen. Außerdem wurde überprüft, ob der Fischer und Scheiblechner *z*-Test überzufällig häufig Items mit Differential Item Functioning erkennt. Bei 37 von 60 Items war die Präzision der Itemparameter nach 500 zur Kalibrierung eingesetzten Personen bei den adaptiv kalibrierten Tests signifikant höher als bei den zufällig kalibrierten Tests. Bei den Kalibrierungsstadien nach 100 Personen schnitt hingegen die zufällige Itemkalibrierung besser ab, nach 250 Personen zeigte sich kein Unterschied zwischen den beiden Kalibrierungsstrategien. Beim Einsatz der fertig kalibrierten Tests zur Messung der Personenparameter war kein Unterschied in der Präzision der adaptiven zur zufälligen Vorgabestrategie feststellbar. Der Fischer und Scheiblechner *z*-Test konnte 95.0% der Items mit Differential Item Functioning erkennen und gab bei 2.9% der Items ein falsch positives Ergebnis aus und kann überzufällig häufig Items mit Differential Item Function erkennen. In Zukunft könnte die adaptive Itemkalibrierung eine alternative zur Itemkalibrierung mittels zufälliger Itemauswahl darstellen, damit jede Person zur Testkalibrierung optimal genutzt wird.

Schlüsselwörter: adaptiv, Itemkalibrierung, Rasch-Modell, Fischer und Scheiblechner z-Test, Simulation

Abstract English

An important step during the development of psychological assessment tests is the scaling of the items or the so called item calibration. The Item selection during the item calibration is at random (random means that every item in the pool hast the same chance to be selected). Accordingly, very easy items are still (randomly) admitted to subjects from whom is already known that they are very competent or vice versa. This is a problem, because information is not optimally exploited. Hence an adaptive item selection algorithm for test calibration has been developed. The algorithm is able to calibrate a test with normal distributed person parameters and evenly distributed item parameters. The algorithm is also able to conduct model testing with the Fischer and Scheiblechner *z*-Tetst during the item calibration process. Not only person and item parameters are considered for item selection but also the previously gathered information. The algorithm has been tested in a simulation study to measure the accuracy and precision of this calibration method compared with a random item calibration strategy. Furthermore, it was verified if the Fischer and Scheiblechner *z*-Test is able to find more often items with differential item than by simply random chance. 37 of 60 items showed a higher precision in the adaptive calibration procedure compared with the random item calibration strategy after the test has been administered to 500 examinees. At earlier calibration stages the precision in the random calibration strategy was higher (after 100 examinees) or even (after 250 examinees). There was no detectable difference in the precision of the person parameters after the calibration was completed (500 eximanees). The Fisher and Scheiblechner *z*-Test is able to detect items with differential item functioning. 95.0% of the Items with differential item functioning have been recognized and 2.9% were rated false positive. Because the information of each examinee is used in an optimal manner, the adaptive item calibration strategy could be an option for test calibration in the near future.

Keywords: adaptive, item calibration, Rasch model, Fischer and Scheiblechner *z*-test, simulation

Inhaltsverzeichnis

I. Einleitung

Ein wichtiger Schritt bei der Entwicklung neuer psychologisch-diagnostischer Testverfahren ist die Prüfung der Verrechnungsfairness der konstruierten Items. Diese Prüfung der Items (im Sinne der Probabilistischen Testtheorie) wird als Itemkalibrierung bezeichnet. Das übliche Vorgehen bei der Itemkalibrierung ist, dass eine bestimmte Anzahl an Items (aus einem Itempool) einer sehr großen Stichprobe von Testpersonen zufällig vorgegeben wird; zufällig heißt dabei, dass jedes Item stets dieselbe Wahrscheinlichkeit hat vorgegeben zu werden. Die Information über die Testperson, die durch die Beantwortung bereits vorgebebener Items während der Testkalibrierung gewonnen wird, bleibt für die Vorgabe weiterer Items während der Testkalibrierung unberücksichtigt. Dementsprechend erhalten Testpersonen, bei denen sich bereits aufgrund ihres bisherigen Antwortverhaltens zum Beispiel eine besonders hohe Fähigkeit erkennen lassen würde, auch weiterhin (zufällig) sehr leichte Items zur Bearbeitung.

Das Problem bei dieser zufälligen Vorgabe ist, dass die Information, die über die vorgegeben Items während der Testkalibrierung gewonnen werden könnte, nicht optimal ausgeschöpft wird. Zum Beispiel, wenn eine Person mit besonders hoher Fähigkeit ein sehr leichtes Items vorgegeben bekommt, so ist es sehr wahrscheinlich, dass diese Person das Item auch lösen wird. Wenn diese sehr fähige Person dann auch tatsächlich dieses leichte Item löst, ist der Zugewinn an Information über die Fähigkeit der Person gering, da dies bereits mit hoher Wahrscheinlichkeit vermutet wurde. Folglich sind mehr Personen zur Testkalibrierung notwendig, als dies bei einer Itemvorgabestrategie der Fäll wäre, welche die bereits gewonnene Information über die Fähigkeit der Testperson berücksichtigt und die Itemauswahl dementsprechend anpasst.

Die maximale Information über die Testperson (als auch über das Item) wird dadurch gewonnen, indem der Testperson ein Item mit der Schwierigkeit entsprechend ihrer Fähigkeit vorgegeben wird. Dieser Ansatz wird im sogenannten „adaptiven Testen“ im Rahmen der Item-Response-Theorie bereits genutzt. Ein bedeutender Vorteil des adaptiven Testens ist, dass die Fähigkeit der Testperson, im Vergleich zu einer klassischen Vorgabe, mit weniger Items gleich genau geschätzt wird (Kubinger, 2014a).

Nach Makransky (2009) lässt sich der Vorteil des adaptiven Testens zum maximalen Informationsgewinn über die Items bereits im Rahmen der Itemkalibrierung anwenden. Dadurch könnte die üblicherweise notwendige sehr große Stichprobe (meist über tausend Personen) im

Rahmen der Testkalibrierung deutlich reduziert werden. Makransky (2009) verglich im Rahmen einer Simulationsstudie unterschiedliche Kalibrierungsstrategien und kam zu dem Ergebnis, dass eine adaptive Vorgabe in welcher die Itemparameter nach jeder Testperson aktualisiert werden, die genauesten Ergebnisse liefert. Sowohl die Itemparameter als auch die Personenparameter entsprachen in der Studie von Makransky (2009) der Standardnormalverteilung und *Differential Item Functioning* wurde nicht simuliert. Unter Differential Item Functioning versteht man, dass ein Item in diversen Subpopulationen eine unterschiedliche Messgenauigkeit aufweisen. Dies ist üblicherweise nicht erwünscht, da hierbei die geforderte Eindimensionalität des Tests verletzt wird. Das typische Vorgehen bei der Testentwicklung ist derartige Items zu erkennen (Modellprüfung) und auszuschließen.

Ziel dieser Arbeit ist es, einen Algorithmus zu entwickeln und empirisch zu prüfen, der geeignet ist ein Testverfahren mit gleichverteilten Itemparametern und standardnormalverteilten Personenparametern adaptiv zu kalibrieren. Der Algorithmus soll außerdem Items erkennen, welche Differential Item Functioning (DIF) aufweisen und diese von der weiteren Vorgabe automatisch ausschließen. Dadurch werden Ressourcen frei, die in die vermehrte Vorgabe von modellkonformen Items investiert werden können. Durch die Anwendung dieses Algorithmus sollen bei gleicher Anzahl an zur Kalibrierung verwendeten Personen, genauere Ergebnisse erreicht werden, als dies bei einer Kalibrierung durch die sonst übliche zufällige Vorgabe der Fall ist.

II. Theoretischer Teil

Zunächst werden die Grundlagen der Probabilistischen Testtheorie mit speziellem Fokus auf das dichotom logistische Modell von Rasch (1960) dargestellt. In weiterer Folge wird kurz und in einfacher Weise auf die mathematischen Grundlagen des Modells sowie auf die Möglichkeiten der Modellprüfung eingegangen. Aus den beschriebenen Voraussetzungen des Rasch-Modells lässt sich das sogenannte *Adaptive Testen* als bedeutendes psychologisch-diagnostisches Verfahren ableiten, die in Kapitel 3 dargestellt werden. Im letzten Kapitel wird beschrieben, wie die Vorteile des adaptiven Ansatzes bereits während der Itemkalibrierung genutzt werden können.

1. Die Probabilistische Testtheorie

Die Probabilistische Testtheorie ist eine Sammlung an Modellen, die vorgeben, wie psychologisch-diagnostische Tests und andere Verfahren zu konstruieren sind (Kubinger, 2014b). Allen gemein ist, dass die manifesten (beobachtbaren) Reaktionen bzw. Antworten auf ein vorgegebenes Item durch eine latente (nicht direkt beobachtbare und nicht direkt messbare) Eigenschaft verursacht werden. Diese latente Eigenschaft der Testpersonen wird indirekt durch das Antwortverhalten auf die vorgegebenen Items ermittelt. Erfasst wird die Wahrscheinlichkeit für eine bestimme Reaktion bzw. ein bestimmtes Antwortverhalten in Abhängigkeit von der indirekt gemessenen (latenten) Eigenschaft (Kubinger, 2014b). Die so geschätzten Werte der Eigenschaft der Testpersonen werden als *Personenparameter* und jene der zu lösenden bzw. zu bearbeitenden Aufgaben als *Itemparameter* bezeichnet (Kubinger, 1989).

Die Probabilistische Testtheorie stellt durch ihre grundlegenden Modellannahmen eine Antwort auf wesentliche Schwächen der Klassischen Testtheorie dar. Dementsprechend erhielt die Probabilistische Testtheorie in der Erwartung, dass sie die Klassische Testtheorie ablösen würde, die zusätzliche Bezeichnung „Moderne Testtheorie“ (Becker, 2000; Fischer, 1974). Der wesentliche Unterschied zur Klassischen Testtheorie besteht darin, dass die Modelle der Probabilistischen Testtheorie empirisch prüfbar sind. Die Modelle der Klassischen Testtheorie sind hingegen nicht empirisch prüfbar, sondern sind als Axiome vorausgesetzt (Fischer, 1974). Abgeleitet aus den Modellannahmen ergeben sich bedeutende Vorteile der Probabilistischen Testtheorie. Zum Beispiel ist es im Rahmen der Probabilistischen Testtheorie unerheblich an welcher Stichprobe das Verfahren kalibriert wurde. Die Items werden, sofern sie modell-konform sind, in unterschiedlichen

Stichproben gleich schwer geschätzt. Außerdem ist jedes Item unabhängig vom gesamten Verfahren aussagekräftig, wodurch die Ergebnisse nicht vom Verfahren abhängig sind, sondern nur von den vorgegeben Items. Dies ermöglicht objektive Vergleiche zwischen Personen, denen völlig unterschiedliche Items zur Bearbeitung vorgegeben wurden (Bortolotti, Tezza, Andrade, Bornia, & Sousa Júnior, 2013).

2. Das dichotom logistische Testmodell von Rasch

Das dichotom logistische Testmodell des dänischen Statistikers Georg Rasch (1960) gilt als das Grundmodell der Probabilistischen Testtheorie. Im Rahmen dieser Arbeit wird dieses Modell in weiterer Folge als „Rasch-Modell“ bezeichnet. Es beschreibt die Reaktionswahrscheinlichkeit auf das Item *i* mit dem Itemparameter σ durch die Person *v* mit dem Personenparameter ξ (Kubinger, 1989). Die Verwendung des Rasch-Modells ist dabei auf nur zwei festgelegte (dichotome) Reaktionskategorien beschränkt. Es wird deshalb am häufigsten bei Verfahren eingesetzt, bei denen die Antworten auf die Items entweder mit richtig oder falsch bewertet werden (z.B. bei Leistungstests). Items, die seltener richtig beantwortet werden als andere Items sind schwieriger. Personen, die gegenüber anderen Personen auch schwierigere Items richtig beantworten können sind fähiger. Aus diesem Grund verwendet man für den Itemparameter σ häufig die Bezeichnung „Schwierigkeitsparameter“ sowie für den Personenparameter ξ die Bezeichnung „Fähigkeitsparameter“. Eine weitere Annahme des Modells besteht darin, dass sowohl die Fähigkeit der Person als auch die Schwierigkeit des Items eindimensional und deshalb nur mit jeweils einem Parameter zu charakterisieren sind (deshalb auch die alternative und vor allem im angelsächsischen Raum verbreitete Bezeichnung „1PL – Model“). Die Wahrscheinlichkeit, ob eine Person ein Item löst oder nicht löst, ist dementsprechend ausschließlich abhängig von der Fähigkeit der Person und der Schwierigkeit des Items (und nicht etwa von einer weiteren Dimension, wie zum Beispiel dem Geschlecht, Alter oder Kultur). Die Lösungswahrscheinlichkeit ist allerdings nicht davon abhängig, welche anderen Aufgaben die Person bereits gelöst hat oder noch lösen wird. Die Reaktionen sind somit *lokal stochastisch unabhängig* (Kubinger, 2014c). Dieses Kriterium wäre beispielsweise bei Verfahren verletzt, bei welchen während der Bearbeitung Lernprozesse stattfinden, die zur Lösung der noch nicht bearbeiteten Aufgaben hilfreich sind und dadurch die Lösungswahrscheinlichkeit der folgenden Aufgaben beeinflussen (Kubinger, 1989).

Bei Gültigkeit des Rasch-Modells (Kapitel II 2.2) sind Vergleiche zwischen zwei Personen auch dann möglich, wenn keine Information darüber vorliegt, welche konkreten Items diese

Personen bearbeitet haben. Diese Paarvergleiche gelten in gleicherweise für die Items. Es ist möglich zu behaupten, dass Item 1 schwerer ist als Item 2, ohne zu berücksichtigen von welchen Personen diese beiden Items bearbeitet wurden. Dies wird als *spezifisch objektive Vergleiche* bezeichnet (Kubinger, 1989). Außerdem ist für die Schätzungen der Parameter bei Gültigkeit des Rasch-Modells irrelevant, welche Stichprobe zur Kalibrierung herangezogen wurde. Dies wird als *Stichprobenunabhängigkeit* der Parameter bezeichnet.

Die Darstellung der Lösungswahrscheinlichkeit der Person *v* mit dem Fähigkeitsparameter ξ bei Bearbeitung des Items *i* mit dem Schwierigkeitsparameter σ erfolgt nach folgender Formel (Fischer, 1974):

$$P(+|\xi_v, \sigma_i) = \frac{e^{\xi_v - \sigma_i}}{1 + e^{\xi_v - \sigma_i}} \quad (1)$$

Formel 1 zeigt, dass je höher die Fähigkeit ξ der Person *v* und je leichter die Aufgabenschwierigkeit σ des Items *i* ist, desto näher liegt die Lösungswahrscheinlichkeit bei 1. Für besonders schwere Items und weniger leistungsfähigen Personen strebt die Lösungswahrscheinlichkeit dementsprechend gegen 0 (Kubinger, 1989).

Der Zusammenhang zwischen Schwierigkeit, Fähigkeit und Lösungswahrscheinlichkeit lässt sich am besten anhand einer *Item Characteristic Curve* (ICC) darstellen. Die Schwierigkeit eines Items bestimmt die Lage dieses Items auf der ICC. Die Steigung der Kurve und somit die Eigenschaft, wie stark ein Item zwischen fähigen und weniger leistungsfähigen Personen differenziert, ergibt sich aufgrund des *Diskriminationsparameters* (Baker, 2001). Dieser ist allerdings beim Rasch-Modell konstant (Hambleton, Swaminathan, & Rogers, 1991; Kubinger, 2014b) und somit für die vorliegende Arbeit nicht weiter von Bedeutung. Da die Lösungswahrscheinlichkeit im Rasch-Modell ausschließlich vom Schwierigkeitsparameter σ und Fähigkeitsparameter ξ abhängt und somit bei sehr geringem ξ und sehr hohem σ asymptotisch gegen 0 strebt, muss als Voraussetzung gelten, dass ein richtiges Ergebnis nicht durch bloßes Raten erzielt werden kann (Hambleton et al., 1991). Dementsprechend zeigt die ICC im Rasch-Modell für die verschiedenen Items S-förmige Kurven (Abbildung 1), die asymptotisch gegen 0 und 1 streben und parallel verlaufen (Fischer, 1974).

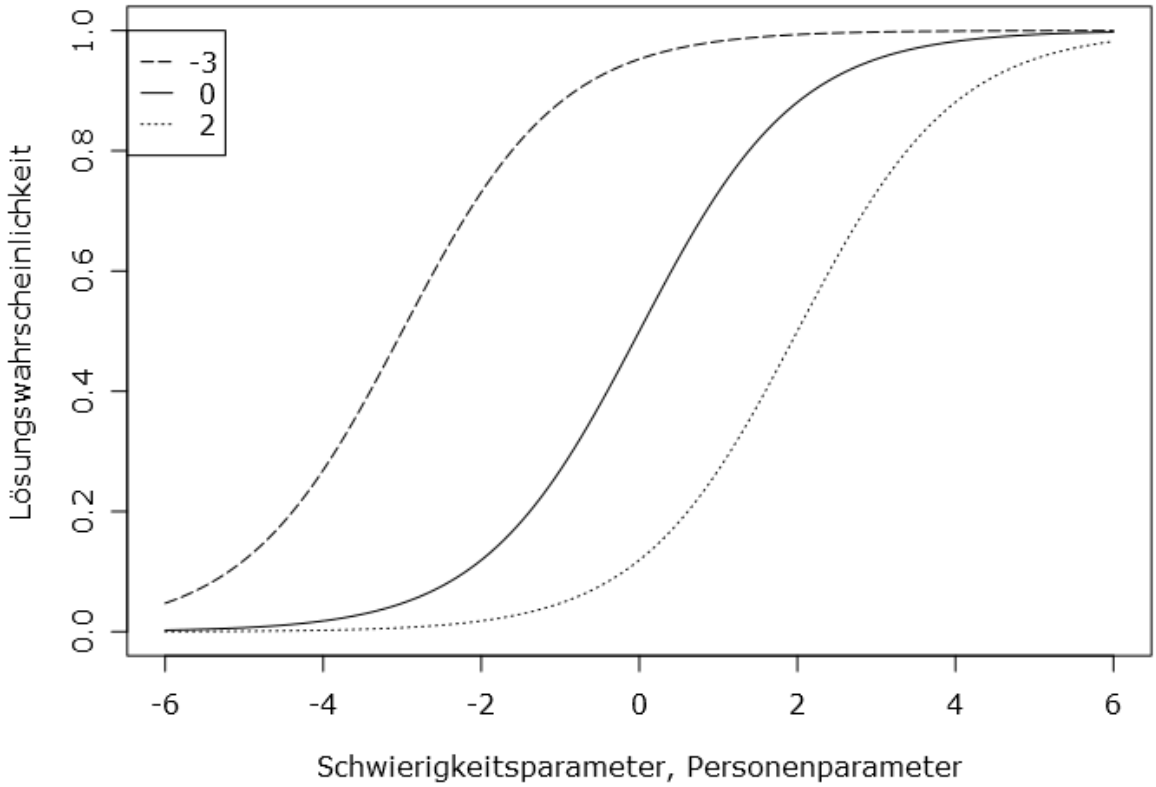


Abbildung 1: Die S-förmigen ICCs drei verschiedener Items mit den Schwierigkeitsparametern σ -3, 0 und 2 und variablen Personenparameter ξ. Die Lösungswahrscheinlichkeit beträgt .5 wenn σ = ξ, also wenn die Person gleich fähig, wie das Item schwer ist.

2.1. Parameterschätzung

Das Ziel der Parameterschätzung ist, von einem konkreten und bekannten Antwortverhalten auf die Fähigkeits- (ξ) und Itemparameter (σ) zu schließen. Erschwert wird die Zielerreichung durch die Tatsache, dass meistens sowohl ξ als auch σ unbekannt sind und dass es sich um einen nichtlinearen Zusammenhang handelt (Hambleton et al., 1991).

Für die Parameterschätzungen innerhalb der Probabilistische Testtheorie stehen verschiedene Verfahren zur Verfügung, die sich anhand ihrer unterschiedlichen Stärken und Schwächen unterscheiden. Für die vorliegende Arbeit sind vor allem die Conditional Maximum Likelihood, Marginal Maximum Likelihood und Weighted Maximum Likelihood von Bedeutung.

2.1.1. Fähigkeitsparameter

Hambleton et al. (1991) geben eine detaillierte Beschreibung der Berechnung der Item- bzw. Personenparameter, die in weiterer Folge nur ausschnittsweise und beginnend mit den Fähigkeitsparametern, wiedergegeben werden soll. Für eine genauere Darstellung sei auf den Originaltext verwiesen.

Die Wahrscheinlichkeit für das Antwortverhalten durch eine Person mit der Fähigkeit ξ und den Antworten U auf die k Items j stellt sich bei Annahme lokaler stochastischer Unabhängigkeit folgendermaßen dar:

$$P(U_1, U_2, \ldots, U_k|\xi) = \prod_{j=1}^{k} P(U_j|\xi) \tag{2}$$

Aufgrund der dichotomen Auswertung (0 oder 1) lässt sich schließen:

$$P(U_1, U_2, \ldots, U_k|\xi) = \prod_{j=1}^{k} P_j^{U_i} Q_j^{1-U_j} \tag{3}$$

$$P_j = P(U_j|\xi) \qquad \& \qquad Q = 1-P(U_j|\xi) \tag{4}$$

Die Wahrscheinlichkeit für ein beobachtetes Antwortmuster wird als Likelihood-Funktion bezeichnet und folgendermaßen dargestellt:

$$L(u_1, u_2, \ldots, u_k|\xi) = \prod_{j=1}^{k} P_j^{u_i} Q_j^{1-u_j} \tag{5}$$

Diese kann bei bekanntem Itemparameter berechnet werden und soll die Frage beantworten, welcher Personenparameter den erhaltenen Daten am besten entspricht. Damit die Derivate der Produktsumme leichter bestimmt werden können, wird die Likelihood-Funktion in aller Regel logarithmiert:

$$\ln L(u|\xi) = \sum_{j=1}^{k} [u_j \ln P_j + (1-u_j)\ln(1-P_j)] \tag{6}$$

Grafisch lässt sich diese Log-Likelihood-Funktion wie in Abbildung 2 darstellen.

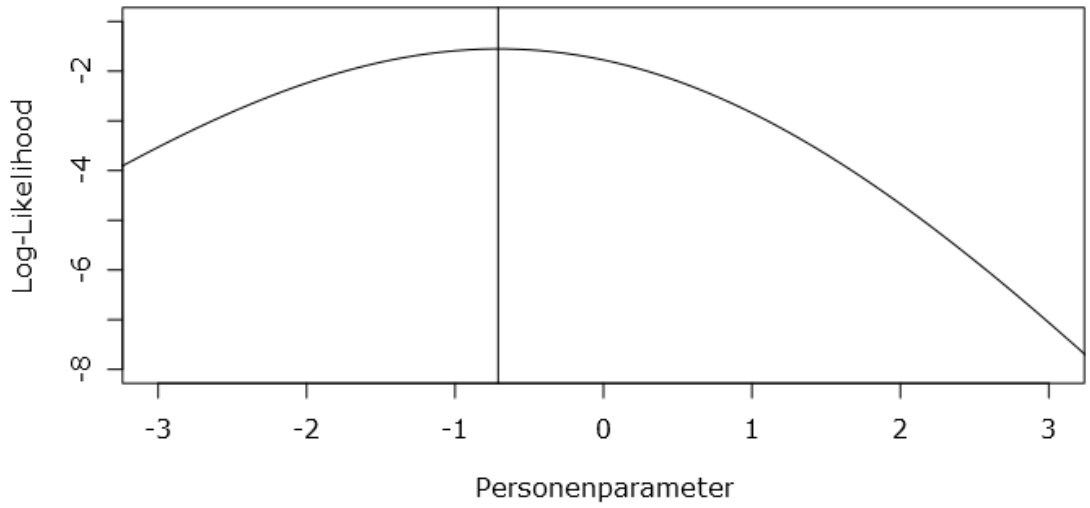


Abbildung 2: Die Log-Likelihood einer Person die 5 Items mit den Schwierigkeitsparametern -2, -1, 0, .4 und 2 beantwortet hat. Lediglich die beiden leichtesten Items wurden gelöst, weshalb die Funktion ihren höchsten Wert bei -1,55 erreicht.

Zu Problemen führt die Log-Likelihood-Funktion, wenn von der Person alle Items gelöst bzw. nicht gelöst werden. Hier strebt der Maximalwert gegen unendlich, weshalb dessen Schätzung mit den Methoden der klassischen Statistik (im Gegensatz zur Bayesianischen) nicht möglich ist (Abbildung 3).

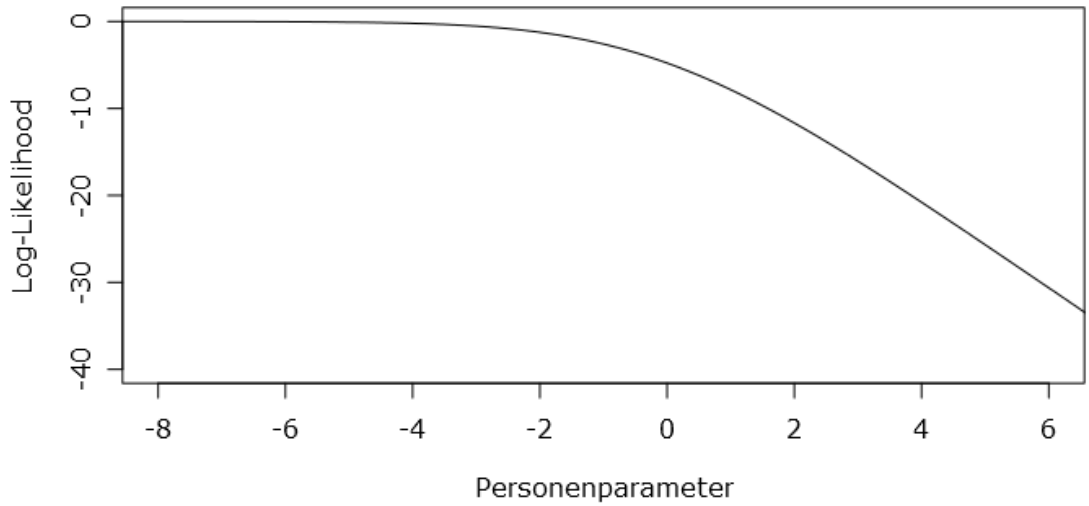


Abbildung 3: Keine der Aufgaben wurde gelöst. Ein eindeutiger Maximalwert der Funktion und somit der entsprechende Personenparameter, lässt sich nicht feststellen.

2.1.2. Itemparameter

Da die Item- bzw. Schwierigkeitsparameter nur bei bereits kalibrierten Verfahren bekannt sind, müssen diese während der Verfahrensentwicklung zusätzlich zu den Fähigkeitsparametern geschätzt werden.

Für den (unwahrscheinlichen) Fall, dass die Personenparameter bereits bekannt sind und lediglich die Itemparameter berechnet werden müssen, ist dies für das Rasch-Modell so durchzuführen wie bereits in Kapitel 2.1.1 dargestellt, lediglich mit dem Unterschied, dass die Personenparameter mit den Itemparametern getauscht werden müssen.

2.1.3. Kombinierte Schätzung der Item- und Fähigkeitsparameter

Häufig sind weder die Item- noch die Personenparameter bekannt. Aus diesem Grund ist es oft notwendig, beide zu berechnen. Bei vorliegendem Antwortverhalten von *N* Testpersonen sowie *k* Items unter der Annahme der lokalen stochastischen Unabhängigkeit, gilt für das Rasch-Modell:

$$L(u_1, u_2, \dots, u_N | \vec{\xi}) = \prod_{i=1}^{N} \prod_{j=1}^{k} P_{ij}^{u_{ij}} Q_{ij}^{1-u_{ij}} \tag{7}$$

In Formel 7 entspricht u_i dem Antwortverhalten der Person *i* auf *k* Items und $\vec{\xi}$ ist der Vektor von *N* Personenparametern. Das Problem an dieser Funktion ist, dass die Personenparameter keine vorgegebene Skalierung aufweisen und sich somit in weiterer Folge für die Likelihood-Funktion kein Maximum feststellen lässt. Deshalb verwendet man für die *N* Personenparameter in aller Regel willkürlicherweise den Mittelwert 0 und die Standardabweichung 1. Darauf aufbauend können unterschiedliche Schätzverfahren angewendet werden, die sich anhand verschiedener Vor- und Nachteile unterscheiden. Folgend werden drei wichtige Verfahren beschrieben, die auch für die vorliegende Arbeit relevant sind: 1) Joint Maximum Likelihood (JML) & Marginal Maximum Likelihood (MML), 2) Conditional Maximum Likelihood (CML), und 3) Weighted Maximum Likelihood (WML).

2.1.4. Joint Maximum Likelihood (JML) & Marginal Maximum Likelihood (MML)

Die JML-Schätzung ist ein zweistufiges Verfahren. In der ersten Stufe werden (vorläufige) Werte für die Fähigkeitsparameter festgelegt. Hierfür wird das logarithmierte Verhältnis der Anzahl

der gelösten zu der Anzahl der ungelösten Items verwendet. Anschließend werden diese Ergebnisse mit Mittelwert 0 und Standardabweichung 1 standardisiert. In der zweiten Stufe können die Fähigkeitsparameter bereits als gegeben betrachtet werden und die weitere Berechnung der Itemparameter kann, wie vorab (Kapitel II 2.1.3) beschrieben, durchgeführt werden. Die Schätzwerte der Itemparameter dienen in weiterer Folge zur erneuten Berechnung der Personenparameter, welche wiederum zur Berechnung der Itemparameter herangezogen werden. Dieser Vorgang wird so lange wiederholt bis sich die Schätzwerte zwischen den Schritten (Iterationen) nur noch unerheblich unterscheiden (Hambleton et al., 1991).

Der Vorteil der JML-Schätzung liegt darin, dass das Verfahren leicht nachvollziehbar ist. Außerdem werden bei der computerisierten Schätzung mittels JML verhältnismäßig wenig Ressourcen benötigt. Dadurch sind umfangreiche Simulationsstudien selbst auf eher leistungsschwachen Computern möglich.

Ein Nachteil der JML-Schätzung ist, dass diese zu inkonsistenten Ergebnissen führt, weil sich bei der Schätzung weniger relevanter Itemparameter viele überflüssige Personenparameter störend auf die Genauigkeit auswirken (Hambleton et al., 1991; Jansen, Wollenberg, & Arnold, 1988). Dies ist auf ein Problem zurückzuführen, welches erstmals von Neyman & Scott (1948) beschrieben wurde: Wenn bei einer JML-Schätzung die zugrunde liegenden beobachtbaren Variablen nicht nur aus Einheiten besteht, die tatsächlich zum Ergebnis der JML-Schätzung beitragen, führt dies zu einem Bias, der weder bei einer unendlich großen Stichprobe gegen Null strebt, noch durch eine einfache Adjustierung korrigierbar ist.

Aus diesem Grund wird statt der JML mittlerweile bevorzugt die Marginal Maximum Likelihood (MML) Methode angewendet, in der die störenden Parameter herausintegriert werden. Die MML-Schätzung bezahlt die bessere Genauigkeit allerdings mit einem erhöhten Rechenaufwand, was Simulationsstudien zeitaufwändiger gestaltet als dies bei JML der Fall wäre. Ein weiteres Problem ist, dass eine hohe Anzahl an Testpersonen und Items erforderlich ist, um mit dieser Methode eine ausreichend genaue Schätzung zu erhalten (Hambleton et al., 1991).

2.1.5. Conditional Maximum Likelihood (CML)

Eine weitere Alternative zur Behebung des Problems der inkonsistenten Ergebnisse der JML-Schätzung ist die CML-Schätzung. Wenngleich dieses Verfahren den testtheoretischen Vorteil besitzt, dass eine Modellprüfung anstelle eines Anpassungstests durchgeführt werden kann (Kubinger, Steinfeld, Reif, & Yanagida, 2012), findet es außerhalb Europas nur wenig Verbreitung.

Allerdings ist für die CML-Schätzung ist ein höherer Rechenaufwand (Baker & Seock-Ho, 2004) erforderlich. Deshalb ist dieses Schätzverfahren eher auf leistungsfähige Computer beschränkt bzw. für Simulationen von nur geringem bis mäßigem Umfang geeignet. Außerdem weisen die Ergebnisse der CML-Schätzung einen Bias der Item-Parameter-Schätzer auf, wenn die Vorgabe adaptiv erfolgt (Kubinger et al., 2012).

2.1.6. Weighted Maximum Likelihood (WML)

Einen relativ neuen Ansatz, für die Berechnung der Personenparameter, stellt die WML-Schätzung nach Warm (1989) dar. Diese besticht durch ihre ressourcenschonende Berechnung und den verhältnismäßig genauen Ergebnissen, vor allem auch beim adaptiven Testen (Kapitel 3). Allerdings ist bisher kaum Literatur vorhanden, welche dieses Schätzverfahren mit den bereits besser etablierten Verfahren, wie zum Beispiel der CML, vergleicht. Ebenso stellt Software zur Berechnung der WML noch eher die Ausnahme dar.

2.2. Modelltests und Differential Item Functioning

Die Frage, wie letztendlich für ein Verfahren ausreichend belegt werden kann, dass das Rasch-Modell gilt, ist derzeit nicht eindeutig geklärt und es mangelt an einem festen Schema zur Prüfung der Modellgültigkeit. Kubinger & Draxler (2007) schlagen deshalb vor, die Bewährung des Modells zu prüfen. Das ist dann der Fall, wenn wiederholt für ein konkretes Verfahren die Gültigkeit des Rasch-Modells nicht falsifiziert werden konnte. Dazu sind Vergleiche mit anderen Stichproben notwendig, was in der Praxis allerdings nur selten durchgeführt wird. Stattdessen begnügt man sich häufig mit den Ergebnissen der ersten Kalibrierungsstichprobe und schließt anhand dieser jene Items aus, die dem Rasch-Modell widersprechen. Da die meisten der Tests zur Erkennung nicht-konformer Items auf dem Kriterium der Stichprobenunabhängigkeit fußen (Kubinger, 1989), ist dieses Vorgehen allerdings problematisch.

Im Zusammenhang mit der Stichprobenunabhängigkeit wird auch häufig der Begriff „Differential Item Functioning“ (DIF) verwendet. DIF liegt vor, wenn Items in verschiedenen Gruppen, die sich durch eine Moderatorvariable (z. B. Geschlecht oder Nationalität) unterscheiden, unterschiedliche Parameter erhalten. Dies würde bedeuten, dass das Item nicht nur die zu messende Dimension, sondern auch die Moderatorvariable misst. Dasselbe Item kann dementsprechend in der einen Gruppe als schwer, in der anderen als leicht eingestuft werden. Kurz: die Items *funktionieren* in Abhängigkeit von der Moderatorvariable unterschiedlich (Wirtz, 2014). DIF stellt somit einen

Widerspruch zur postulierten Stichprobenunabhängigkeit des Rasch-Modells dar. Um die Gültigkeit des Rasch-Modells für das jeweilige Verfahren zu erhalten, müssen deshalb Items mit DIF ausgeschlossen werden.

Eine Alternative zur Modellprüfung und in weiterer Folge dem Ausschluss von Items mit DIF, stellt die Verwendung gruppenspezifischer Itemparameter dar (Makransky, 2012). Wenngleich diese Vorgehensweise zumindest auf den ersten Blick attraktiv erscheint, so ist sie dennoch nicht unproblematisch. Um ähnlich genaue Itemparameterschätzungen zu erhalten, benötigt man nämlich eine höhere Anzahl an Testpersonen, da die Itemparameter für jede Gruppe explizit geschätzt werden müssen. Außerdem darf bei dieser Vorgehensweise natürlich nicht von einer Gültigkeit des Rasch-Modells ausgegangen werden. Es ist nämlich nicht mehr unerheblich, in welcher Gruppe der Test kalibriert wurde. Das Kriterium der Stichprobenunabhängigkeit ist damit verletzt. Gleiche Testergebnisse dürfen, um die Testfairness sicherzustellen, nicht mehr gleich interpretiert werden und führen letztendlich zu unterschiedlichen Konsequenzen. Institutionen, die eine Auswertung nach Gruppenzugehörigkeit durchführen, setzen sich der Gefahr von öffentlicher Kritik und kostspieligen Klagen aufgrund von Diskriminierung aus. Insbesondere ist dies der Fall, wenn bei der Verrechnung der Ergebnisse testtheoretische Erkenntnisse nicht ausreichend berücksichtigt werden. Sehr eindrucksvoll war dies zuletzt an den Reaktionen auf den Eignungstest für das Medizinstudium in Österreich (EMS) zu sehen (Spiel, Schober, & Litzenberger, 2008; Wikipedia, 2014)

2.2.1. Grafische Modellkontrolle

Bei diesem Verfahren werden die Itemparameter aus zwei verschiedenen Stichproben bzw. aus einer, anhand eines Kriteriums (z.B. männlich vs. weiblich oder hoher vs. niedriger Score), geteilten Stichprobe auf zwei normal aufeinander stehenden Achsen aufgetragen. Items, die in beiden Gruppen dieselben Parameter erzielen, liegen exakt auf einer 45°-Geraden. Deutliche Abweichungen von dieser Linie bedeuten, dass vermutlich DIF vorliegt (Abbildung 4) (Kubinger, 1989). Es lässt sich durch die grafische Modellkontrolle alleine allerdings nicht erkennen, ob eine Abweichung auch statistisch signifikant ist und das Item tatsächlich ausgeschlossen werden sollte.

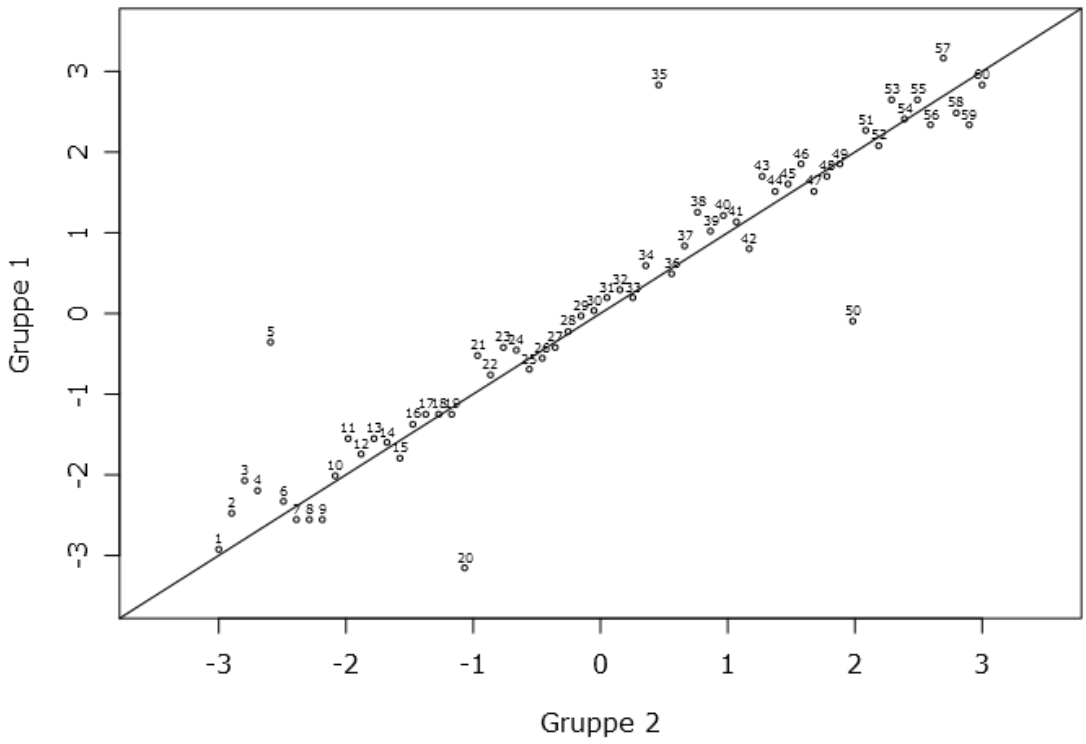


Abbildung 4: Die Items 5, 20, 35, und 50 weisen eine deutliche Abweichung von der 45°-Geraden auf. Hier ist DIF zu anzunehmen.

2.2.2. Likelihood-Quotienten-Test (LQT)

Eine weitere Möglichkeit zur Prüfung der Modellgültigkeit stellt der Likelihood-Quotienten-Test nach Andersen (1973) dar. Dieser setzt voraus, dass die Parameter anhand der CML geschätzt wurden. Ein signifikantes Ergebnis bedeutet, dass das Verfahren vom Rasch-Modell abweicht und DIF vorliegt. Welche Items exakt betroffen sind, lässt sich durch den LQT allerdings nicht erkennen. Aus diesem Grund wird häufig Andersens Likelihood-Quotienten-Test mit der grafischen Modellkontrolle kombiniert und jeweils das Item mit der höchsten Abweichung von der 45°-Geraden ausgeschlossen. Dieser Vorgang wird so lange wiederholt, bis der LQT nicht mehr signifikant ist (Kubinger, 1989).

2.2.3. Fischer und Scheiblechner z-Test

Eine Möglichkeit für die Modellprüfung auf Itemebene ist der Fischer und Scheiblechner *z*-Test (dieses Verfahren ist vor allem im angelsächsischen Raum unter der Bezeichnung *Lord's Wald Test besser bekannt*). Mit diesem lässt sich mit Hilfe der Schätzfehlervarianz $VAR(\sigma_i)$ (Kapitel II

3.2) und der Itemparameterschätzer σ_i der jeweiligen Gruppen F und R eine Testgröße z_i für das Item *i* erstellen (Langer, 2008):

$$z_i = \frac{\hat{\sigma}_R - \hat{\sigma}_F}{\sqrt{VAR(\hat{\sigma}_{R_i}) + VAR(\hat{\sigma}_{F_i})}} \quad (8)$$

Da allerdings nicht nur eine einzige Signifikanzprüfung für das gesamte Verfahren durchgeführt wird, sondern so viele, wie das Verfahren an Items hat, ist davon auszugehen, dass bei vielen Items fälschlicherweise DIF festgestellt wird. Aus diesem Grund stellen Kubinger, Rasch, & Yanagida (2011) dem Fischer und Scheiblechner *z*-Test folgendes Zeugnis aus: „Zwar ist die Testgröße *z* asymptotisch standardnormalverteilt, doch taugt die große Anzahl solcher Werte wegen der extremen Überhöhung des Risikos 1. Art nicht als Signifikanztest“ (p. 557). Diese Überlegungen sind naheliegend, da bei einer angestrebten Irrtumswahrscheinlichkeit von 5% schon bei einem relativ kleinen Verfahren mit lediglich 40 (modellkonformen) Items davon auszugehen ist, dass bei durchschnittlich zwei Items fälschlicherweise DIF festgestellt wird. Dementsprechend sollte die Irrtumswahrscheinlichkeit des Fischer und Scheiblechner *z*-Test an die Erfordernisse des zu entwickelnden Verfahrens angepasst angepasst werden. Die Eigenschaft, dass die Modellprüfung auf Itemebene anstatt auf Verfahrensebene stattfindet, macht dieses Verfahren hingegen besonders attraktiv für eine adaptive Itemkalibrierung im Sinne der vorliegenden Arbeit (Kapitel II 4.2).

3. Adaptives Testen

Adaptiv leitet sich vom lateinischen Wort adaptare = anpassen ab. Adaptives Testen entspricht einer Testvorgabe, die sich an der Leistung der Testperson orientiert. Der Test ist also an die bisherigen Ergebnisse der jeweiligen Person angepasst. Dem liegt der Gedanke zugrunde, dass die Vorgabe von Items, welche die Person mit wissentlich sehr hoher Wahrscheinlichkeit in eine Richtung (richtig oder falsch) beantworten wird, nur wenig informativ ist (Kapitel II 3.1). Damit eine adaptive Vorgabe möglich ist, muss das Verfahren anhand der Probabilistische Testtheorie entwickelt werden. Durch die Vorgabe der für die jeweilige Person informativsten Items können die Fähigkeitsparameter dieser Person mit weniger Items gleich genau geschätzt werden, was zu einer erhöhten Testökonomie führt (Kubinger, 2014a).

Vor allem beim Testen in den Extrembereichen (z.B. besonders fähige Personen) sind durch adaptives Testen genauere Ergebnisse erzielbar, da anders als bei einem konventionellen Test, nicht die Items des Durchschnittsbereichs die Mehrzahl der vorgegebenen Aufgaben darstellen. Je nach

Leistung der Testperson können hierbei nämlich durchaus auch die Items der Extrembereiche den Schwerpunkt der Vorgabe ausmachen (Kubinger, 2009).

3.1. Information

Dass ein Item besonders informativ ist, wenn es der Fähigkeit der Person entspricht, soll an folgendem Beispiel verdeutlicht werden: man erfährt nur wenig über die sprachlichen Fähigkeiten eines Kleinkinds, wenn die Aufgaben des Tests für wesentlich ältere Schulkinder konzipiert wurden, da schon a priori davon ausgegangen werden kann, dass das Kleinkind diese nicht lösen kann. Die Vorgabe solcher Items liefert aufgrund der schlechten Passung zwischen Fähigkeit und Schwierigkeit nur wenig Information über das zu untersuchende Kleinkind.

Mathematisch lässt sich die *statistische Information* der Schwierigkeit des Items *i* in Bezug auf die Fähigkeit der Person *v* und umgekehrt der Person *v* in Bezug auf den Itemparameter *i* im Rahmen des Rasch-Modells mit folgender Funktion ausdrücken (Fischer, 1974):

$$I = \frac{e^{\xi_v - \sigma_i}}{\left(1 + e^{\xi_v - \sigma_i}\right)^2} \tag{9}$$

Abbildung 5 zeigt, dass je geringer der Unterschied zwischen Fähigkeit ξ und Schwierigkeit σ ist, desto mehr Information wird durch die Bearbeitung des jeweiligen Items von der bestimmten Person erhalten. Die maximale Menge an Information von .25 wird generiert wenn ξ = σ.

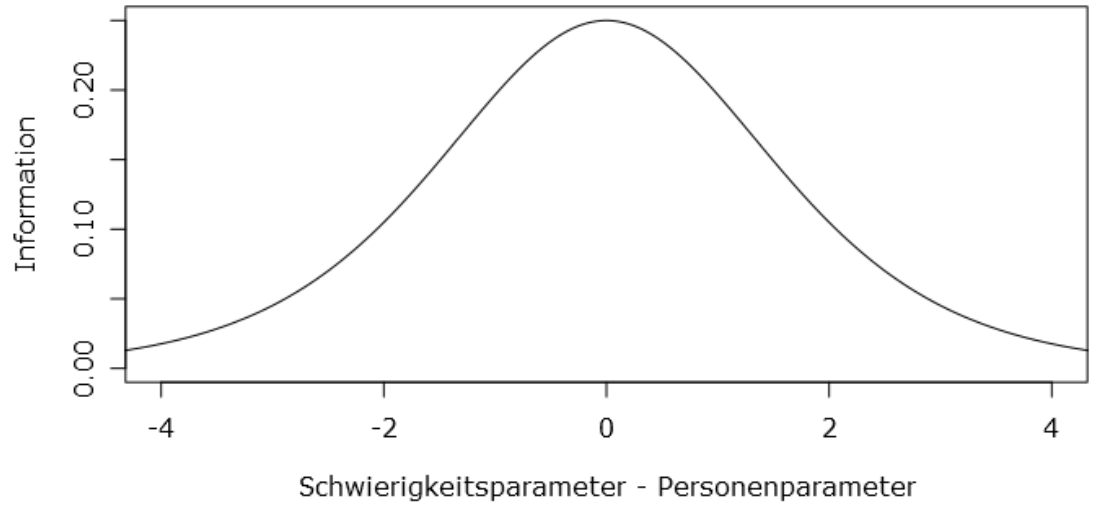


Abbildung 5: Die Informationsfunktion: Je geringer der Unterschied zwischen Fähigkeit und Itemschwierigkeit, desto mehr Information.

Zur Schätzung eines unbekannten Parameters in der Statistik ist es erforderlich, dass über diesen Parameter Information akkumuliert werden kann. Wie Fischer (1974) zeigt, kann im Rahmen der Parameterschätzung innerhalb der Probabilistische Testtheorie die gewonnene Information *I* der Personen *n* über das Item *i* summiert werden (selbstverständlich gilt das auch umgekehrt für die Information der Items *k* über die Person *v)*:

$$I(\sigma_i) = \sum_{v=1}^{n} I_v(\sigma_i) \tag{10}$$

3.2. Schätzfehlervarianz und Konfidenzintervalle

Aus der summierten Information ergibt sich, mit welcher Genauigkeit die Item- und Personenparameter geschätzt werden können. So ist die Schätzfehlervarianz des Items *i* der Kehrwert der gesamten Information des Items *i*:

$$VAR(\sigma_i) = \frac{1}{I(\sigma_i)} \tag{11}$$

Der Standardfehler ist lediglich die Wurzel der Schätzfehlervarianz. Mit diesen Informationen lässt sich ein Konfidenzintervall, welches mit einer bestimmten Wahrscheinlichkeit (abhängig von *z*) den wahren Wert des Itemparameters σ umschließt, berechnen (Fischer, 1983):

$$\hat{\sigma}_i - z_\alpha\sqrt{VAR(\sigma_i)} \leqslant \sigma_i \leqslant \hat{\sigma}_i + z_\alpha\sqrt{VAR(\sigma_i)} \tag{12}$$

4. Itemkalibrierung

Bei der Entwicklung eines Verfahrens existiert bestenfalls eine grobe Vorstellung darüber, wie schwierig ein Item tatsächlich ist. Um die Werte der tatsächlichen Itemparameter zu schätzen zu können, müssen diese an Versuchspersonen vorgegeben werden. In weiterer Folge können die beschriebenen Schätzverfahren (JML, MML und CML) zur Berechnung der Itemschwierigkeiten eingesetzt werden. Dieser Prozess wird als Itemkalibrierung bezeichnet.

4.1. Zufällige Itemkalibrierung

Das übliche Vorgehen bei der Itemkalibrierung ist, dass eine bestimmte Anzahl an Items (aus einem Itempool) den Testpersonen zufällig vorgegeben wird (zufällig bedeutet, dass jedes Item stets

dieselbe Wahrscheinlichkeit hat vorgegeben zu werden). Unabhängig von ihrer Fähigkeit erhalten Testpersonen sehr leichte bis sehr schwere Items.

Die Information über die Fähigkeit einer Testperson, die durch die Beantwortung bereits vorgegebener Items während der Testkalibrierung gewonnen wird, bleibt für die Vorgabe weiterer Items unberücksichtigt. Dadurch wird die maximale zu gewinnende Information über die Testperson (als auch über das Item) nicht optimal ausgeschöpft. Wie im Kapitel 3 Adaptives Testen beschrieben, wird die meiste Information über eine Person sowie über ein Item dann gewonnen, wenn Itemparameter und Fähigkeitsparameter optimal passend sind. Dies wird durch die zufällige Itemvorgabe nicht realisiert. Folglich sind mehr Personen zur Testkalibrierung notwendig, als dies bei einer Itemvorgabestrategie der Fall wäre, welche die bereits gewonnene Information über die Fähigkeit der Testperson berücksichtigt und die Itemauswahl dementsprechend anpasst.

4.2. Adaptive Itemkalibrierung

Der Vorteil, dass durch eine optimale Passung der Item- und Personenparameter die meiste Information gewonnen wird, kann durch eine adaptive Itemvorgabe bereits während der Itemkalibrierung genutzt werden (Makransky, 2009). Dadurch können während der Kalibrierung mit weniger Personen gleich genaue Ergebnisse erzielt werden. Makransky (2009) verglich im Rahmen einer Simulationsstudie unterschiedliche Kalibrierungsstrategien im Rahmen des Rasch- und 2PL-Modells (letzteres berücksichtigt zusätzlich den Diskriminationsparameter) mit dem Ziel anhand von möglichst wenig Items maximal genaue Schätzungen der Personenparameter zu erhalten.

Makransky (2009) zeigte, dass durch eine adaptive Itemvorgabe, bei der die Itemparameter nach jeder Testperson neu berechnet werden, die genauesten Ergebnisse erzielt werden. Schon nach 500 simulierten Testpersonen in einem Itempool $k = 100$ und der Vorgabe von je 20 Items pro Person konnte fast so genau gemessen werden wie mit 4000 simulierten Testpersonen und zufälliger Itemvorgabe. Da eine adaptive Vorgabe ohne Information über die Schwierigkeit der Items nicht möglich ist, hat Makransky (2009) verschiedene Einstiegspunkte geprüft bis zu welchem die Items zufällig ausgewählt wurden (10; 25; 50; 100; 200 Personen). Der Einstiegspunkt mit 25 Personen erzielte die genauesten Ergebnisse. Für die Berechnung der Itemparameter verwendete Makransky (2009) die MML-Schätzung (Kapitel II 2.1.4) und für die Personenparameter die WML-Schätzung (Kapitel II 2.1.6).

Makransky (2009) erhob in seiner Simulationsstudie die Abweichung der Personenparameterschätzer von den wahren Werten. Da sein Verfahren zur Online-Itemkalibrierung (der zu entwickelnde Test kann bereits während der Kalibrierungsphase zu diagnostischen Zwecken eingesetzt werden) konzipiert ist, wurde die Abweichung nur anhand von den zur Testkalibrierung herangezogenen Personen geschätzt. Angaben über die Ergebnisse bzw. die Genauigkeit der geschätzten Itemparameter fehlen. Lediglich die Häufigkeit der Itemvorgabe wurde erhoben, mit dem Ergebnis, dass in der kontinuierlichen Vorgabestrategie im Rasch-Modell alle Items etwa gleich häufig vorgegeben wurden. Dies ist der Tatsache geschuldet, dass sowohl Itemparameter als auch Personenparameter standardnormalverteilt waren, wodurch eine adaptive Vorgabe nach dem Kriterium der maximalen Information ($\xi = \sigma$) leicht möglich war, ohne dass Items in bestimmten Schwierigkeitsbereichen übermäßig häufig vorgegeben wurden.

5. Fragestellungen

In der Simulationsstudie von Makransky (2009) waren sowohl Itemparameter als auch Personenparameter standardnormalverteilt und alle Items modellkonform (DIF wurde nicht simuliert). Makransky (2012) empfiehlt auf eine Modellprüfung zu verzichten und ausschließlich gruppenspezifische Itemparameter zu verwenden. Allerdings gehen dadurch die Vorteile des Rasch-Modells wie Stichprobenunabhängigkeit oder Testfairness verloren (Kapitel II 2).

Im Rahmen der vorliegenden Arbeit soll ein Algorithmus entwickelt und geprüft werden, der in der Lage ist, ein Verfahren mit adaptiver Itemvorgabe zu kalibrieren. Die Personenparameter sind dabei standardnormalverteilt und die Itemparameter gleichverteilt. Die adaptive Kalibrierung sollte effektiver und effizienter sein, als dies durch eine zufällige Itemvorgabe möglich wäre. Außerdem soll die Gültigkeit des Rasch-Modells noch während der Testkalibrierung geprüft und Items, die dem Rasch-Modell widersprechen, von der weiteren Vorgabe ausgeschlossen werden. Die konkreten Fragestellungen sind dementsprechend:

5.1. Fragestellung 1

Werden die Itemparameter bei einer adaptiven Testkalibrierung effektiver geschätzt als bei einer Testkalibrierung mittels Zuvallsvorgabe?

- Hypothese 1a: Die mittleren Residuen der Itemparameter sind bei der adaptiven Kalibrierung am Ende der Testkalibrierung geringer als bei der zufälligen Kalibrierung (Genauigkeit).
- Hypothese 1b: Die Standardabweichung der Residuen der Itemparameter ist bei der adaptiven Kalibrierung am Ende der Testkalibrierung geringer als bei der zufälligen Kalibrierung (Präzision).

5.2. Fragestellung 2

Werden die Itemparameter bei einer adaptiven Testkalibrierung effizienter geschätzt als bei einer Testkalibrierung mittels Zuvallsvorgabe?

- Hypothese 2: Es sind mittels der adaptiven Itemkalibrierung weniger Personen nötig, um die Items gleich genau zu schätzen wie mittels zufälliger Itemvorgabe.

5.3. Fragestellung 3

Werden die Personenparameter durch die Items der adaptiven Kalibrierung genauer geschätzt als durch die Items der Zufallskalibrierung?

- Hypothese 3a: Die mittleren Residuen der Personenparameter sind bei der Vorgabe der adaptiv kalibrierten Items am Ende der Testkalibrierung geringer als bei der Vorgabe der zufällig kalibrierten Items (Genauigkeit).
- Hypothese 3b: Die Standardabweichung der Residuen der Personenparameter ist bei der Vorgabe der adaptiv kalibrierten Items am Ende der Testkalibrierung geringer als bei der Vorgabe der zufällig kalibrierten Items (Präzision).

Entspricht die Rangreihung der Personen nach Personenparameter bei der adaptiven Kalibrierung der tatsächlichen Fähigkeit der Personen?

- Hypothese 3c: Die Rangkorrelation der Personenparameter ist bei der Vorgabe der adaptiv kalibrierten Items nach abgeschlossener Testkalibrierung höher als bei der Vorgabe der zufällig kalibrierten Items.

5.4. Fragestellung 4

Werden durch den Fischer und Scheiblechner *z*-Test Items mit DIF erkannt?

- Hypothese 4: Es werden überzufällig häufig Items mit DIF erkannt.

III. Empirischer Teil

Dieses Kapitel beginnt mit dem Methodenteil, in welchem das Verfahren beschrieben wird. Anhand der Fragestellungen wird erläutert wie diese beantwortet wurden. Es folgt die Darstellung der Ergebnisse. Anschließend werden die Ergebnisse im Kontext zu den Fragestellungen diskutiert. Abgeschlossen wird dieses Kapitel durch eine Zusammenfassung dieser Arbeit.

1. Methode

Ziel der adaptiven Itemkalibrierung ist es die Genauigkeit und Präzision zu erhöhen. Dazu ist es notwendig, dass die wahren Werte der Versuchspersonen und Items bekannt sind, dies ist unter realen Bedingungen nicht möglich. Deshalb ist für die vorliegende Arbeit eine Simulationsstudie das Mittel der Wahl. Dadurch lassen sich Daten mit genau festgelegten Kriterien simulieren und man ist in der Lage, die Eigenschaften statistischer Tests unter verschiedenen Bedingungen zu überprüfen (Burton, Altman, Royston, & Holder, 2006). Eine Simulationsstudie benutzt bestimmte Annahmen über den Zufall (Zufallszahlen) der interessierenden Parameter.

1.1. Verfahrensbeschreibung

Ein Test mit k = 60 Items, wovon j = 15 Items pro Person vorgegeben werden, soll anhand von n = 500 Personen kalibriert werden. Da es sich bei der vorliegenden Arbeit um eine Simulationsstudie handelt, wurden keine echten Personen getestet. Wenn in weiterer Folge von Personen die Rede ist, sind stets nur simulierte Personen gemeint. Diese 500 Personen sind auf zwei gleich große Gruppen zu je 250 Personen aufgeteilt (eine Gruppe wurde als *männlich* und eine als *weiblich* bezeichnet). Es fanden r = 1000 Replikationen statt.

Die wahren Itemparameter sind gleichverteilt von -3 bis +3. Da die Kalibrierung eines Verfahrens bei zu vielen Items mit DIF problematisch ist (im Extremfall lässt sich nicht mehr feststellen, was die eigentlich zu messende Dimension ist), wurde für 4 Items DIF simuliert. Deshalb weisen die Items mit der Nummer 5, 20, 35 und 50 der Gruppe weiblich andere Werte als in der Gruppe männlich auf (Tabelle 2 im Anhang).

Die Verteilung der Personenparameter entspricht anders als jene der Itemparameter der Standardnormalverteilung. Abbildung 6 zeigt die Häufigkeitsverteilungen der Personenparameter

und Itemparameter. 50% der Itemparameter sind kleiner als -1.5 und größer als 1.5 (Abbildung 6, senkrechte Achsen), wohingegen sich nur 13.4% der Personenparameter in diesen Bereichen befinden. Somit sind im Verhältnis zu den Personen relativ wenig Items im Durchschnittsbereich vorhanden, aber viele in den Extrembereichen.

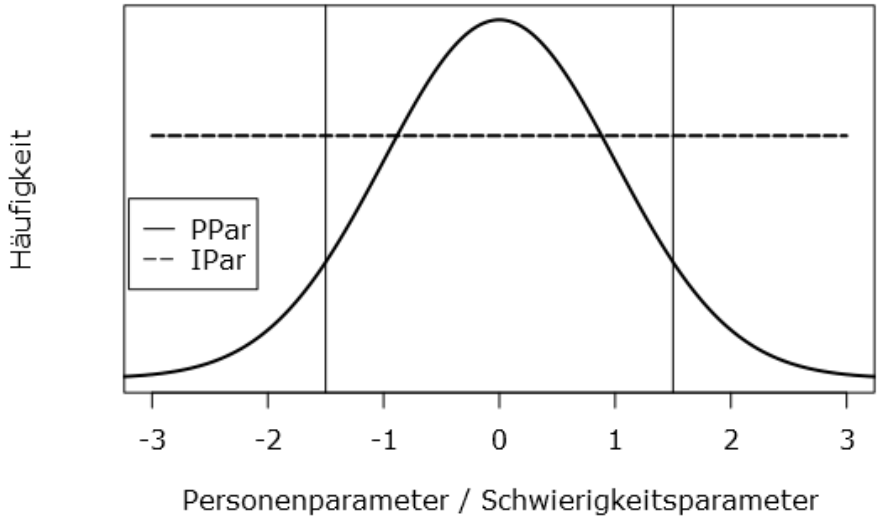


Abbildung 6: Häufigkeitsverteilung der Itemparameter (IPar) und Personenparameter (PPar).

1.2. Testkalibrierung

Die Kalibrierung des Verfahrens erfolgte sowohl mittels adaptiver als auch mittels zufälliger Itemvorgabe. Der Standardfehler der Items wird bei der adaptiven Itemvorgabe so lange berücksichtigt, bis für alle Items die Parameter schätzbar sind. Dieses Vorgehen erwies sich während der Entwicklung des Programms als vorteilhaft, da so die Items relativ gleichmäßig kalibriert werden konnten. Die Prüfung auf DIF mittels Fischer und Scheiblechner *z*-Test begann ab der 100. Versuchsperson. Dadurch konnte die Gefahr reduziert werden, dass schon in einem sehr frühen Stadium der Testkalibrierung Items fälschlicherweise ausgeschlossen wurden. Eine detaillierte Beschreibung der Programmentwicklung befindet sich im Anhang (5.2).

Nach 100 und 250 Personen sowie nach abgeschlossener Kalibrierung (500 Personen) wurde die aktuelle Genauigkeit des Verfahrens überprüft. Hierfür wurde der Test 5000 Personen (2500 weiblich und 2500 männlich) zu den Testkalibrierungsstadien nach 100, 250 und 500 Personen vorgegeben.

Für die Simulation wurde die freie und quelloffene Software *R* Version 3.2 verwendet. Die Programmierung fand unter der Entwicklungsumgebung *Rstudio* Version .95.953 statt. Eine ausführlich Darstellung des Algorithmus befindet sich im Anhang Abschnitt 5.3.

1.3. Datenauswertung

Die Datenauswertung orientierte sich anhand der Fragestellungen, da abhängig von diesen unterschiedliche Kennwerte berechnet werden mussten.

1.3.1. Effektivität der Itemparameterschätzung

Die Effektivität stellt ein Maß der Zielerreichung, unabhängig der eingesetzten Mittel, dar. In dem Kontext dieser Arbeit bedeutet effektiv, dass ein etwaig vorhandener Bias möglichst klein ist (*Genauigkeit*) und, dass die geschätzten Itemparameter möglichst gering um die wahren Werte streuen (*Präzision*). Zur Berechnung der Genauigkeit wurde die Differenz der geschätzten von den wahren Itemparameter berechnet (Residuen) und das Mittel über jedes einzelne Item sowie über den Test insgesamt (das Mittel des Mittels der Residuen der Items) gebildet. Für die Präzision wurde die mittlere Standardabweichung der Residuen je Item und über den Test als Ganzes (der Mittelwert der Standardabweichung der Residuen aller Items) berechnet. Verglichen wurde die adaptive Kalibrierung mit der Zufallskalibrierung. Für die Präzsion auf Itemparameterebene wurde außerdem ein *F*-Test berechnet um zu überprüfen, ob sich die Werte der adaptiven Kalibrierung von jenen der zufälligen signifikant unterscheiden. Als Irrtumswahrscheinlichkeit wurde $\alpha = .01$ gewählt.

1.3.2. Effizienz der Itemparameterschätzung

Die Effizienz ist ein Maß der Zielerreichung in Abhängigkeit von den eingesetzten Mitteln. Ein Verfahren ist dann effizient, wenn bei einem möglichst geringen Aufwand das Ausmaß der Zielerreichung maximal groß ist. Zur Beschreibung der Effizienz wurde die Präzision herangezogen, welche zu unterschiedlichen Testkalibrierungsstadien (nach 100, 250 und 500 Versuchspersonen) gemessen wurde. Die Auswertung der Präzision erfolgte für diese Fragestellung über den Test als Ganzes (der Mittelwert der Standardabweichung der Residuen aller Items). Verglichen wurde die adaptive Kalibrierung mit der Zufallskalibrierung.

1.3.3. Personenparameter nach abgeschlossener Kalibrierung

Die mittlere Genauigkeit und Präzision der Personenparameter wurde über alle 5000 Personen in den beiden Kalibrierungsstrategien berechnet. Um zu überprüfen, wie genau die Rangordnung der geschätzten Personenparameter die wahre Fähigkeit repräsentiert, wurde außerdem eine Rangkorrelation nach Spearman für jede Replikation berechnet. Die 1000 Rangkorrelationskoeffizienten wurden Fisher *z* transformiert. Aus den Fisher *z* transformierten Werten wurde ein Mittelwert gebildet. Dieser wurde in weiterer Folge wieder zurück in den Korrelationskoeffizienten *r* transformiert.Verglichen wurde die adaptive Kalibrierung mit der Zufallskalibrierung.

1.3.4. Fischer u. Scheiblechner z-Test

Zur Frage, ob der Fischer und Scheiblechner *z*-Test bei Items mit DIF häufiger positiv ausfällt als bei Items ohne DIF, wurde der Vierfelderkorrelationskoeffizient über jede Replikation berechnet. Es folgte eine Fisher *z* Transformation der 1000 Korrelationskoeffizienten, um aus diesen Werten in weiterer Folge einen Mittelwert zu bilden. Der Mittelwert wurde in weiterer Folge wieder zurück in den Korrelationskoeffizienten *r* transformiert.

2. Ergebnisse

Es folgt die Darstellung der Ergebnisse anhand der Fragestellungen.

2.1. Effektivität der Itemparameterschätzung

Abbildung 7 zeigt die mittlere Genauigkeit auf Itemebene nach abgeschlossener Kalibrierung (500 Personen). Dargestellt werden nur Items ohne DIF. Es zeigt sich, dass die Residuen der Items im mittleren Bereich in der zufälligen Kalibrierung geringer sind als bei der adaptiven Kalibrierung. Im Randbereich, bei den sehr schweren / sehr leichten Items sind die Residuen bei den adaptiv kalibrierten Verfahren geringer. Beiden Kalibrierungsstrategien ist gemein, dass schwere Items noch schwerer geschätzt und leichte Items noch leichter geschätzt werden, als sie tatsächlich sind.

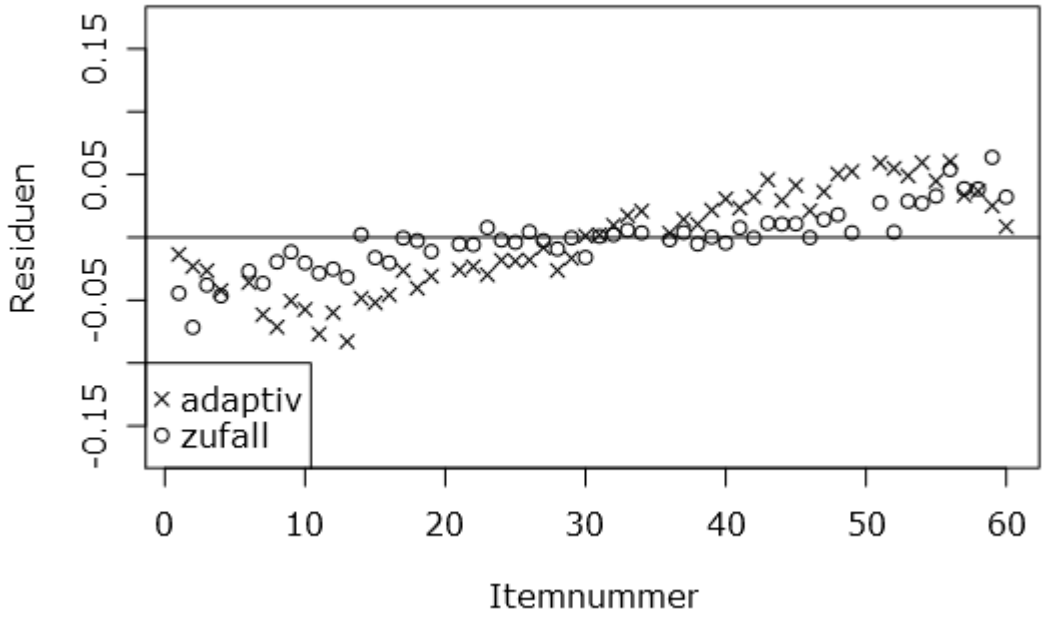


Abbildung 7: Mittlere Genauigkeit auf Itemebene

Abbildung 8 zeigt die mittlere Präzision auf Itemebene. Dargestellt werden nur Items ohne DIF. Bei 37 Items weisen die adaptiv kalibrierten Items eine signifikant höhere Präzision (geringere Werte der Standardabweichung der Residuen) auf als die zufällig kalibrierten Items. Die Werte sind im Detail in Tabelle 3 im Anhang dargestellt.

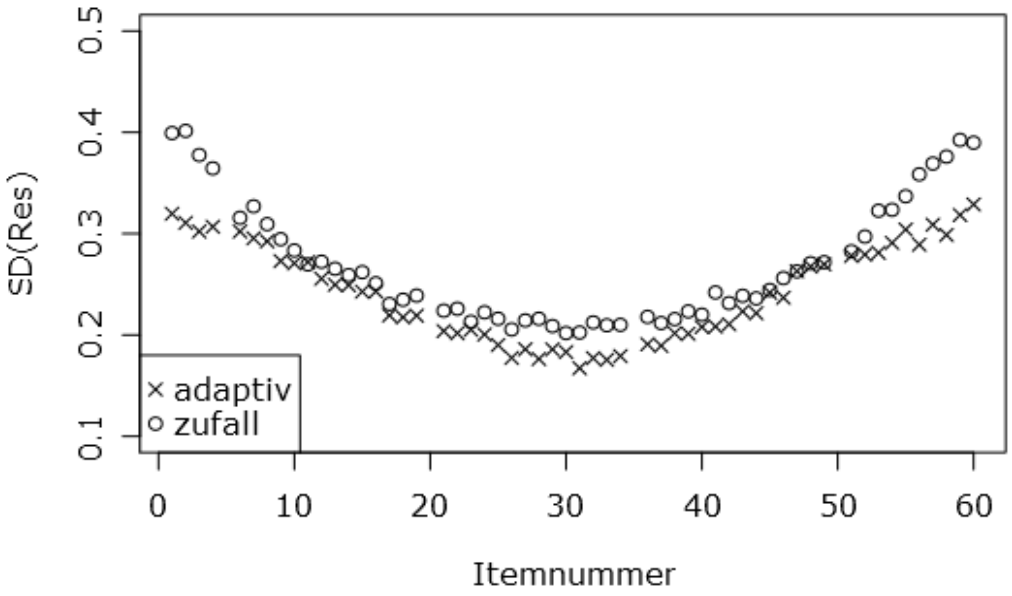


Abbildung 8: Mittlere Präzision auf Itemebene

Die mittlere Genauigkeit des Gesamtverfahrens (adaptive Kalibrierung) beträgt -.002. Die zufällige Kalibrierung wies eine mittlere Genauigkeit von -.001 auf. Die mittlere Präzision nach abgeschlossener Kalibrierung war in der adaptiven Kalibrierung mit .242 höher als in der zufälligen Kalibrierung mit .270 (Tabelle 3 im Anhang).

2.2. Effizienz der Itemparameterschätzung

Abbildung 9 zeigt die mittlere Präzision auf Verfahrensebene zu unterschiedlichen Kalibrierungsstadien. Nicht zu allen Testkalibrierungsstadien konnten stets alle Itemparameter berechnet werden. Dies ist der Tatsache geschuldet, dass sich die Itemparameter nicht berechnen lassen, wenn ein Item immer gelöst bzw. nicht gelöst wurde (weitere Informationen zu dieser Tatsache befinden sich im Anhang in Kapitel 5.2.1). Die Replikationen in denen nicht alle Itemparameter berechnet werden konnten, wurden in der Darstellung der Ergebnisse nicht berücksichtigt.

Die Präzision ist nach 100 Personen in der adaptiven Kalibrierung mit .549 (693 berücksichtigte Replikationen) geringer als bei der zufälligen Kalibrierung mit .521 (154 berücksichtigte Replikationen). Am Ende der Testkalibrierung zeigt sich ein umgekehrtes Bild: nach 500 Personen ist die Präzision in der adaptiven Kalibrierung mit .242 (1000 berücksichtigte Replikationen) höher als in der zufälligen Kalibrierung mit .270 (1000 berücksichtigte Replikationen). Nach 250 Personen war die Präzision in beiden Kalibrierungsstrategien ähnlich mit .352 (1000 berücksichtigte Replikationen) in der adaptiven Vorgabe und .351 (1000 berücksichtigte Replikationen) in der zufälligen Vorgabe. Bei interpolierten Werten ist die adaptive Strategie mit 436 Personen so genau wie du zufällige mit 500 Personen. Damit ist die Effizienz der adaptiven Kalibrierungsstrategie höher als jene der zufälligen. Die Werte sind im Detail in Tabelle 4 im Anhang dargestellt.

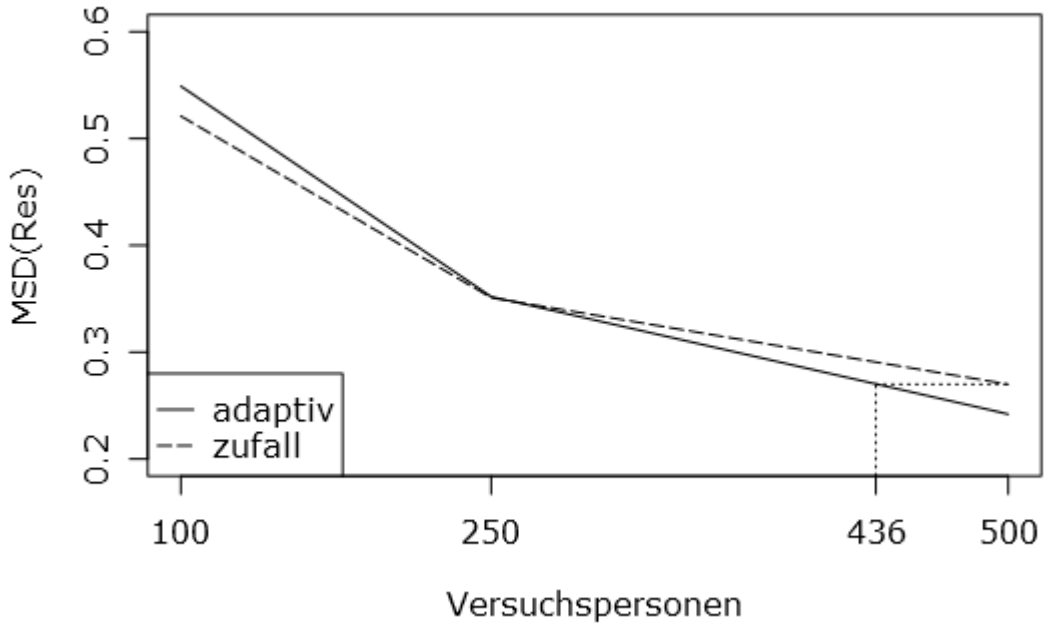


Abbildung 9: Mittlere Präzision auf Verfahrensebene zu unterschiedlichen Kalibrierungsstadien

2.3. Personenparameter nach abgeschlossener Kalibrierung

Tabelle 1 zeigt den mittleren Spearman Rangkorrelationskoeffizient, die Präzision und die Genauigkeit nach abgeschlossener Kalibrierung für die Personenparameter. Beide Kalibrierungsstrategien weisen eine mittlere Korrelation von .854 auf. Die Genauigkeit beträgt bei der adaptiven Kalibrierung -.002 und bei der zufälligen Kalibrierung -.001. Bei der adaptiven Kalibrierung wurde eine Präzision von .590 festgestellt, bei der zufälligen Kalibrierung von .583. Zwischen der adaptiven und zufälligen Itemkalibrierung besteht kein Unterschied in der Rangreihung der Personen nach ihrer Fähigkeit.

Tabelle 1

Mittlere Rankorrelation, Präzision und Genauigkeit der Personenparameter nach abgeschlossener Kalibrierung

Kalibrierung	*r*	MSD(Res)	M(Res)
adaptiv	.854	.590	-.002
zufällig	.854	.583	-.001

Hinweis: *r* = Spearmans Rangkorrelationskoeffizient; MSD(Res) = Mittlere Standardabweichung der Residuen (Präzision); M(Res) = Mittelwert der Residuen (Genauigkeit)

2.4. Ergebnisse Fischer u. Scheiblechner *z*-Test

Der mittlere Vierfelderkorrelationskoeffizient nach abgeschlossener Itemkalibrierung beträgt .919 ($p < .001$). Insgesamt wurden über die 1000 Replikationen 95.0% der Items mit DIF (3799 von 4000) durch den Fischer und Scheiblechner *z*-Test erkannt. Der Fischer und Scheiblechner *z*-Test gab bei 2.9% der modellkonformen Items (1647 von 56000) ein falsch positives Ergebnis aus. Der Fischer u. Scheiblechner *z*-Test erkennt überzufällig häufig Items mit DIF.

3. Diskussion

Ziel der vorliegenden Arbeit war es, einen Algorithmus zu entwickeln und zu prüfen, der in der Lage ist, einen Test mit gleichverteilten Itemparametern und standardnormalverteilten Personenparametern mittels adaptiver Itemvorgabe zu kalibrieren. Dieser Algorithmus wurde im Rahmen einer Simulationsstudie mit 1000 Replikationen untersucht. Geprüft wurde, ob sich die Genauigkeit und Präzision der Item- und Personenparameter durch die Anwendung des Algorithmus im Vergleich zur zufälligen Itemvorgabe verbessert. Außerdem wurde untersucht, ob durch die Modellprüfung mittels Fischer und Scheiblechner *z*-Test Items mit Differential Item Functioning (DIF) überzufällig häufig erkannt werden.

Bezüglich der Effektivität konnte gezeigt werden, dass die Itemparameterschätzungen nach abgeschlossener Testkalibrierung (500 Personen), sowohl bei der adaptiven als auch bei der zufälligen Itemvorgabe, einen Bias aufweisen. Dieser ist bei der adaptiven Vorgabe tendenziell höher. Der Bias tritt nur auf Itemebene und kaum auf Verfahrensebene auf. Dies ist der Tatsache geschuldet, dass schwere Items schwerer und leichte Items leichter geschätzt werden als sie tatsächlich sind. Dadurch tritt der Bias im Mittel über das Gesamtverfahren kaum auf. Hinsichtlich der Präzision nach abgeschlossener Kalibrierung konnte gezeigt werden, dass diese bei 37 der 60 Items bei der adaptiven Kalibrierungsstrategie höher ist als bei der zufälligen.

Bezüglich der Effizienz ergibt sich kein eindeutiges Bild. Nach 500 Personen ist die Präzision der Itemparameterschätzung, wie bereits erwähnt, in der adaptiven Kalibrierung höher als in der zufälligen. Nach 250 Personen sind die beiden Kalibrierungsstrategien sehr ähnlich und bei 100 Personen zeigt sich ein umgekehrtes Bild. Zu diesem Zeitpunkt weist die zufällige Kalibrierung eine höhere Präzision auf. Es ist allerdings zu beachten, dass nur 154 von 1000 Replikationen zum Kalibrierungszeitpunkt 100 Personen bei der zufälligen Itemvorgabe berücksichtigt wurden, da nicht alle Itemparameter berechnet werden konnten (bei der adaptiven Vorgabe waren es hingegen 693). Der Grund für das bessere Abschneiden der zufälligen Kalibrierungsstrategie zu diesem Zeitpunkt könnte dadurch zustande kommen, dass tendenziell eher jene Replikationen der Zufallskalibrierung berücksichtigt wurden, die eine höhere Präzision aufweisen. Weitere Untersuchungen sind hierfür notwendig.

Bei der Vorgabe des fertig kalibrierten Verfahrens an 5000 Personen konnte gezeigt werden, dass die mittlere Präzision der Personenparameter der adaptiv kalibrierten Verfahren nicht höher ist als jene der mittels Zufallsvorgabe kalibrierten Verfahren. Unabhängig der Kalibrierungsstrategie ist die gemessene Präzision der Personenparameter mit .590 und .583 sehr niedrig. Diese Präzision

würde für die Praxis zu hohe Konfidenzintervalle ergeben. Um die Präzision zu erhöhen sollten mehr als 15 Items je Person vorgegeben werden.

Der Fischer und Scheiblechner *z*-Test konnte 95.0% der Items mit DIF erkennen. Bei 2.9% der Items fiel er falsch positiv aus. Dies spricht dafür, dass dieses Verfahren im Kontext der adaptiven Itemkalibrierung für die Modellprüfung geeignet ist.

Die Stärke der vorliegenden Arbeit liegt darin, dass erstmals ein Algorithmus entwickelt und empirisch geprüft wurde, der in der Lage ist ein Verfahren zu kalibrieren, in welchem die Modellprüfung auf Gültigkeit des Rasch-Modells während der Testkalibrierung durchgeführt wird. Dadurch bleiben die Vorteile des Rasch-Modells (siehe Kapitel II 2.) für die fertig kalibrierten Verfahren und somit den Einsatz in der Praxis erhalten. Items mit DIF werden noch während der Kalibrierung erkannt und von der weiteren Vorgabe ausgeschlossen. Dies macht Ressourcen für die Vorgabe von modellkonformen Items frei. Außerdem wurde erstmals geprüft, wie ein Verfahren adaptiv zu kalibrieren ist, wenn die Itemparameter gleichverteilt und die Personenparameter standardnormalverteilt sind.

Eine der Limitierungen der vorliegenden Arbeit betrifft die Methode. Simulationsstudien können die Realität nur bis zu einem bestimmten Grad wiedergeben. Die Daten werden unter bestimmten Voraussetzungen und Annahmen simuliert. So ist zum Beispiel eine exakte Gleichverteilung der Items unwahrscheinlich. Viel eher dürften sich in der Realität die Mehrzahl der Items im Durchschnittsbereich befinden oder mehrere zu leicht oder zu schwer sein. Außerdem wurde der Algorithmus während der Testentwicklung auf andere Kennwerte optimiert (Korrelation der Itemparameter mit den wahren Werten), die sich letztendlich für die Fragestellungen als nicht relevant erwiesen haben. Der Algorithmus ist zusätzlich ausschließlich für ein Verfahren mit 60 Items, wovon 15 je Person vorgegeben werden, entwickelt worden. Für andere Verfahren müsste der Algorithmus angepasst werden. Wenngleich diese Anpassungen auch leicht durchführbar wären, so ist doch zu prüfen, wie sich die neuen Modelleinstellungen auf die Ergebnisse auswirken. Das macht die Anwendung in der Praxis aufwendig. In der aktuellen Version werden die Itemparameter für beide simulierten Geschlechtergruppen getrennt voneinander vorgenommen. Information, welche in der Gruppe weiblich zur Verfügung stand, wurde nicht für die Vorgabe in der Gruppe männlich genutzt (und umgekehrt).

Zum aktuellen Stand der Forschung ist die adaptive Itemkalibrierung eingeschränkt für den Einsatz in der Praxis geeignet. Es konnten zwar die Effektivität und Effizienz gegenüber der Zufallsvorgabe erhöht werden, doch es sind weitere Optimierungen notwendig: Um die Präzision

der adaptiven Kalibrierung gegenüber der zufälligen Kalibrierung weiter zu verbessern, sind andere Verfahren zur Itemauswahl zu überprüfen, die die bereits vorhandene Information (auch in anderen Gruppen) besser berücksichtigen. Der Algorithmus sollte außerdem auf die Kennwerte Präzision und Genauigkeit weiter optimiert werden.

Langfristig ist zu erwarten, dass die adaptive Itemkalibrierung eine Möglichkeit der Testkalibrierung bei Verfahren darstellt, deren Zielgruppe nur unter hohem Aufwand für die Testkalibrierung zu gewinnen ist. Hier ist es von besonderer Bedeutung, dass eine maximale Menge an Information je Versuchsperson gewonnen wird.

4. Literaturverzeichnis

Andersen, E. B. (1973). A goodness of fit test for the Rasch model. *Psychometrika*, *38*(1), 123–140.

Baker, F. (2001). The basic of item response theory. Retrieved August 17, 2014, from http://echo.edres.org:8080/irt/baker/chapter1.pdf

Baker, F. B., & Seock-Ho, K. (2004). *Item response theory: parameter estimation techniques* (2nd ed.). Boca Raton: Taylor & Francis.

Becker, J. (2000). *Computergestütztes adaptives testen (CAT) von angst entwickelt auf der grundlage der item response theorie (IRT)*. Freie Universität Berlin.

Bortolotti, S. L. V., Tezza, R., Andrade, D. F., Bornia, A. C., & Sousa Júnior, A. F. (2013). Relevance and advantages of using the item response theory. *Quality & Quantity*, *47*(4), 2341–2360. doi:10.1007/s11135-012-9684-5

Fischer, G. H. (1974). *Einführung in die theorie psychologischer tests*. Bern - Stuttgart - Wien: Verlag Hans Huber.

Fischer, G. H. (1983). Neuere testtheorie. In C. F. Graumann, T. Herrmann, H. Hörmann, M. Irle, H. Thomae, & F. E. Weinert (Eds.), *Enzyklopädie der Psychologie Bd. 3 Messen und Testen* (1st ed., pp. 604–692). Göttingen, Toronto, Zürich: Hogrefe.

Hambleton, R. K., Swaminathan, H., & Rogers, J. H. (1991). *Fundamentals of item response theory.* Newbury Park: Sage Publications, Inc.

Jansen, P. G. W., Wollenberg, D., & Arnold, L. (1988). Unconditional parameter estimates in the rasch model for inconsistency. *Applied Psychological Measurment*, *12*(3), 297–306.

Kubinger, K. D. (1989). Aktueller stand und kritische würdigung der probabilistischen testtheorie. In K. D. Kubinger (Ed.), *Moderne Testtheorie : Ein abriss samt neuesten beiträgen* (2nd ed., pp. 19–83). Weinheim und Basel: Beltz Verlag.

Kubinger, K. D. (2009). *Psychologische diagnostik* (2nd ed.). Göttingen: Hogrefe Verlag GmbH & Co. KG.

Kubinger, K. D. (2014a). Adaptives testen. (M. A. Wirtz, Ed.), *Dorsch – lexikon der psychologie*. Retrieved from https://portal.hogrefe.com/dorsch/adaptives-testen/

Kubinger, K. D. (2014b). Item-response-theorie (IRT). (M. A. Wirtz, Ed.), *Dorsch – lexikon der psychologie.* Retrieved from https://portal.hogrefe.com/dorsch/item-response-theorie-irt/

Kubinger, K. D. (2014c). Rasch-modell. (M. A. Wirtz, Ed.), *Dorsch – lexikon der psychologie.* Retrieved from https://portal.hogrefe.com/dorsch/rasch-modell/

Kubinger, K. D., & Draxler, C. (2007). Probleme bei der testkonstruktion nach dem Rasch-modell. *Diagnostica*, *53*(3), 131–143. doi:10.1026/0012-1924.53.3.131

Kubinger, K. D., Rasch, D., & Yanagida, T. (2011). *Statistik in der psychologie* (1st ed.). Göttingen: Hogrefe Verlag GmbH & Co. KG.

Kubinger, K. D., Steinfeld, J., Reif, M., & Yanagida, T. (2012). Biased (conditional) parameter estimation of a Rasch model calibrated item pool administered according to a branched testing design. *Psychological test and assessment modeling*, *54*(4), 450–460.

Langer, M. M. (2008). *A reexamination of Lord's Wald Test for differential item functioning using item response theory and modern error estimation*. University of North Carolina.

Magis, D., & Gilles, R. (2012). Random generation of response patterns under computerized adaptive testing with the R package catR. *Journal of statistical software*, *48*(8), 1–31. Retrieved from http://www.jstatsoft.org/v48/i08/

Mair, P., Hatzinger, R., & Maier, M. J. (2014). eRm: extended Rasch modeling. R package version 0.15-4. Retrieved from http://erm.r-forge.r-project.org/

Makransky, G. (2009). An automatic online calibration design in adaptive testing. In D. J. Weiss (Ed.), *Proceedings of the 2009 GMAC conference on computerized adaptive testing*. Retrieved from www.psych.umn.edu/psylabs/CATcentral

Makransky, G. (2012). *Computerized adaptive testing in industrial and organizational psychology.* University of Twente.

Neyman, J., & Scott, E. L. (1948). Consistent estimates based on partially consistent observations. *Econometrica*, *16*(1), 1–32.

Rasch, G. (1960). *Probabilistic models for some intelligence and attainment tests*. Copenhagen: Danish Institute for Educational Research.

Robitzsch, A. (2014). sirt: supplementary item response theory models. *R package version 0.44-48*. Retrieved January 3, 2015, from http://cran.r-project.org/package=sirt

Spiel, C., Schober, B., & Litzenberger, M. (2008). *Projektbericht evaluation der eignungstests für das medzinstudium in österreich*.

Warm, T. (1989). Weighted likelihood estimation of ability in item response theory. *Psychometrika*, *54*(3), 427–450. Retrieved from http://link.springer.com/article/10.1007/BF02294627

Wikipedia. (2014). Eignungstest für das medizinstudium in Österreich. *Wikipedia*. Retrieved January 6, 2015, from http://de.wikipedia.org/wiki/Eignungstest_für_das_Medizinstudium_in_Österreich#Genderunterschiede_der_Ergebnisse_in_.C3.96sterreich

Willse, J. T. (2014). mixRasch: mixture rasch models with JMLE. R package version 0.1. Retrieved from http://cran.r-project.org/package=mixRasch

Wirtz, M. A. (2014). Differential item functioning (DIF). (24.8.2014, Ed.)*Lexikon der Psychologie*. Retrieved from https://portal.hogrefe.com/dorsch/differential-item-functioning-dif/

5. Anhang

5.1. Tabellen

Tabelle 2: Die Itemparameter der beiden Gruppen m (ItemparM) und w (ItemparW)

Itemnummer	ItemparM	ItemparW	Itemnummer	ItemparM	ItemparW
1	-3,00	-3,00	31	0,05	0,05
2	-2,90	-2,90	32	0,15	0,15
3	-2,80	-2,80	33	0,25	0,25
4	-2,69	-2,69	34	0,36	0,36
5	-2,59	-0,59	35	0,46	2,46
6	-2,49	-2,49	36	0,56	0,56
7	-2,39	-2,39	37	0,66	0,66
8	-2,29	-2,29	38	0,76	0,76
9	-2,19	-2,19	39	0,86	0,86
10	-2,08	-2,08	40	0,97	0,97
11	-1,98	-1,98	41	1,07	1,07
12	-1,88	-1,88	42	1,17	1,17
13	-1,78	-1,78	43	1,27	1,27
14	-1,68	-1,68	44	1,37	1,37
15	-1,58	-1,58	45	1,47	1,47
16	-1,47	-1,47	46	1,58	1,58
17	-1,37	-1,37	47	1,68	1,68
18	-1,27	-1,27	48	1,78	1,78
19	-1,17	-1,17	49	1,88	1,88
20	-1,07	-3,07	50	1,98	-0,02
21	-0,97	-0,97	51	2,08	2,08
22	-0,86	-0,86	52	2,19	2,19
23	-0,76	-0,76	53	2,29	2,29
24	-0,66	-0,66	54	2,39	2,39
25	-0,56	-0,56	55	2,49	2,49
26	-0,46	-0,46	56	2,59	2,59
27	-0,36	-0,36	57	2,69	2,69
28	-0,25	-0,25	58	2,80	2,80
29	-0,15	-0,15	59	2,90	2,90
30	-0,05	-0,05	60	3,00	3,00

Hinweis: ItemparM = Itemparameter der Gruppe männlich, ItemparW = Itemparameter der Gruppe weiblich

Tabelle 3

Mittlere Residuen und Standardabweichung der Residuen auf Item- und Verfahrensebene

	adaptiv		zufall			
Item	M(Res)	SD(Res)	M(Res)	SD(Res)	F-Ratio	p
1	-.014	.320	-.044	.399	1.561	< .001
2	-.023	.311	-.072	.401	1.670	< .001
3	-.027	.302	-.038	.378	1.562	< .001
4	-.042	.307	-.046	.364	1.412	< .001
6	-.036	.302	-.027	.316	1.092	.083
7	-.062	.295	-.037	.327	1.225	< .001
8	-.071	.292	-.020	.309	1.119	.038
9	-.051	.273	-.012	.295	1.165	.008
10	-.057	.271	-.020	.283	1.096	.074
11	-.077	.272	-.029	.270	1.017	.397
12	-.060	.256	-.025	.272	1.136	.022
13	-.083	.250	-.032	.265	1.131	.026
14	-.048	.249	.002	.259	1.082	.106
15	-.052	.243	-.016	.262	1.165	.008
16	-.046	.243	-.020	.251	1.068	.150
17	-.026	.219	.000	.231	1.105	.058
18	-.040	.217	-.003	.235	1.166	.008
19	-.031	.219	-.011	.239	1.190	.003
21	-.026	.204	-.005	.224	1.208	.001
22	-.023	.202	-.006	.226	1.255	< .001
23	-.030	.205	.008	.213	1.079	.114
24	-.018	.200	-.002	.222	1.231	< .001
25	-.019	.190	-.004	.216	1.292	< .001
26	-.018	.177	.004	.205	1.342	< .001
27	-.008	.186	-.003	.214	1.329	< .001
28	-.026	.176	-.009	.216	1.502	< .001
29	-.017	.186	.000	.209	1.264	< .001
30	.001	.183	-.016	.202	1.216	.001
31	.002	.167	.001	.202	1.466	< .001
32	.010	.177	.002	.212	1.439	< .001
33	.017	.175	.006	.210	1.428	< .001
34	.021	.179	.004	.210	1.375	< .001
36	.003	.191	-.002	.218	1.305	< .001
37	.014	.189	.004	.212	1.253	< .001
38	.010	.202	-.005	.216	1.144	.017
39	.022	.201	.000	.223	1.232	< .001
40	.031	.208	-.004	.220	1.121	.035
41	.023	.208	.008	.242	1.349	< .001

42	.032	.211	-.001	.231	1.208	.001
43	.046	.223	.011	.239	1.144	.017
44	.029	.221	.011	.236	1.136	.022
45	.041	.242	.011	.244	1.023	.362
46	.021	.237	.000	.256	1.170	.007
47	.036	.263	.014	.263	1.002	.489
48	.050	.267	.018	.271	1.026	.342
49	.052	.270	.004	.272	1.016	.402
51	.059	.278	.028	.283	1.033	.305
52	.055	.279	.004	.297	1.132	.025
53	.049	.281	.029	.323	1.318	< .001
54	.059	.291	.027	.324	1.238	< .001
55	.045	.304	.033	.337	1.229	< .001
56	.060	.289	.054	.359	1.538	< .001
57	.034	.309	.039	.369	1.430	< .001
58	.037	.299	.039	.376	1.585	< .001
59	.025	.318	.064	.393	1.520	< .001
60	.009	.329	.032	.390	1.406	< .001
M	-.002	.242	-.001	.270		

Hinweis: M(Res) = Mittelwert der Residuen; SD(Res) = Standardabweichung der Residuen

Tabelle 4

Mittlere Präzision zu unterschiedlichen Kalibrierungsstadien

	adaptiv		zufall	
Personen	MSD(Res)	Replikationen	MSD(Res)	Replikationen
100	.549	693	.521	154
250	.352	1000	.351	936
500	.242	1000	.270	1000

Hinweis: MSD(RES) = Mittlere Standardabweichung der Residuen über das Gesamtverfahren; Replikationen = Anzahl der berücksichtigten Replikationen

5.2. Entwicklung des Programms

Die Entwicklung des in dieser Arbeit verwendeten Algorithmus zur adaptiven Itemkalibrierung geschah in mehreren Schritten. In den ersten Programmversionen erfolgten beispielsweise die Parameterschätzungen noch mittels der CML (Kapitel II 2.1.5), doch es zeigte sich bereits sehr früh, dass dieses Verfahren aufgrund des hohen Rechenaufwands nicht für die vorliegende Simulationsstudie geeignet ist. So hätten die Berechnungen im vorgesehenen Umfang von 1000 Replikationen über ein Jahr in Anspruch genommen. Ein weiterer Nachteil der CML war der Bias der Parameterschätzungen.

Im Laufe der Programmentwicklung traten außerdem unerwartete Probleme auf, aber auch Möglichkeiten, diesen durch unterschiedliche Itemvorgabestrategien zu begegnen. Die verschiedenen Lösungsansätze (5.2.1) wurden in mehreren Prätests (5.2.5) mit einer geringen Anzahl an Replikationen (r = 25) miteinander verglichen, um abschätzen zu können, welche Vorgabeeinstellungen zu tendenziell besseren und welche zu eher schlechteren Ergebnissen führen. Die eigentliche Simulation mit 1000 Replikationen wurde den Ergebnissen der Prätests angepasst.

5.2.1. Aufgetretene Schwierigkeiten und Lösungsansätze

Während der Entwicklung des Programms zeigte sich, dass eine Itemvorgabe nur anhand der optimalen Passung zwischen Itemparameter und Personenparameter problematisch sein kann. Wenn die Itemparameter eine andere Häufigkeitsverteilung als die Personenparameter aufweisen, führt dies nämlich dazu, dass die Items im Durchschnittsbereich sehr viel häufiger vorgegeben werden als jene im Randbereich (Abbildung 10). Dies ist zwar durchaus im Sinne der adaptiven Itemkalibrierung und notwendig, um eine maximale Menge an Information zu erhalten, führt allerdings auch dazu, dass extrem schwere und extrem leichte Items nur sehr ungenau geschätzt werden. Außerdem trat immer wieder der Fall ein, dass für einzelne Items schon zu Beginn sehr extreme Parameter geschätzt wurden, die oftmals auch weit von den wahren Werten entfernt waren. Da es an Personen mit ebenso extremen Werten mangelte, wurde das Item, wenn überhaupt, nur noch vereinzelt vorgegeben. Deshalb konnten diese sehr ungenauen Schätzungen im Randbereich in weiterer Folge kaum noch korrigiert werden. Diese Fehleinschätzungen waren auch bei der Modellprüfung problematisch, da das Verfahren stets unabhängig voneinander in den beiden Gruppen männlich und weiblich kalibriert wurde. Somit kam es häufig vor, dass in einer Gruppe eine starke Fehleinschätzung auftrat und durch die Modellprüfung mittels Fischer und Scheiblechner z-Test für jenes Item DIF festgestellt wurde. Das Item wurde in weiterer Folge aufgrund eines falsch positiven Ergebnisses ausgeschlossen. Im ungünstigsten Fall musste in

weiterer Folge die Testkalibrierung abgebrochen werden, weil nicht mehr genügend Items für eine weitere Vorgabe im Sinne der adaptiven Itemkalibrierung zur Verfügung standen.

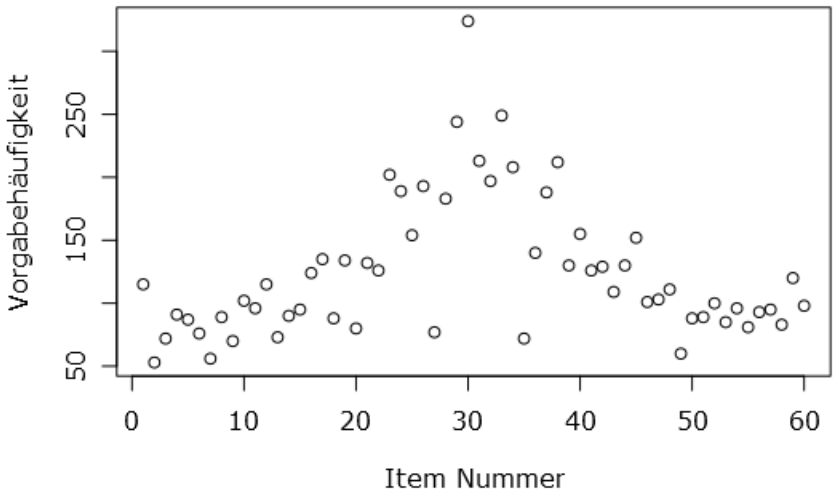


Abbildung 10: Die Vorgabehäufigkeit der Items eines Verfahrens, welches nach dem Kriterium der maximalen Information kalibriert wurde.

Eine weitere Herausforderung war der Umgang mit Items, die bisher nie / immer gelöst wurden. Über diese existiert keine Information im Sinne des Rasch-Modells, da die Parameterschätzung nicht möglich ist (Kapitel II 2.1). Gleiches gilt für Personen, die alle / keine Items gelöst haben. Aus diesem Grund schließen in aller Regel die gängigen Programmpakete zur Personen- und Itemparameterschätzung die betreffenden Personen und Items aus bzw. müssen vorher vom Benutzer manuell ausgeschlossen werden. Im Rahmen der Testkalibrierung ist es deshalb von besonderem Interesse, Testpersonen zu meiden, die alle vorgegebenen Items lösen / nicht lösen, da in diesem Fall Ressourcen für die Testung aufgewendet werden, allerdings kein Informationsgewinn stattfindet. Items ohne Information (in weiterer Folge „Problemitems“ genannt) sollten deshalb vermieden werden. Ein Sonderfall sind Problemitems, die bisher sowohl gelöst als auch nicht gelöst wurden (wie Item 4 im folgendem Beispiel). Diese werden erst durch den sukzessiven Ausschluss anderer Problemitems und Personen, welche alle Items gelöst / nicht gelöst haben, erkannt, wie folgendes Beispiel zeigt:

Tabelle 5:

Problemitems I

VP	Item 1	Item 2	Item 3	Item 4	Item 5
1	0	0	0	0	1
2	0	1	1	1	1
3	0	1	0	1	1
4	1	0	1	1	1
5	0	0	1	1	1

Hinweis: VP = Versuchsperson

Tabelle 5 zeigt, dass Item 5 bisher immer gelöst wurde. Das Item steht also bei der Berechnung des Rasch-Modells nicht zur Verfügung und muss ausgeschlossen werden. Es ergibt sich nun folgende Matrix:

Tabelle 6:

Problemitems II

VP	Item 1	Item 2	Item 3	Item 4
1	0	0	0	0
2	0	1	1	1
3	0	1	0	1
4	1	0	1	1
5	0	0	1	1

Hinweis: VP = Versuchsperson

Tabelle 6 zeigt, dass Testperson 1 ebenfalls keine Information im Sinne des Rasch-Modells liefert, weshalb diese von der weiteren Berechnung ausgeschlossen werden muss, bis mehr Information vorhanden ist. Die Matrix verändert sich wie folgt:

Tabelle 7:

Problemitems III

VP	Item 1	Item 2	Item 3	Item 4
2	0	1	1	1
3	0	1	0	1
4	1	0	1	1
5	0	0	1	1

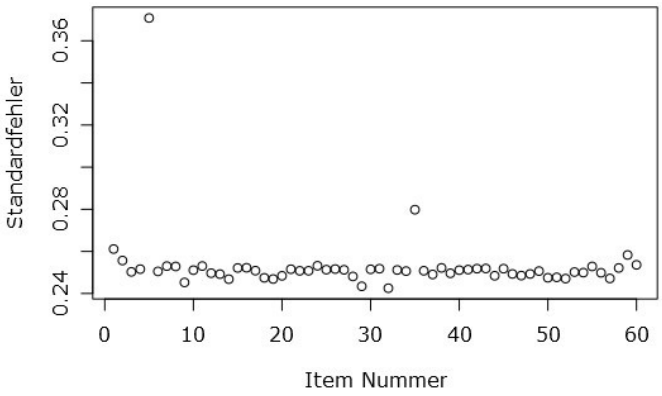
Hinweis: VP = Versuchsperson

Tabelle 7 zeigt, dass Item 4 ebenfalls keine Information mehr liefern kann und von der Berechnung des Rasch-Modells ausgeschlossen werden muss.

Die während der Entwicklung verwendeten R-Pakete zur Parameterschätzung, eRm (Mair, Hatzinger, & Maier, 2014), sirt (Robitzsch, 2014) und mixRasch (Willse, 2014) haben mit einem derartigen Antwortmuster, vor allem bei kleinen Stichprobengrößen, Schwierigkeiten und geben eine Fehlermeldung aus. Dies hängt damit zusammen, dass dieses Problem in erster Linie dann auftritt, wenn der Test bisher nur von wenigen Personen bearbeitet wurde. Bei, wie in der Praxis üblich, größeren Stichproben ($n > 100$) tritt dieses Problem nur noch vereinzelt auf. Vermutlich ist dies der Grund, warum den verwendeten Programmen ein Algorithmus zur Erkennung solcher Problemitems fehlt. Trotzdem ist diesem Problem relativ einfach zu begegnen, indem Schritt für Schritt die Datenmatrix nach informationslosen Items und Personen überprüft wird und diese in weiterer Folge ausgeschlossen werden. Dieser Vorgang muss so lange wiederholt werden bis keine Items und Personen mehr vorhanden sind, über die keine Information zur Verfügung steht.

5.2.2. Lösungsansatz I: Verwendung des Standardfehlers als Vorgabekriterium

Wie bereits in Kapitel II 3.2 beschrieben wurde, resultiert der Standardfehler eines Items aus der gesammelten Information zu diesem Item. Je weniger Information zu einem Item vorhanden ist, desto höher ist deshalb der Standardfehler. Gibt man immer jenes Item mit dem höchsten Standardfehler vor, führt dies dazu, dass der gesamte Itempool weitgehend gleich genau geschätzt wird (Abbildung 11 links). Abbildung 11 rechts zeigt, dass vor allem die sehr leichten und sehr schweren Items am häufigsten vorgegeben wurden. Die Items des Durchschnittsbereichs, also jene über welche man am meisten Information erhalten könnte, da sie die beste Passung zur Mehrheit der standardnormalverteilten Personenparameter hätten, werden am seltensten vorgegeben. Insgesamt würde diese Kalibrierungsstrategie eher ungenaue Parameterschätzungen ergeben und wäre einer zufälligen Itemvorgabe unterlegen. Somit ist der Standardfehler als alleiniges Vorgabekriterium zur Itemkalibrierung nicht geeignet.



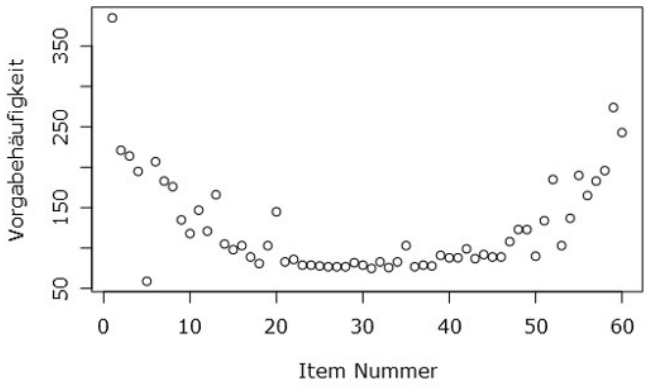


Abbildung 11 links: Bei der Kalibrierung dieses Verfahrens wurde immer jenes Item mit dem höchsten Standardfehler vorgegeben. Die Standardfehler der Items sind am Ende der Testkalibrierung weitgehend gleich. Lediglich Item 5 und 35 weichen ab (beide Items weisen DIF auf).

rechts: Bei Verwendung des Standardfehlers als alleiniges Vorgabekriterium zeigt sich, dass Items in den Extrembereichen besonders häufig vorgegeben werden.

5.2.3. Lösungsansatz II: Adjustierung der Itemparameterschätzer

Wie in Kapitel II 3.2 beschrieben, kann mit Hilfe des Standardfehlers ein Konfidenzintervall berechnet werden, welches mit der Irrtumswahrscheinlichkeit α den Wert des wahren Itemparameters umschließt. Je mehr Information zu einem Item vorhanden ist, desto kleiner ist die Schätzfehlervarianz und desto kleiner ist das Konfidenzintervall. Dieses Faktum lässt sich nützen, um sehr extreme (und meist auch falsche) Werte, die vor allem zu Beginn der Itemkalibrierung auftreten, in Abhängigkeit von der bisher vorhandenen Information zu adjustieren. So wird abhängig von der gewählten Irrtumswahrscheinlichkeit z_α (Formel 12) ein Konfidenzintervall für jedes Item berechnet. Jener Wert des Konfidenzintervalls, welcher sich näher bei 0 befindet, dient als Grundlage für die weitere Itemvorgabe. Sollte des Konfidenzintervall den Wert 0 umfassen, wird als Itemparameter 0 verwendet. Folgendes Beispiel soll das Vorgehen illustrieren:

Die Schwierigkeitsparameter σ_i beträgt .3. Die Schätzfehlervarianz des Items beträgt .7. Für die Irrtumswahrscheinlichkeit wurde willkürlicherweise $z_\alpha = .5$ gewählt. Setzt man die Werte in die Formel 12 ein, so erhält man ein Konfidenzintervall für den Schwierigkeitsparameter des Items *i* von - .12 bis .72. Da das Konfidenzintervall den Wert 0 umschließt, wird für die Itemauswahl als Schwierigkeitsparameter 0 verwendet.

Items, über welche bisher nur wenig Information vorhanden ist, werden stärker Richtung 0 adjustiert als jene, über die bereits relativ viel Information akkumuliert werden konnte. Dies führt dazu, dass das Ausmaß der Adjustierung mit steigender Menge an Information im Laufe der Testkalibrierung immer geringer wird.

5.2.4. Lösungsansatz III: Veränderung des Vorgabekriteriums

Die Anzahl der Problemitems, also Items ohne Information (5.2.1), lässt sich als Indikator für den Fortschritt der Testkalibrierung nutzen. Zu Beginn der Kalibrierung sind viele Problemitems vorhanden, die im Laufe der Kalibrierung immer weniger werden. Gegen Ende sollten idealerweise die Parameter des gesamten Itempools geschätzt werden können. So lange noch Problemitems vorhanden sind, sollte die Itemvorgabe stärker darauf ausgerichtet sein, über diese Items ebenfalls Information zu erhalten. Dies kann durch das Vorgabekriterium des höchsten Standardfehlers (Lösungsansatz I) erreicht werden. Hierfür muss für die Problemitems ein maximal hoher Standardfehler festgelegt werden. Dies führt dazu, dass Items ohne Information bevorzugt vorgegeben werden. Sobald keine Problemitems mehr vorhanden sind, wird das Vorgabekriterium verändert und zum Beispiel jenes der maximalen Information verwendet.

5.2.5. Prätests

Die verschiedenen Lösungsansätze wurden (zum Teil kombiniert) miteinander verglichen, um abzuschätzen, welche sich besser bewähren. Die Vergleiche erfolgten, wie bereits erwähnt, mit nur 25 Replikationen. Dementsprechend sind die Ergebnisse nur unter Vorbehalt zu interpretieren und geben bestenfalls eine Tendenz wider. Letztendlich sollen die Ergebnisse der Prätests lediglich eine Entscheidungsgrundlage für die Modelleinstellungen der eigentlichen Simulationsstudie liefern. Insgesamt ist, bis auf die für den jeweiligen Prätest relevanten Einstellungen, der Algorithmus derselbe wie in der eigentlichen Simulation, weshalb an dieser Stelle für Details auf Kapitel 5.3 verwiesen wird.

Erhoben wurde die Korrelation der geschätzten Item- und Personenparameter mit den wahren Werten und die Anzahl der als richtig und falsch als DIF erkannten Items. Von den Korrelationen wurde der Fisher *z*-transformierte Durchschnitt über die 25 Vorgaben berechnet. Außerdem wurde erhoben, wie oft die Prüfung auf DIF zu den verschiedenen Kalibrierungszeitpunkten möglich war (diese kann erst durchgeführt werden, wenn in beiden Gruppen keine Problemitems mehr vorhanden sind). Diese Zahl kann zwischen 0 (der Fischer und Scheiblechner *z*-Test konnte in

keiner der 25 Replikationen durchgeführt werden) und 25 schwanken (der Fischer und Scheiblechner *z*-Test konnte in allen Replikationen durchgeführt werden).

Eine Prüfung auf DIF erwies sich zum Zeitpunkt mit nur 100 Testpersonen zur Kalibrierung nur vereinzelt als möglich, da noch nicht für alle Items in beiden Gruppen die Parameter geschätzt werden konnten. Die Modellprüfung mittels Fischer und Scheiblechner *z*-Test war erst ab 250 Personen im Großteil der Replikationen möglich. Deshalb wurde auf die Darstellung der Anzahl der DIF Items für den Kalibrierungszeitpunkt 100 Testpersonen verzichtet.

5.2.6. Berücksichtigung des Standardfehlers

Bei diesem Schema wurde der Standardfehler als zusätzliches Vorgabekriterium gewählt (Kapitel III5.2.2, Lösungsansatz I: Verwendung des Standardfehlers als Vorgabekriterium). Eine bestimmte Anzahl an Items mit dem aktuell kleinsten Standardfehler, wurde von der weiteren Vorgabe im Rahmen der Kalibrierung vorläufig ausgeschlossen. Die nicht ausgeschlossenen Items werden anhand der maximalen Information vorgegeben. Dies führt dazu, dass stets nur jene Items zur adaptiven Vorgabe verwendet wurden, über die am wenigsten Information vorhanden ist. Die Prätests wurden mit 0 (entspricht dem Vorgabekriterium der maximalen Information), 15 und 30 ausgeschlossenen Items durchgeführt. Im Rahmen dieses Prätests wurde auf eine ausschließliche Vorgabe anhand des Standardfehlers verzichtet (entspricht 45 ausgeschlossenen Items), da sich diese Vorgabestrategie schon während der Entwicklung nicht bewährt hat.

Tabelle 8:

Ausgeschlossene Items aufgrund des Standardfehlers

Ausgeschlossene Items	0	15	30
Korrelationskoeffizient Items 100	.961 (.006)	.959 (.005)	.955 (.011)
Korrelationskoeffizient Personen 100	.853 (.004)	.854 (.004)	.852 (.005)
Prüfung auf DIF möglich 100	0	0	1
Korrelationskoeffizient Items 250	.983 (.006)	.984 (.004)	.982 (.004)
Korrelationskoeffizient Personen 250	.860 (.005)	.860 (.004)	.859 (.004)
Prüfung auf DIF möglich 250	16	19	22
DIF falsch positiv 250	2.625 (1.310)	1.632 (1.211)	.727 (.927)
DIF richtig positiv 250	2.688 (1.138)	3.211 (.713)	3.227 (.087)
Korrelationskoeffizient Items 500	.991 (.003)	.992 (.002)	.992 (.002)
Korrelationskoeffizient Personen 500	.862 (.003)	.863 (.003)	.862 (.003)
Prüfung auf DIF möglich 500	24	24	25
DIF falsch positiv 500	2.833 (1.685)	2.041 (1.546)	1.34 (.550)
DIF richtig positiv 500	3.917 (.282)	3.875 (.338)	4 (.0)

Hinweis: Standardfehler sind in Klammern

Tabelle 8 zeigt, dass bei 5 oder 30 aufgrund des Standardfehlers ausgeschlossenen Items, tendenziell seltener fälschlicherweise DIF festgestellt wird als dies bei der Kalibrierung anhand der maximalen Information (0 ausgeschlossene Items) der Fall ist. Bei 30 ausgeschlossenen Items wurde zu allen Kalibrierungszeitpunkten (nach 100; 250; 500 Personen) stets der niedrigste mittlere Korrelationskoeffizient der geschätzten mit den wahren Personenparametern festgestellt. Außerdem ist es bei einer Kalibrierung, die mehr Items aufgrund des Standardfehlers ausschließt, häufiger möglich, alle Itemparameter in den beiden Gruppen zu schätzen. Das führt dazu, dass eine Prüfung auf DIF häufiger durchgeführt werden konnte. Besonders deutlich ist dies nach 250 Personen zu erkennen: während der Fischer und Scheiblechner z-Test bei 0 ausgeschlossenen Items 16 mal möglich war, konnte dieser bei 30 ausgeschlossenen Items 22 mal durchgeführt werden.

5.2.7. Adjustierung der Itemparameterschätzer

Hierbei handelt es sich um Lösungsansatz II (Kapitel III 5.2.3, Adjustierung der Itemparameter). Die geschätzten Itemparameter werden in Abhängigkeit vom Standardfehler in Richtung 0 adjustiert. Dies soll dazu führen, dass für Itemparameter, die als sehr leicht oder sehr schwer eingeschätzt wurden, neutralere Werte für die weitere Vorgabe angewendet werden. Der *z*-

Wert zur Adjustierung betrug .5. In den Spalten mit 15 und 30 ausgeschlossenen Items wird dieses Vorgehen mit dem Lösungsansatz I Verwendung des Standardfehlers als Vorgabekriterium (Kapitel III 5.2.2) kombiniert. Die Prätests wurden mit 0,; 15 und 30 ausgeschlossenen Items durchgeführt. Tabelle 9:

Adjustierung der Itemparameterschätzer

Ausgeschlossene Items	0	15	30
Korrelationskoeffizient Items 100	.960 (.008)	.957 (.011)	.954 (.010)
Korrelatioskoeffiziet Personen 100	.854 (.003)	.854 (.004)	.852 (.006)
Prüfung auf DIF möglich 100	0	0	2
Korrelationskoeffizient Items 250	.983 (.004)	.985 (.004)	.984 (.004)
Korrelationskoeffizient Personen 250	.860 (.003)	.861 (.004)	.860 (.004)
Prüfung auf DIF möglich 250	21	20	23
DIF falsch positiv 250	2.429 (1.469)	1.5 (1.235)	.870 (1.014)
DIF richtig positiv 250	2.952 (.590)	3.2 (.834)	2.957 (.825)
Korrelationskoeffizient Items 500	.991 (.002)	.992 (.002)	.992 (.002)
Korrelationskoeffizient Personen 500	.863 (.004)	.863 (.002)	.863 (.004)
Prüfung auf DIF möglich 500	25	24	25
DIF falsch positiv 500	2.56 (1.356)	1.542 (1.817)	1.2 (1.080)
DIF richtig positiv 500	3.76 (.436)	3.917 (.282)	3.92 (.277)

Hinweis: Standardfehler sind in Klammern

Tabelle 9 zeigt, dass im direkten Vergleich zu den Ergebnissen der nicht adjustierten Itemparametern (Kapitel III 5.2.6) keine Verbesserung erkennbar ist. Da diese Vorgabestrategie die Komplexität des Programms erhöht, ohne, dass sich die Korrelation zwischen den wahren und geschätzten Werten erhöht, wurde diese Lösungsstrategie verworfen.

5.2.8. Veränderung des Vorgabekriteriums

Erneut werden Items mit dem geringsten Standardfehler ausgeschlossen. Der Unterschied zu den vohergehenden Prätests ist, dass, sobald alle Items geschätzt werden können (und keine Problemitems mehr vorhanden sind), die Vorgabe nur noch anhand der maximalen Information erfolgt. Es handelt sich also um Lösungsansatz III (Veränderung des Vorgabekriteriums, Kapitel III 5.2.4,) in Kombination mit Lösungsansatz I (Verwendung des Standardfehlers als Vorgabekriterium, Kapitel III 5.2.2). Die Prätests wurden mit 15, 30 und 45 aufgrund des Standardfehlers

ausgeschlossenen Items durchgeführt. Sobald alle Problemitems geschätzt werden konnten, erfolgte die Vorgabe anhand er maximalen Information.

Tabelle 10:

Veränderung des Vorgabekriteriums

Ausgeschlossene Items	15	30	45
Korrelationskoeffizient Items 100	.959 (.006)	.959 (.009)	.952 (.013)
Korrelationskoeffizient Personen 100	.854 (.005)	.854 (.005)	.854 (.007)
Prüfung auf DIF möglich 100	0	0	8
Korrelationskoeffizient Items 250	.983 (.005)	.984 (.003)	.983 (.004)
Korrelationskoeffizient Personen 250	.860 (.004)	.861 (.005)	.860 (.004)
Prüfung auf DIF möglich 250	17	23	25
DIF falsch positiv 250	1.765 (1.147)	1.233 (.971)	.76 (.723)
DIF richtig positiv 250	3.235 (.752)	2.93 (.907)	2.64 (1.319)
Korrelationskoeffizient Items 500	.992 (.002)	.992 (.002)	.992 (.002)
Korrelationskoeffizient Personen 500	.864 (.003)	.864 (.004)	.862 (.003)
Prüfung auf DIF möglich 500	24	25	25
DIF falsch positiv 500	2.292 (1.546)	1.563 (1.243)	1.56
DIF richtig positiv 500	3.875 (.338)	3.906 (.296)	3.68

Hinweis: Standardfehler stehen in Klammer. Da sich das Vorgabekriterium verändert, sind die ausgeschlossenen Items nur so lange gültig, bis keine Problemitems mehr vorhanden sind. Ab dann erfolgt die Itemvorgabe anhand der maximalen Information.

Tabelle 10 zeigt, dass, wenn sich die Vorgabe stärker am Kriterium der maximalen Information (also weniger ausgeschlossene Items) orientiert, so steigt das Risiko des Fehlers 1. Art des Fischers und Scheiblechners z-Test (DIF falsch positiv). Außerdem wurde bei 15 und 30 ausgeschlossenen Items seltener eine Prüfung auf DIF durchgeführt als bei 45 ausgeschlossenen Items, da häufiger nicht in beiden Gruppen die Schätzung aller Itemparameter möglich war.

5.2.9. Zusammenfassung der Prätests

Die verstärkte Vorgabe anhand des Kriteriums der maximalen Information (weniger ausgeschlossene Items) erwies sich vor allem in Kombination mit dem Fischer und Scheiblechner z-Test als problematisch. Hier konnte wiederholt ein tendenziell höheres Risiko des Fehlers 1. Art festgestellt werden. Insgesamt erwies es sich als hilfreich nur jene Items vorzugeben, über die bisher wenig Information zur Verfügung steht. Items über die bereits mehr Information akkumuliert

werden konnte und die damit einen geringeren Standardfehler aufweisen, werden ausgeschlossen. Dies ist vor allem am Beginn der Testkalibrierung hilfreich, damit vom gesamten Itempool ein erster Eindruck gewonnen werden und darauf die weitere Kalibrierung aufgebaut werden kann. Die Berücksichtigung des Standardfehlers (Kapitel III 5.2.6) erscheint also als eine erfolgsversprechende Vorgehensweise.

Die Adjustierung der Itemparameter in Abhängigkeit des Standardfehlers führte weder zu erkennbar besseren Korrelationen der geschätzten zu den wahren Parametern, noch konnte dadurch die Erkennung der Items mit DIF verbessert werden. Deshalb wird dieser Lösungsansatz im Rahmen dieser Arbeit nicht weiter verfolgt.

Letztendlich wurde als Strategie für die eigentliche Simulationsstudie die Veränderung des Vorgabekriteriums, sobald keine Problemitems mehr vorhanden sind, gewählt. Bis zu diesem Zeitpunkt werden stets jene 30 Items mit den geringsten Standardfehler ausgeschlossen. Es handelt sich also um jene Vorgabe deren Ergebnisse des Prätests sich in Tabelle 10, Spalte 2 befinden. Diese Strategie scheint ein guter Kompromiss zwischen den getesteten Varianten zu sein, da in allen überprüften Bereichen zufriedenstellende Ergebnisse erzielt werden konnten.

5.3. Das Programm im Detail

Um zu verstehen, wie die Datengenerierung und letztendlich die Testkalibrierung abgelaufen ist, erfolgt in weiterer Folge eine Darstellung der Funktionsweise des Algorithmus. Der entwickelte R-Code befindet sich in Kapitel 5.4.

5.3.1. Laden der notwendigen R Pakete

Die Software ruft im ersten Schritt die Pakete „sirt“ (Robitzsch, 2014) und „catR“ (Magis & Gilles, 2012) auf. Sirt wird zur Berechnung der Schwierigkeitsparameter und Personenparameter nach jeder Testperson verwendet, catR dient zur Computerisierten Adaptiven Vorgabe (CAT - „computerized adaptive testing“). Außerdem wird das Arbeitsverzeichnis festgelegt, in welchem die simulierten Daten gespeichert werden sollen.

5.3.2. Laden der notwendigen Funktionen

Im zweiten Schritt werden die eigenhändig programmierten Funktionen geladen, die das Rückgrat der Software darstellen. Das fertige Programm ist eine Abfolge der einzelnen Schritte

dieser Funktionen. Eine Funktion selbst ist wiederum eine Kombination von meist mehreren Anweisungen. Die verschiedenen R Pakete sind letztendlich nichts anderes als eine Sammlung an (meist eher komplexen) Funktionen zu einem bestimmten Thema.

5.3.3. Abruf der Modelleinstellungen

Hier werden die unterschiedlichen Modelleinstellungen wie Stichprobengröße n (500), Itempool k (60), Itemparameter (Tabelle 2) und Anzahl an Replikationen r (1000) vorgenommen. Die Einstellungen werden außerhalb des eigentlichen Simulationsalgorithmus festgelegt. Dies dient in erster Linie der besseren Übersichtlichkeit und ermöglicht es ohne größeren Aufwand den Algorithmus mit anderen Einstellungen zu überprüfen.

5.3.4. Simulation

Nun kann die eigentliche Simulation der adaptiven Itemkalibrierung starten. Diese wird so lange wiederholt bis entweder die eingestellte Anzahl an Replikationen erreicht ist, der Algorithmus vom Benutzer abgebrochen wird oder ein Fehler auftritt. Im letzteren Fall gibt R eine Meldung aus, die auf die Ursachen des Abbruchs hinweist.

5.3.5. Datensatzerstellung

Das Programm simuliert die vollständigen Antworten von n = 500 Personen und k = 60 Items (im Programm gespeichert als Objekt „datall“). Die n Personen teilen sich wiederum in die zwei Gruppen m und w zu jeweils 250 Personen auf. Die Reihenfolge zwischen männlichen und weiblichen Testpersonen ist innerhalb des Gesamttests zufällig. 4 der k Items weisen eine DIF in der Gruppe w gegenüber Gruppe m auf. Das sind die Items 5, 20, 35 und 50 (Tabelle 2).

Außerdem wird das Objekt „dat“ erstellt, welches nur die bereits gegeben Antworten der simulierten Testpersonen (VP) beinhaltet. Bis zur ersten Testperson besteht „dat“ also lediglich aus einer Matrix mit n*k = 30 000 Elementen ohne Antwort (NA).

Jede Replikation ist klar definiert durch einen Seed (Objekt „anfang“), wodurch die Ergebnisse reproduzierbar sind. Der Seed beginnt mit 1 und wird mit jeder Replikation um 1 erhöht. Die letzte Replikation erhält somit den Seed 100.

5.3.6. Simulation auf Testpersonenebene

Im ersten Schritt werden die Zähler der Testpersonen auf 0 gesetzt, wobei zwischen den Zählern „am“, „aw“ und „a“ unterschieden wird: „am“ wird immer dann um + 1 erhöht wenn das aktuell simulierte Antwortverhalten von einer VP der Gruppe „m“ stammt, dasselbe gilt äquivalent für den Zähler „aw“ und Gruppe „w“. Das Objekt „a“ wird bei jeder VP, unabhängig von der Gruppenzugehörigkeit, um +1 erhöht und stellt somit dar, die wievielte VP das Verfahren aktuell bearbeitet.

Bis jedes Item mindestens einmal in der jeweiligen Gruppe vorgegeben wurde, werden für die aktuelle VP jene Items, die bisher noch nicht in der entsprechenden Gruppe bearbeitet wurden, zufällig ausgewählt.

Für Items, bei denen noch keine Parameter berechnet werden konnten, werden die Schwierigkeiten in Abhängigkeit von der bisherigen Lösungshäufigkeit angenommen. Jene Items, über welche bisher noch keine Information im Sinne des Rasch-Models vorhanden sind („Problemitems“, Kapitel III.5.2.1) und welche immer gelöst wurden, bekommen einen Parameter zugeordnet, der um -0,01 kleiner ist als der Schätzer des leichtesten bisher bekannten Items. Items, die bisher nicht gelöst wurden, erhalten umgekehrt einen um 0,01 höheren Wert als das schwerste bekannte Item der jeweiligen Gruppe.

Die Berechnung der Personenparameter erfolgt unter der Anwendung der WML-Schätzung mittels des R-Pakets catR (Magis & Gilles, 2012), welches außerdem für die adaptive Vorgabe angewendet wurde. Da der Personenparameter erst nach dem zweiten Item berechnet werden kann, wird als erstes jenes Item dessen Schwierigkeitsparameter am nächsten zu 0 ist und als zweites, abhängig von dem Ergebnis des ersten bearbeiteten Items, entweder das am schwersten oder leichteste geschätzte Item des Pools, vorgegeben.

Sobald jedes Item mindestens einmal pro Gruppe vorgegeben wurde, erfolgt die weitere Vorgabe anhand der minimalen Differenz zwischen Fähigkeitsschätzer der jeweiligen VP und Itemparameter. Allerdings sind, so lange noch Problemitems (Kapitel III 5.2.1) vorhanden sind, jene 30 Items mit dem kleinsten Standardfehler (und somit der größten Menge an vorhandener Information) von der adaptiven Vorgabe ausgenommen. Äquivalent zu den Itemparameter werden die Standardfehler für Items, die bisher noch nicht berechnet werden konnte, um 0,01 extremer als die bisher höchsten bekannten angenommen. Für Problemitems, welche sowohl gelöst als auch nicht gelöst wurden, wurde zur weiteren Vorgabe ein neutraler Parameter angenommen (0).

Für den Fall, dass nach der Bearbeitung des 14. Items der Personenparameter größer 1,5 oder kleiner -1,5 geschätzt wurde, wird als Vorgabe des letzten Items ein Problemitem erzwungen. Personen mit einem Fähigkeitsschätzer größer 1,5 erhalten jene Problemitems, die bisher seltener als in 50% der Vorgaben gelöst werden konnten. Umgekehrt erhalten Personen mit einem Fähigkeitsschätzer kleiner -1,5, jene Problemitems die zu 50% oder mehr gelöst werden konnten. Damit soll erreicht werden, dass möglichst früh alle Itemparameter geschätzt werden können.

Sobald die Parameter des gesamten Itempools berechnet werde konnten, werden für die restliche Testkalibrierung keine Items mehr aufgrund des Standardfehlers ausgeschlossen.

5.3.7. Simulation auf Verfahrensebene

Nach jeder Testperson, die die erforderlichen 15 Items bearbeitet hat, erfolgt nach Möglichkeit die Itemparameterschätzung unter der Anwendung des Paket sirt (Robitzsch, 2014) und die Itemparameter werden somit für die nächste VP aktualisiert. Außerdem werden in dieser Phase Problemitems erkannt und gespeichert, da diese, wie bereits angemerkt, bei der Itemvorgabe besonders berücksichtigt werden. Die Kalibrierung erfolgt abhängig von der Gruppenzugehörigkeit der jeweiligen Testperson. Es existieren somit für beide Gruppen unterschiedliche Itemparameter. Dies ermöglicht es, dass der Fischer und Scheiblechner z-Test (Kapitel II 2.2.3) ohne zusätzlich Berechnung des Rasch-Modells angewendet werden kann, da in den Ergebnissen der Parameterschätzung mit sirt bereits die Itemparameter sowie Standardfehler der jeweiligen Gruppe enthalten sind.

Ab der 10. Testperson erfolgt nach jeder Testperson eine Prüfung auf DIF und bei signifikantem Ergebnis, wird das oder die betreffende Item(s) von der weiteren Vorgabe ausgeschlossen. Items, auf die das zutrifft, werden allerdings weiterhin bei der Kalibrierung berücksichtigt. So kann es durchaus vorkommen, dass ein signifikantes Ergebnis bei fortschreitender Kalibrierung wieder nicht signifikant wird und das Item wieder zur Vorgabe zur Verfügung steht.

Das Signifikanzniveau des Fischers und Scheiblechners z-Test wird mit einem Risiko des Fehlers 1. Art von 1% und somit einem z-Wert von 2,58 festgelegt. Da vermieden werden soll, dass schon in einem frühen Stadium der Kalibrierung bei noch sehr ungenauen Ergebnissen viele Items ausgeschlossen werden, wird der z-Wert schrittweise dem Endwert angenähert:

- bis zur 30. VP: z-Wert größer 3,5

- ab der 30. VP: z-Wert größer 3
- ab der 50. VP: z-Wert größer 2,58

Nach 100, 250 und 500 Testpersonen wird der Test 5 000 Testpersonen (2500 Gruppe w, 2500 Gruppe m) vorgegeben, um zu überprüfen wie genau dieser aktuell, die Personenparameter (und somit das letztendlich eigentlich Interessierende) misst. Items für die bisher noch keine Parameter geschätzt werden konnten, werden von der Vorgabe ausgeschlossen. Gemessen wird die Abweichung zwischen wahren Personenparameter und geschätztem Wert, sowie die Korrelation zwischen diesen beiden Variablen. Die Ergebnisse der einzelnen Testpersonen sowie der Itemparameter werden jeweils unmittelbar nach deren Berechnung gespeichert und aus dem Zwischenspeicher gelöscht, um Systemressourcen für die weitere Kalibrierung zu sparen.

Die beschriebenen Vorgänge auf Verfahrensebene und Testpersonenebene werden so lange wiederholt, bis der Test von der eingestellten Anzahl an „n“ Testpersonen (500) absolviert wurde.

5.3.8. Datenspeicherung

Nach 500 Testpersonen, also nach Abschluss der Kalibrierung eines Tests, werden die Ergebnisse (Korrelationskoeffizient geschätzte Itemparameter mit den wahren Werten ohne Items mit DIF, Abweichung geschätzte Itemparameter von den wahren Werten ohne Items mit DIF, Anzahl falsch erkannter positiver DIF Items, Anzahl richtig erkannter positiver DIF Items) und das konkrete Antwortverhalten der Testpersonen des aktuellen Verfahrens (sowie die Ergebnisse der Kalibrierung desselben Datensatzes mittels Zufallsvorgabe), zu verschiedenen Kalibrierungsstadien (nach 100, 250 & 500 Testpersonen) gespeichert.

5.3.9. Warnmeldungen

Während der Entwicklung des Programms erschienen wiederholt Warnmeldungen, die darauf hinwiesen, dass die Berechnung des Korrelationskoeffizienten gescheitert ist, da die Standardabweichung 0 ist. Diese Warnmeldungen traten vorwiegend zu Beginn der Testkalibrierung in unregelmäßigen Abständen auf, wenn bisher nur wenige VP (a <= 150) getestet wurden. Häufig kam es vor, dass die Meldung bei ein und derselben VP mehrmals erschien und bei der nächsten VP wieder verschwand.

Eine nähere Untersuchung des Programmpakets ergab, dass die Meldungen durch das R-Paket sirt (Robitzsch, 2014) in der Funktion rasch.mml2 verursacht wurden. Diese Funktion dient der

Berechnung des Rasch-Models und wurde im Rahmen dieser Arbeit zur Schätzung der Itemparameter verwendet.

Trotz der Warnmeldung konnten die Itemparameter scheinbar korrekt erhoben werden. Es bleibt zu vermuten, dass sich die fehlgeschlagene Berechnung des Korrelationskoeffizient auf andere, für diese Arbeit weniger interessante Outcomes auswirkte, zumal die Erhebung der Korrelation für die Itemparameter innerhalb der MML-Schätzung (Kapitel II.2.1.4) nicht erforderlich ist. Doch selbst wenn die Itemparameter verfälscht wären, hätte dies zur Folge, dass lediglich einzelne Testpersonen nicht die für die Kalibrierung optimalen Items erhielten, da die Warnmeldung in aller Regel bei der nächsten Person nicht mehr erschien. Der Effekt auf die Ergebnisse insgesamt sollte dadurch eher gering sein. Aus diesen Gründen wurden letztendlich die Warnmeldungen während der Anwendung der Funktion rasch.mml2 bis zur 15. Testperson deaktiviert.

5.4. R-Code

```
library(sirt)
library(catR)
setwd("D:\\sim\\p1")

#####################################################
####################funktionen#######################
#####################################################

#####################################################
#datsav speichert die generierten daten in einer liste
datsav <- function(dat, name, env = global){
  if(missing(name)) name <- deparse(substitute(dat))
  NAME <- toupper(name)
  newdat <- as.name(NAME)
  rds <- ".rds"
  NAMErds <- paste(NAME, rds, sep = "")
  listdat <- c()
  if(file.exists(NAMErds)) listdat <- readRDS(NAMErds)
  listdatname <- deparse(substitute(listdat))
  if(!is.null(listdat)){
    listdat[length(listdat) + 1] <- list(dat)
  } else{
    listdat <- list(dat)
  }
  saveRDS(listdat, file = NAMErds)
}

#datoutsav speicher die outputs in einer .csv datei
datoutsav <- function(datout, name){
  if(missing(name)) name <- paste(deparse(substitute(datout)), ".csv", sep = "")
  if(!file.exists(name)){
    write.csv2(datout, name, row.names = F)
  } else {
    datfalt <- read.csv2(name)
    datout <- rbind(datfalt, datout)
    write.csv2(datout, name, row.names = F)
  }
}
```

```
##################################################
#minimaxi definiert bei problemitems die itemparameter
minimaxi <- function(dat, itemspar, schwierigkeit){

  spaltmittel <- colMeans(dat, na.rm = T)

  maxi <- which(spaltmittel == 0) #items die nie gelöst wurden -> maximal schwer
  mini <- which(spaltmittel == 1) #items, die immer gelöst wurden

  if(any(mini) | any(maxi)) {
    if(any(mini)){
      itemspar[mini, 2] <- 0
      ifelse(missing("schwierigkeit"), itemspar[mini, 2] <- min(itemspar[, 2] - .01),
             itemspar[mini, 2] <- min(itemspar[, 2]))
    }
    if(any(maxi)){
      itemspar[maxi, 2] <- 0
      ifelse(missing("schwierigkeit"), itemspar[maxi, 2] <- max(itemspar[, 2] + .01),
             itemspar[maxi, 2] <- max(itemspar[, 2]))
    }
  }
  return(itemspar)
}


#################die funktion datimp entfernt problematische items
datimp <- function(dat){

  spaltmittel <- colMeans(dat, na.rm = T)
  reihmittel <- rowMeans(dat, na.rm = T)

  minp <- which(reihmittel == 0) #personen die keine items gelöst haben
  maxp <- which(reihmittel == 1) #personen die alle items gelöst haben
  missp <- which(rowSums(is.na(dat) == F) == 0)

  mini <- which(spaltmittel == 0) #items die nie gelöst wurden
  maxi <- which(spaltmittel == 1) #items, die immer gelöst wurden
  missi <- which(colSums(is.na(dat) == F) == 0)

  ausi <- c(mini, maxi, missi) #items die ignoriert werden sollen
```

```
  ausp <- c(minp, maxp, missp) #personen die ignoriert werden sollen

  if(any(ausi)) {
    dat <- dat[, -ausi]  #items die ignoriert werden sollen
    if(is.matrix(dat)) return(dat)
    else return(matrix(dat))
  }
  if(any(ausp)) {
    dat <- dat[-ausp, ] #personen die ignoriert werden sollen
    if(is.matrix(dat)) return(dat)
    else return(matrix(dat))
  }
}

###################prüft ob der datensatz noch verbesserwert werden kann
datcheck <- function(dat) {
  if(length(dat) == 0)
    return(F)
  if(sum(is.na(dat) == F) == 0)
    return(F)
  if(sum(is.na(dat[1, ]) == F) < 2)
    return(F)
  if(sum(is.na(dat[2, ]) == F) < 2)
    return(F)
  else {
    spaltmittel <- colMeans(dat, na.rm = T)
    reihmittel <- rowMeans(dat, na.rm = T)
    if(any(na.exclude(spaltmittel == 1)) | any(na.exclude(spaltmittel == 0)) | any(na.exclude(reihmittel == 1)) |
         any(na.exclude(reihmittel == 0)) | any(rowSums(is.na(dat) == F) == 0) | any(colSums(is.na(dat) == F) == 0))
      return(T)
    else return(F)
  }
}

#rmcomp gibt schwierigkeitsparameter, standarderror und obere sowie untere bereiche der ki aus
rmcomp <- function(dat){
  if(datcheck(dat)) {
    datneu <- dat
    while(datcheck(datneu)) datneu <- datimp(datneu)
  }
  ifelse(exists("datneu"), raschmodell <- rasch.mml2(datneu, progress = F), raschmodell <- rasch.mml2(dat, progress = F))
```

```
  schwierigkeit <- raschmodell$item[, "b"]
  names(schwierigkeit) <- raschmodell$item[, "item"]
  if(length(schwierigkeit) != k) return("error - length schwierigkeit != k")
  standarderror <- raschmodell$se.b
  names(standarderror) <- names(schwierigkeit)
  konfi <- matrix(NA, ncol = 4, nrow = k)
  colnames(konfi) <- c("schwierigkeit", "standarderror", "KI+", "KI-")
  konfi[, 1] <- schwierigkeit
  konfi[, 2] <- standarderror
  konfi[, 3] <- konfi[, 1] + 1.96 * konfi[, 2]
  konfi[, 4] <- konfi[, 1] - 1.96 * konfi[, 2]
  return(konfi)
}

#rmcompdif erkennt items mit dif
rmcompdif <- function(dat){
  konfim <- rmcomp(subset(dat, row.names(dat) == "m"))
  konfiw <- rmcomp(subset(dat, row.names(dat) == "w"))
  if(!is.matrix(konfim)) return("error - length schwierigkeit group m != k")
  if(!is.matrix(konfiw)) return("error - length schwierigkeit group w != k")
  ztest <- abs((konfim[, 1] - konfiw[, 1])/sqrt(konfiw[, 2]^2 + konfim[, 2]^2))
  ausschluss <- which(ztest > 2.58)
  if(a < 300) ausschluss <- which(ztest > 3.5)
  if(a < 500 & a >= 300) ausschluss <- which(ztest > 3)
  if(a >= 500) ausschluss <- which(ztest > 2.58)
  return(ausschluss)
}

#liefert den output, wie er zum speichern notwendig ist
rmoutput <- function(dat, datrand, trueitempar){
  aus <- rmcompdif(dat)
  if(is.character(aus)) aus <- rep(1, 100)
  ausz <- rmcompdif(datrand)
  if(is.character(ausz)) ausz <- rep(1, 100)
  konfi <- rmcomp(dat)
  konfiz <- rmcomp(datrand)
  #adaptive vorgabe
  if(is.matrix(konfi)){
    cora <- cor(konfi[- c(5, 20, 35, 50), 1], trueitempar[- c(5, 20, 35, 50)])
    re <- mean(sqrt((konfi[- c(5, 20, 35, 50), 1] - trueitempar[- c(5, 20, 35, 50)])^2))
    se <- mean(konfi[- c(5, 20, 35, 50), 2])
```

```
    ausrichtig <- length(which(is.element(aus, c(5, 20, 35, 50))))
    ausfalsch <- length(which(!is.element(aus, c(5, 20, 35, 50))))
  } else{
    cora <- NA
    re <- NA
    se <- NA
    ausrichtig <- NA
    ausfalsch <- NA
  }
  #zufall
  if(is.matrix(konfiz)){
    corz <- cor(konfiz[- c(5, 20, 35, 50), 1], trueitempar[- c(5, 20, 35, 50)])
    rez <- mean(sqrt((konfiz[- c(5, 20, 35, 50), 1] - trueitempar[- c(5, 20, 35, 50)])^2))
    sez <- mean(konfiz[- c(5, 20, 35, 50), 2])
    ausrichtigz <- length(which(is.element(ausz, c(5, 20, 35, 50))))
    ausfalschz <- length(which(!is.element(ausz, c(5, 20, 35, 50))))
  } else{
    corz <- NA
    rez <- NA
    sez <- NA
    ausrichtigz <- NA
    ausfalschz <- NA
  }
  return(data.frame(cora, re, se, ausrichtig, ausfalsch, corz, rez, sez, ausrichtigz, ausfalschz))
}


############################################################
##############modelleinstellungen###########################
############################################################
#der seed start
anfang <- 1
if(file.exists("datout50.csv")) anfang <- anfang + nrow(read.csv2("datout50.csv"))


#anzahl an items
k <- 60

#itemrange von itemrangemin bis itemrangemax nur relevant, wenn itemparameter nicht zufällig generiert werden
itemrangemin <- -3
itemrangemax <- 3
```

```
#ab wann die prüfung auf dif erfolgen soll
m <- 100

#anzahl an Testpersonen
n <- 500

#anzahl an replikationen
r <- 1000
if(file.exists("datout50.csv")) r <- r - nrow(read.csv2("datout50.csv"))

#anzahl an items je person
j <- 15

#z wert für die konfidenzintervalle zur adjustierung der vorgabeitemparameter (1.96 = .05 irrtumswahrscheinlichkeit)
z <- .0

#wie stark soll ich die vorgabe an dem standardfehler oder an der passung fähigkeit/schwierigkeit orientieren?
#werte von 0 bis (k-j) sind erlaubt. 0 -> Vorgabe nur anhand des standardfehlers der itemschätzer,
#(k-j) -> vorgabe nur anhand der passung. 14 liefert bei k = 60 gute ergebnisse.
passung <- 15

#ab der wievielten Testperson die items volladaptiv vorgegeben werden sollen (entspricht passung k-j)
#wenn passungvoll > n, dann bleibt diese bedingung inaktiv
passungvoll <- 10000

#bei difvor = T werden items mit einem hohen z-wert nach fischer und scheiblechner z-test bevorzugt vorgegen
difvor <- F

#sobald über alle items information vorhanden ist, wird die passung auf 45 gesetzt (bei F, bleibt die passung unverändert)
probab <- T



#########################################################
####################datensatzerstellung##################
#########################################################

#"trueitempar" sind die wahren itemparameter
trueitempar <- seq(itemrangemin, itemrangemax, length = k)

```

```
244 names(trueitempar) <- c(1:k)
245
246 #welche items in welchem ausmaß eine dif aufweisen sollen
247 trueitempardif <- trueitempar
248 trueitempardif[5] <- trueitempardif[5] + 2
249 trueitempardif[20] <- trueitempardif[20] - 2
250 trueitempardif[35] <- trueitempardif[35] + 2
251 trueitempardif[50] <- trueitempardif[50] - 2
252
253
254
255 ################################################
256 ###################simulation###################
257 ################################################
258 zeitstart <- date()
259
260 for(i in 1:r){
261
262   #"datall" ist der datensatz, der auftreten würde, wenn jede person alle items bearbeitet
263   set.seed(anfang + i)
264   datall <- sim.raschtype(rnorm(n/2), trueitempar)
265   datall <- rbind(datall, sim.raschtype(rnorm(n/2), trueitempardif))
266   colnames(datall) <- c(1:k)
267
268   #die Testpersonen werden nach "geschlecht" zufällig angeordnet
269   verteilung <- sample(1:n)
270   datall <- datall[verteilung, ]
271
272   #die reihen von datall werden nach geschlecht bezeichnet
273   row.names(datall) <- c(rep("m", n))
274   row.names(datall)[which(verteilung > n/2)] <- c(rep("w", n/2))
275
276   #"dat" zeigt nur die bereits bearbeiteten daten an
277   dat <- matrix(NA, nrow = n, ncol = k)
278   colnames(dat) <- c(1:k)
279   #die reihen von data werden nach geschlecht bezeichnet
280   row.names(dat) <- c(rep("m", n))
281   row.names(dat)[which(verteilung > n/2)] <- c(rep("w", n/2))
282
283
284   ##########################################################################################
```

```
#################die zufallsvorgabe als alternativmodell#########################################################
#################################################################################################################
zufallsvor <- function(x){
  x[sample(1:k)[1:(k-j)]] <- NA
  return(x)
}
datrand <- t(apply(datall, 1, zufallsvor))


################################################
##############Testpersonenebene###############
################################################
#a ist die nummer der Testperson
am <- 0
aw <- 0
a <- 0
ausschluss <- c()

for(i in 1:n){
  a <- a + 1
  sefa <- passung

  ifelse(row.names(dat)[a] == "m", am <- am + 1, aw <- aw +1)

  #anzeige der aktuellen vp nummer
  print("vp nummer:")
  print(a)

  #"itemspar" sind die schätzer der itemparameter; sollten noch keine berechnet worden sein werden werte = 0 angenommen
  if(exists("itemsparm") == F){
    itemsparm <- matrix(0, nrow = k, ncol = 4)
    itemsparm[, 1] <- 1
    itemsparm[, 4] <- 1
  }

  if(exists("itemsparw") == F){
    itemsparw <- matrix(0, nrow = k, ncol = 4)
    itemsparw[, 1] <- 1
    itemsparw[, 4] <- 1
  }

```

```
326       namen <- c(1:k)
327       rownames(itemsparw) <- namen #muss jedesmal wieder neu benannt werden, da itemspar auf verfahrensebene neu
328       #generiert wird
329       rownames(itemsparm) <- namen
330
331
332       #die schwierigkeitsparameter für items, bei denen noch keine information vorhanden ist, werden definiert
333       #gruppe w
334       ifelse(exists("schwierigkeitw"),
335              itemsparw <- minimaxi(subset(dat, row.names(dat) == "w"), itemsparw, schwierigkeitw),
336              itemsparw <- minimaxi(subset(dat, row.names(dat) == "w"), itemsparw))
337       if(exists("problemitemsw") & exists("schwierigkeitw")){
338         #für problemitems welche sowohl gelöst als auch nicht gelöst werden konnten, müssen keine extremeren werte angenommen werden
339         #alle anderen problemitems werden als extrem schwer (+ 0,01 als das schwierigste item) / extrem leicht (- 0,01) angenommen
340         problemitemsw <- problemitemsw[which(itemsparw[problemitemsw, 2] != 0)]
341         itemsparw[problemitemsw, 2]  <- (abs(itemsparw[problemitemsw, 2]) + .01) * (abs(itemsparw[problemitemsw, 2]) /
342                                                                                     itemsparw[problemitemsw, 2])
343       }
344
345       #die schwierigkeitsparameter für items, bei denen noch keine information vorhanden ist, werden definiert
346       #gruppe m
347       ifelse(exists("schwierigkeitm"),
348              itemsparm <- minimaxi(subset(dat, row.names(dat) == "m"), itemsparm, schwierigkeitm),
349              itemsparm <- minimaxi(subset(dat, row.names(dat) == "m"), itemsparm))
350       if(exists("problemitemsm") & exists("schwierigkeitm")){
351         #für problemitems welche sowohl gelöst als auch nicht gelöst werden konnten, werden die itemparameter = 0 gesetzt
352         #alle anderen problemitems werden als extrem schwer (+ 0,01 als das schwierigste item) / extrem leicht (- 0,01) angenommen
353         problemitemsm <- problemitemsm[which(itemsparm[problemitemsm, 2] != 0)]
354         itemsparm[problemitemsm, 2]  <- (abs(itemsparm[problemitemsm, 2]) + .01) * (abs(itemsparm[problemitemsm, 2]) /
355                                                                                     itemsparm[problemitemsm, 2])
356       }
357
358       #"catitems" wird von catR kreiert und ist für die adaptive vorgabe notwendig
359       catitemsw <- createItemBank(itemsparw, model ="1PL", cb = F, thMin=-10, thMax=10)
360       catitemsm <- createItemBank(itemsparm, model ="1PL", cb = F, thMin=-10, thMax=10)
361
362       #bearbges sortiert die items nach der häufigkeit, die diese vorgegeben wurden
363       ifelse(am > 1 & row.names(dat)[a] == "m",
364              bearbgesm <- as.numeric(names(sort(colSums(is.na(subset(dat, row.names(dat) == "m")) == F), decreasing = T))), bearbgesm <-
365   c(0))
366       ifelse(aw > 1 & row.names(dat)[a] == "w",
```

```
            bearbgesw <- as.numeric(names(sort(colSums(is.na(subset(dat, row.names(dat) == "w")) == F), decreasing = T))), bearbgesw <-
c(0))

    #vor einer neuen vp werden die personenparameter wieder auf null gesetzt:
    bearbeitet <- c()
    faehigkeit <- c(0)

    if(probab == T){
      #sobald keine problemitems mehr existieren wird sefa auf k-j gesetzt
      if(exists("schwierigkeitw")){
        if(row.names(dat)[a] == "w" & length(schwierigkeitw) == k & a > 20) sefa <- k-j
      }
      if(exists("schwierigkeitm")){
        if(row.names(dat)[a] == "m" & length(schwierigkeitm) == k & a > 20) sefa <- k-j
      }
    }

    #ab passungvoll wird nur noch vollständig adaptiv vorgegeben
    if(a > passungvoll) sefa <- k-j

    #x ergibt sich durch sefa und die anzahl an ausschlussitems und legt fest, wieviele items für die adaptive vorgabe zur
    #verfügung stehen
    ifelse(length(ausschluss > 0), x <- sefa + length(ausschluss), x <- sefa)
    if(x > k-j) x <- k-j

    #die adaptive vorgabe startet erst wenn jedes item mindestens einmal in der jeweiligen gruppe vorgegeben wurde
    #zuerst gruppe w
    if(row.names(dat)[a] == "w") {
      if(aw*j/k > 1) {
        bearbgesw <- c()
        if(exists("konfidenzintervallw")){
          #die k-j-x items mit der geringen standardfehler werden von der nächsten vorgabe ausgeschlossen
          bearbgesw <- as.numeric(names(sort(konfidenzintervallw[, 2])[1:(k-j-x)]))
          #sollte rein adaptiv vorgegeben werden, so wird bearbges immer als leer gesetzt
          if(x + length(ausschluss) >= (k-j)) bearbgesw <- c()
        }

        for(i in 1:j) {

          #auswahl des nächsten items
          nextcat <- nextItem(catitemsw, faehigkeit, out = c(bearbeitet, bearbgesw, ausschluss), randomesque = 1, method = "WL",
```

```
                        cbControl = NULL)
    naechstes <- as.numeric(nextcat[1])
    #sollte es problemitems passend zur vp geben dann wird zur vorgabe des letzten items, ein problemitem in der
    #richtung des geschätzten fähigkeitsparameters der aktuellen vp erzwungen
    if(faehigkeit < -1.5 & length(bearbeitet) == (j - 1) & exists("probleichtw")){
      naechstes <- sample(probleichtw, 1)
    }
    if(faehigkeit > 1.5 & length(bearbeitet) == (j - 1) & exists("probschwerw")){
      naechstes <- sample(probschwerw, 1)
    }

    #sollte es items geben, bei denen der z-wert hinsichtlich einer dif > 1.5 ist, wird die vorgabe als letztes item ein
    #solches erzwungen
    if(difvor == T){
      if(exists("ztest")){
        if(a > m & any(abs(ztest[-ausschluss]) > 1.5) & length(bearbeitet) == (j - 1) & exists("probleichtw") == F &
             exists("probschwerw") == F){
          difverdacht <- as.numeric(names(sort(ztest[-ausschluss], decreasing = TRUE)))[1]
          naechstes <- as.numeric(names(sort(abs(faehigkeit - konfidenzintervallw[difverdacht, 1]))))[1])
        }
      }
    }

    #bereits vorgegebene items werden gespeichert und stehen für eine erneute vorgabe bei der selben vp nicht mehr zur
    #verfügung
    bearbeitet <- c(bearbeitet, naechstes)

    #speichern des bearbeiteten items in der datenmatrix
    dat[a, naechstes] <- datall[a, naechstes]

    #ich kann erst bei 2 antworten die fähigkeitsparameter schätzen bis dahin wird dieser um 0,01
    #extremer als das schwerste/leichteste item angenommen
    if(length(bearbeitet) > 1){
      faehigkeit <- thetaEst(itemsparw[bearbeitet, ], dat[a, bearbeitet], method = "WL")
    }
    if(length(bearbeitet) == 1){
      if(mean(dat[a, ], na.rm = T) == 0) faehigkeit <- min(itemsparw[, 2] - .01)
      if(mean(dat[a, ], na.rm = T) == 1) faehigkeit <- max(itemsparw[, 2] + .01)

    }
  }
```

```
449
450           #bis jedes item mindestens einmal pro "geschlecht" vorgegeben wurde, werden die nächsten items zufällig ausgewählt
451           if(aw*j/k <= 1) {
452             if(aw == 1){
453               datallw <- subset(datall, row.names(datall) == "w")
454               zufallsvorgabe <- sample(datallw[aw, ], j)
455               dat[a, names(zufallsvorgabe)] <- zufallsvorgabe
456             }
457             if(aw > 1){
458               b <- aw-1
459               zufallsvorgabe <- sample(datallw[aw, - bearbgesw[1:(b*j)]], j)
460               dat[a, names(zufallsvorgabe)] <- zufallsvorgabe
461             }
462           }
463         }
464       }
465
466       #das selbe nun für gruppe m
467       if(row.names(dat)[a] == "m") {
468         if(am*j/k > 1) {
469           bearbgesm <- c()
470           if(exists("konfidenzintervallm")){
471             #die k-j-x items mit der geringen standardfehler werden von der nächsten vorgabe ausgeschlossen
472             bearbgesm <- as.numeric(names(sort(konfidenzintervallm[, 2])[1:(k-j-x)]))
473             #sollte rein adaptiv vorgegeben werden, so wird bearbges immer als leer gesetzt
474             if(x + length(ausschluss) >= (k-j)) bearbgesm <- c()
475           }
476
477           for(i in 1:j) {
478
479             #auswahl des nächsten items
480             nextcat <- nextItem(catitemsm, faehigkeit, out = c(bearbeitet, bearbgesm, ausschluss), randomesque = 1, method = "WL",
481                                 cbControl = NULL)
482             naechstes <- as.numeric(nextcat[1])
483             #sollte es problemitems passend zur vp geben dann wird die vorgabe des letzten items ein problemitem in der
484             #richtung des fähigkeitsparameters erzwungen
485             if(faehigkeit < -1 & length(bearbeitet) == (j - 1) & exists("probleichtm")){
486               naechstes <- sample(probleichtm, 1)
487             }
488             if(faehigkeit > 1 & length(bearbeitet) == (j - 1) & exists("probschwerm")){
489               naechstes <- sample(probschwerm, 1)
```

```
        }

        #sollte es items geben bei denen der z-wert hinsichtlich einer dif > 1.5 ist, wird die vorgabe als letztes item ein
        #solches erzwungen
        if(difvor == T){
          if(exists("ztest")){
            if(a > m & any(abs(ztest[-ausschluss]) > 1.5) & length(bearbeitet) == (j - 1) & exists("probleichtm") == F &
                 exists("probschwerm") == F){
              difverdacht <- as.numeric(names(sort(ztest[-ausschluss], decreasing = TRUE)))[1]
              naechstes <- as.numeric(names(sort(abs(faehigkeit - konfidenzintervallm[difverdacht, 1])))[1])
            }
          }
        }

        #bereits vorgegebene werden gespeichert und stehen für eine erneute vorgabe bei derselben vp nicht mehr zur verfügung
        bearbeitet <- c(bearbeitet, naechstes)

        #speichern des bearbeiteten items in der datenmatrix
        dat[a, naechstes] <- datall[a, naechstes]

        #ich kann erst bei 2 antworten die fähigkeitsparameter schätzen, bei einem bearbeiteten item wird dieser um 0,01
        #extremer als das schwerste/leichteste item angenommen
        if(length(bearbeitet) > 1){
          faehigkeit <- thetaEst(itemsparm[bearbeitet, ], dat[a, bearbeitet], method = "WL")
        }
        if(length(bearbeitet) == 1){
          if(mean(dat[a, ], na.rm = T) == 0) faehigkeit <- min(itemsparm[, 2] - .01)
          if(mean(dat[a, ], na.rm = T) == 1) faehigkeit <- max(itemsparm[, 2] + .01)

        }
      }
    }
  }


  #bis jedes item mindestens einmal pro "geschlecht" vorgegeben wurde, werden die nächsten items zufällig ausgewählt
  #gruppe m
  if(am*j/k <= 1 & row.names(dat)[a] == "m") {
    if(am == 1){
      datallm <- subset(datall, row.names(datall) == "m")
      zufallsvorgabe <- sample(datallm[am, ], j)
```

```
      dat[a, names(zufallsvorgabe)] <- zufallsvorgabe
    }
    if(am > 1){
      b <- am-1
      zufallsvorgabe <- sample(datallm[am, - bearbgesm[1:(b*j)]], j)
      dat[a, names(zufallsvorgabe)] <- zufallsvorgabe
    }
  }
  #gruppe w
  if(aw*j/k <= 1 & row.names(dat)[a] == "w") {
    if(aw == 1){
      datallw <- subset(datall, row.names(datall) == "w")
      zufallsvorgabe <- sample(datallw[aw, ], j)
      dat[a, names(zufallsvorgabe)] <- zufallsvorgabe
    }
    if(aw > 1){
      b <- aw-1
      zufallsvorgabe <- sample(datallw[aw, - bearbgesw[1:(b*j)]], j)
      dat[a, names(zufallsvorgabe)] <- zufallsvorgabe
    }
  }



  ################################################
  ###############verfahrensebene#################
  ################################################

  datm <- subset(dat, row.names(dat) == "m")
  datw <- subset(dat, row.names(dat) == "w")


  #die berechnung der itemparameter ergibt erst sinn, wenn jedes item öfter als einmal vorgegeben wurde und information
  #vorhanden ist
  #zuerst die frauen:
  if(aw*j/k > 1 & row.names(dat)[a] == "w"){
    #rasch.mml2 kann nicht problemlos ausgeführt werden, wenn im datensatz items ohne information vorhanden sind
    #weshalb die entsprechenden items (und ebenso Testpersonen) ausgeschlossen werden
    if(datcheck(datw)){
      datneuw <- datw
      while(datcheck(datneuw)) datneuw <- datimp(datneuw)
```

```
572         }
573         if(exists("datneuw")){
574           if(length(datneuw) <= 4) rm(datneuw)
575         }
576         #selbst bei bereinigung des datensatzes um items ohne information kommt es immer wieder zu warnmeldungen,
577         #da rasch.mml2 eine korrelation berechnet und besonders bei kleinen stichproben gelegentlich die standardabweichung 0 ist.
578         #dies hat allerdings keinen erkennbaren einfluss auf die interessierenden ergebnisse: die itemparameter können trotzdem
579         #berechnet werden.
580         #warnmeldungen werden deshalb deaktiviert.
581         if(a < 150) options(warn = -1)
582         ifelse(exists("datneuw"), raschmodellw <- try(rasch.mml2(datneuw, progress = F), silent = T),
583                raschmodellw <- try(rasch.mml2(datw, progress = F), silent = T))
584         options(warn = 0)
585         try(schwierigkeitw <- raschmodellw$item[, "b"], silent = T)
586         try(names(schwierigkeitw) <- raschmodellw$item[, "item"], silent = T)
587         if(exists("datneuw")) rm("datneuw")
588       }
589
590       #nun die männer
591       if(am*j/k > 1 & row.names(dat)[a] == "m"){
592         #rasch.mml2 kann nicht ausgeführt werden, wenn im datensatz items ohne information vorhanden sind
593         if(datcheck(datm)) {
594           datneum <- datm
595           while(datcheck(datneum)) datneum <- datimp(datneum)
596         }
597         if(exists("datneum")){
598           if(length(datneum) <= 4) rm(datneum)
599         }
600         #selbst bei bereinigung des datensatzes um items ohne information, kommt es immer wieder zu warnmeldungen,
601         #da rasch.mml2 eine korrelation berechnet und besonders bei kleinen stichproben gelegentlich die standardabweichung 0 ist.
602         #dies hat allerdings keinen erkennbaren einfluss auf die interessierenden ergebnisse: die itemparameter können berechnet
603         #werden.
604         #warnmeldungen werden während der anwendung der funktion rasch.mml2 deshalb deaktiviert.
605         if(a < 150) options(warn = -1)
606         ifelse(exists("datneum"), raschmodellm <- try(rasch.mml2(datneum, progress = F), silent = T),
607                raschmodellm <- try(rasch.mml2(datm, progress = F), silent = T))
608         options(warn = 0)
609         try(schwierigkeitm <- raschmodellm$item[, "b"], silent = T)
610         try(names(schwierigkeitm) <- raschmodellm$item[, "item"], silent = T)
611         if(exists("datneum")) rm("datneum")
612       }
```

```
613
614       #ich erhebe welche itemparameter noch nicht berechnet werden konnten
615       #zuerst die frauen:
616       if(exists("schwierigkeitw") & row.names(dat)[a] == "w"){
617         if(exists("problemitemsw")) rm("problemitemsw")
618         if(exists("probschwerw")) rm("probschwerw")
619         if(exists("probleichtw")) rm("probleichtw")
620         if(length(schwierigkeitw) != k){
621           problemitemsw <- which(is.element(1:k, as.numeric(names(schwierigkeitw))) == F)
622           #ich überprüfe, ob die problemitems eher schwer oder leicht sein dürften - items die weniger als 50% gelöst wurden
623           #sind eher schwer
624           if(length(problemitemsw) > 1){
625             probmittelw <- colMeans(dat[, problemitemsw], na.rm = T)
626             probschwerw <- as.numeric(names(which(probmittelw < .5)))
627             if(length(probschwerw) == 0) rm(probschwerw)
628             probleichtw <- as.numeric(names(which(probmittelw >= .5)))
629             if(length(probleichtw) == 0) rm(probleichtw)
630           }
631           if(length(problemitemsw) == 1){
632             if(mean(dat[, problemitemsw], na.rm = T) >= .5) probleichtw <- problemitemsw
633             if(mean(dat[, problemitemsw], na.rm = T) < .5) probschwerw <- problemitemsw
634           }
635
636         }
637       }
638
639       #nun die männer:
640       if(exists("schwierigkeitm") & row.names(dat)[a] == "m"){
641         if(exists("problemitemsm")) rm("problemitemsm")
642         if(exists("probschwerm")) rm("probschwerm")
643         if(exists("probleichtm")) rm("probleichtm")
644         if(length(schwierigkeitm) != k){
645           problemitemsm <- which(is.element(1:k, as.numeric(names(schwierigkeitm))) == F)
646           #ich überprüfe, ob die problemitems eher schwer oder leicht sein dürften - items die weniger als 50% gelöst wurden
647           #sind eher schwer
648           if(length(problemitemsm) > 1){
649             probmittelm <- colMeans(dat[, problemitemsm], na.rm = T)
650             probschwerm <- as.numeric(names(which(probmittelm < .5)))
651             if(length(probschwerm) == 0) rm(probschwerm)
652             probleichtm <- as.numeric(names(which(probmittelm >= .5)))
653             if(length(probleichtm) == 0) rm(probleichtm)
```

```
      }
      if(length(problemitemsm) == 1){
        if(mean(dat[, problemitemsm], na.rm = T) >= .5) probleichtm <- problemitemsm
        if(mean(dat[, problemitemsm], na.rm = T) < .5) probschwerm <- problemitemsm
      }
    }
  }

  ######################Ergebnisse und Einstellungen für die nächste VP
  #zuerst die männer
  if(exists("raschmodellm")){
    if(class(raschmodellm) != "try-error"){
      #holt sich die schwierigkeitsparameter aus den ergebnissen des raschmodells und fügt sie in "itemspar" ein
      if(exists("schwierigkeitm")){
        itemsparm[, 2] <- 0
        itemsparm[names(schwierigkeitm), 2] <- schwierigkeitm #die schwierigkeitsparameter werden übernommen
        itemsparm <- minimaxi(datm[1:am, ], itemsparm, schwierigkeitm) #items ohne informationen bekommen werte zugeordnet

        #ein output der lediglich den aktuellen stand der kalibrierung wiedergeben soll
        print("gelöstm")
        print(length(schwierigkeitm))
        print("durchschnittliches residuumm:")
        print(mean(sqrt((schwierigkeitm - trueitempar[names(schwierigkeitm)])^2)))


        #es wird der standardfehler der schwierigkeitsparameter erhoben
        try(standarderrorm <- raschmodellm$se.b, silent = T)
        names(standarderrorm) <- names(schwierigkeitm)

        #es sollen jene werte des konfidenzintervalls für die weitere vorgabe verwendet werden, die näher an null liegen
        #nur relevant, wenn z > 0
        konfidenzintervallm <- matrix(NA, ncol = 7, nrow = k)
        colnames(konfidenzintervallm) <- c("itemalt", "standarderror", "+ wert", "- wert", "itemneu", "KI+", "KI-")
        rownames(konfidenzintervallm) <- 1:k
        konfidenzintervallm[, 1] <- itemsparm[, 2]
        konfidenzintervallm[names(standarderrorm), 2] <- standarderrorm
        konfidenzintervallm[which(is.element(konfidenzintervallm[, 2], NA)), 2] <- max(standarderrorm) + .1
        konfidenzintervallm[, 3] <- konfidenzintervallm[, 1] + z * konfidenzintervallm[, 2]
        konfidenzintervallm[, 4] <- konfidenzintervallm[, 1] - z * konfidenzintervallm[, 2]
        konfidenzintervallm[, 6] <- konfidenzintervallm[, 1] + 1.96 * konfidenzintervallm[, 2]
        konfidenzintervallm[, 7] <- konfidenzintervallm[, 1] - 1.96 * konfidenzintervallm[, 2]
```

```
695
696           vgm <- konfidenzintervallm[which(konfidenzintervallm[, 1] < 0), 3]
697           konfidenzintervallm[names(vgm[which(vgm < 0)]), 5] <- vgm[which(vgm < 0)]
698           konfidenzintervallm[names(vgm[which(vgm >= 0)]), 5] <- 0
699
700           vkm<- konfidenzintervallm[which(konfidenzintervallm[, 1] > 0), 4]
701           konfidenzintervallm[names(vkm[which(vkm > 0)]), 5] <- vkm[which(vkm > 0)]
702           konfidenzintervallm[names(vkm[which(vkm <= 0)]), 5] <- 0
703
704           konfidenzintervallm[which(konfidenzintervallm[, 1] == 0), 5] <- 0
705
706
707           itemsparm[, 2] <- konfidenzintervallm[, 5]
708
709           print("mittlerer standardfehlerm")
710           print(mean(konfidenzintervallm[, 2]))
711
712           #die grafik der m-gruppe
713           if(exists("grafsem") == F) grafsem <- c(a, mean(konfidenzintervallm[, 2]))
714           grafsem <- rbind(grafsem, c(a, mean(konfidenzintervallm[, 2])))
715
716         }
717       }
718     }
719
720     #nun die frauen
721     if(exists("raschmodellw")){
722       if(class(raschmodellw) != "try-error"){
723         #holt sich die schwierigkeitsparameter aus dem raschmodell und fügt sie in "itemspar" ein
724         if(exists("schwierigkeitw")){
725           itemsparw[, 2] <- 0
726           itemsparw[names(schwierigkeitw), 2] <- schwierigkeitw #die schwierigkeitsparameter werden übernommen
727           itemsparw <- minimaxi(datw[1:aw, ], itemsparw, schwierigkeitw) #items ohne informationen bekommen werte zugeordnet
728
729           #ein output der lediglich den aktuellen stand der kalibrierung wiedergeben soll
730           print("gelöstw")
731           print(length(schwierigkeitw))
732           print("durchschnittliches residuumw:")
733           print(mean(sqrt((schwierigkeitw - trueitempar[names(schwierigkeitw)])^2)))
734
735
```

```
#es wird der standardfehler der schwierigkeitsparameter erhoben

try(standarderrorw <- raschmodellw$se.b, silent = T)
names(standarderrorw) <- names(schwierigkeitw)

#es sollen jene werte des konfidenzintervalls für die weitere vorgabe verwendet werden, die näher an null liegen
konfidenzintervallw <- matrix(NA, ncol = 7, nrow = k)
colnames(konfidenzintervallw) <- c("itemalt", "standarderror", "+ wert", "- wert", "itemneu", "KI+", "KI-")
rownames(konfidenzintervallw) <- 1:k
konfidenzintervallw[, 1] <- itemsparw[, 2]
konfidenzintervallw[names(standarderrorw), 2] <- standarderrorw
konfidenzintervallw[which(is.element(konfidenzintervallw[, 2], NA)), 2] <- max(standarderrorw) + .1
konfidenzintervallw[, 3] <- konfidenzintervallw[, 1] + z * konfidenzintervallw[, 2]
konfidenzintervallw[, 4] <- konfidenzintervallw[, 1] - z * konfidenzintervallw[, 2]
konfidenzintervallw[, 6] <- konfidenzintervallw[, 1] + 1.96 * konfidenzintervallw[, 2]
konfidenzintervallw[, 7] <- konfidenzintervallw[, 1] - 1.96 * konfidenzintervallw[, 2]

vgw <- konfidenzintervallw[which(konfidenzintervallw[, 1] < 0), 3]
konfidenzintervallw[names(vgw[which(vgw < 0)]), 5] <- vgw[which(vgw < 0)]
konfidenzintervallw[names(vgw[which(vgw >= 0)]), 5] <- 0

vkw<- konfidenzintervallw[which(konfidenzintervallw[, 1] > 0), 4]
konfidenzintervallw[names(vkw[which(vkw > 0)]), 5] <- vkw[which(vkw > 0)]
konfidenzintervallw[names(vkw[which(vkw <= 0)]), 5] <- 0

konfidenzintervallw[which(konfidenzintervallw[, 1] == 0), 5] <- 0

itemsparw[, 2] <- konfidenzintervallw[, 5]

print("mittlerer standardfehlerw")
print(mean(konfidenzintervallw[, 2]))

#ein output der lediglich den aktuellen stand der kalibrierung wiedergeben soll
print("ausschluss")
print(ausschluss)

#die grafik der w gruppe
if(exists("grafsew") == F) grafsew <- c(a, mean(konfidenzintervallw[, 2]))
grafsew <- rbind(grafsew, c(a, mean(konfidenzintervallw[, 2])))
plot(grafsew[, 1:2], type = "l", col = "red")
if(exists("grafsem")) lines(grafsem[, 1:2], col = "blue")
```

```
    }
  }
}

#prüfung auf DIF (Fischer und Scheiblechner z-Test)
if(exists("schwierigkeitw") & exists("schwierigkeitm") & a > m){
  ztest <- abs((konfidenzintervallm[, 1] - konfidenzintervallw[, 1])/sqrt(konfidenzintervallw[, 2]^2 +
                                                                          konfidenzintervallm[, 2]^2))
  ausschluss <- which(ztest > 2.58)
  if(a < 300) ausschluss <- which(ztest > 3.5)
  if(a < 500 & a >= 300) ausschluss <- which(ztest > 3)
  if(a >= 500) ausschluss <- which(ztest > 2.58)
  if(length(ausschluss) > 0) if(is.character(ausschluss)) ausschluss <- c()
}



#####################################################################
#vorgabe des tests an eine große anzahl an Testpersonen:
#####################################################################
if(a == 100 | a == 250 | a == 500) {

  adaptvor <- function(zsim){
    bearb <- c()
    nichtvor <- c(ausschluss, itemnummer[-as.numeric(rownames(zwischenitem))])
    for(i in 1:j){
      if(i > 2){
        faeh <- thetaEst(itemspar[bearb, ],
                         zsim[bearb], range = c(-7, 7), method = "WL")
      }
      if(i == 1) faeh = 0
      if(i == 2){
        ifelse(sum(x[bearb]) == 1, faeh <- max(itemspar[, 2]), faeh <- min(itemspar[, 2]))
      }
      bearb <- c(bearb, nextItem(catitems, faeh, out = c(bearb, nichtvor), method = "WL")$item)
    }
    faeh <- c(faeh, bearb)
    return(faeh)
  }

  ###########################################################################
```

```
#gruppe m
#############################################################################
if(datcheck(dat[1:a, ])) {
  datneu <- dat[1:a, ]
  while(datcheck(datneu)) datneu <- datimp(datneu)
}
if(exists("datneu")){
  if(length(datneu) <= 4) rm(datneu)
}
ifelse(exists("datneu"), zwischenergebnis <- rasch.mml2(datneu, progress = F),
       zwischenergebnis <- rasch.mml2(dat[1:a, ], progress = F))
if(exists("datneu")) rm("datneu")

itemnummer <- 1:k

zwischenitem <- zwischenergebnis$item

itemspar <- matrix(0, nrow = k, ncol = 4)
itemspar[, 1] <- 1
itemspar[, 4] <- 1
itemspar[as.numeric(rownames(zwischenitem)), 2] <- zwischenitem[, 4]
itemspar <- minimaxi(dat[1:a, ], itemspar)

catitems <- createItemBank(itemspar, model ="1PL", cb = F, thMin=-10, thMax=10)

truepersonm <- sort(rnorm(2500))
datzwischenallm <- sim.raschtype(truepersonm, trueitempar)

#############################################################################

schaetzp1 <- t(apply(datzwischenallm, 1, adaptvor))
erg1 <- mean(abs(truepersonm - schaetzp1[, 1]))
cor1 <- cor(truepersonm, schaetzp1[, 1])


#############################################################################
#gruppe w
#############################################################################

truepersonw <- sort(rnorm(2500))
datzwischenallw <- sim.raschtype(truepersonw, trueitempardif)
```

```
859
860         ##############################################################################
861
862         schaetzp2 <- t(apply(datzwischenallw, 1, adaptvor))
863         erg2 <- mean(abs(truepersonw - schaetzp2[, 1]))
864
865         plot(itemspar[, 2])
866
867         itemspar[itemnummer[-as.numeric(rownames(zwischenitem))], 2] <- NA
868
869         ##############################################################################
870         #zufall
871         ##############################################################################
872         #für die zufallsvorgabe muss festgestellt werden bei welchen items DIF vorliegt, die ausschlüsse für die
873         #adaptive vorgabe werden unter anderem objektnamen gespeichert, damit kein konflikt mit der
874         #itemkalibrierung entsteht
875         ausschlussorig <- ausschluss
876
877         #Items mit DIF in der zufallsvorgabe werden erhoben
878         if(a >= m){
879           ausschluss <- rmcompdif(datrand[1:a, ])
880           if(length(ausschluss) > 0) {
881             if(is.character(ausschluss)) ausschluss <- c()
882           }
883         }
884
885         if(datcheck(datrand[1:a, ])) {
886           datneu <- datrand[1:a, ]
887           while(datcheck(datneu)) datneu <- datimp(datneu)
888         }
889         if(exists("datneu")){
890           if(length(datneu) <= 4) rm(datneu)
891         }
892         ifelse(exists("datneu"), zwischenergebnis <- rasch.mml2(datneu, progress = F),
893                zwischenergebnis <- rasch.mml2(datrand[1:a, ], progress = F))
894         if(exists("datneu")) rm("datneu")
895
896         zwischenitem <- zwischenergebnis$item
897
898         itemsparz <- matrix(0, nrow = k, ncol = 4)
899         itemsparz[, 1] <- 1
```

```
        itemsparz[, 4] <- 1
        itemsparz[as.numeric(rownames(zwischenitem)), 2] <- zwischenitem[, 4]
        itemsparz <- minimaxi(datrand[1:a, ], itemsparz)
        itemsparalt <- itemspar
        itemspar <- itemsparz

        catitems <- createItemBank(itemsparz, model ="1PL", cb = F, thMin=-10, thMax=10)

        #gruppe m
        schaetzp3 <- t(apply(datzwischenallm, 1, adaptvor))
        erg3 <- mean(abs(truepersonm - schaetzp3[, 1]))
        cor3 <- cor(truepersonm, schaetzp3[, 1])

        #gruppe w
        schaetzp4 <- t(apply(datzwischenallw, 1, adaptvor))
        erg4 <- mean(abs(truepersonw - schaetzp4[, 1]))

        points(itemsparz[, 2], col = "red")

        itemsparz[itemnummer[-as.numeric(rownames(zwischenitem))], 2] <- NA
        itemspar <- itemsparalt

        ##########speichern

        datoutsav(t(schaetzp1[,1]), paste("personmseq", a, ".csv", sep = ""))
        datoutsav(t(schaetzp2[,1]), paste("personwseq", a, ".csv", sep = ""))
        datoutsav(t(schaetzp3[,1]), paste("personmz", a, ".csv", sep = ""))
        datoutsav(t(schaetzp4[,1]), paste("personwz", a, ".csv", sep = ""))
        datoutsav(t(truepersonm), paste("truepersonm", a, ".csv", sep = ""))
        datoutsav(t(truepersonw), paste("truepersonw", a, ".csv", sep = ""))
        datoutsav(t(itemsparz[,2]), paste("itemsparz", a, ".csv", sep = ""))
        datoutsav(t(itemspar[,2]), paste("itemspar", a, ".csv", sep = ""))

        #löschen der datensätze für die simulation der personenparameter
        rm("datzwischenallw", "datzwischenallm", "truepersonw", "truepersonm", "schaetzp1", "schaetzp2", "schaetzp3",
           "schaetzp4", "itemspar", "itemsparz")

        #die ausschlussitems für die adaptive vorgabe werden gespeichert und wiederhergestellt
        datsav(ausschluss, paste("ausschlussz", a, ".rds", sep = ""))
        ausschluss <- ausschlussorig
        datsav(ausschluss, paste("ausschluss", a, ".rds", sep = ""))
```

```
  }

}

#########################################################################
############################datenspeicherung#############################
#########################################################################

datsav(datall)

datsav(dat)

datsav(datrand)

datsav(datallm)

datsav(datallw)

datsav(konfidenzintervallm)

datsav(konfidenzintervallw)

#der datensatz wird mit einer unterschiedlichen anzahl an vp gespeichert
datout100 <- rmoutput(dat[1:100, ], datrand[1:100, ], trueitempar)
datoutsav(datout100)

datout250 <- rmoutput(dat[1:250, ], datrand[1:250, ], trueitempar)
datoutsav(datout250)

datout500 <- rmoutput(dat[1:500, ], datrand[1:500, ], trueitempar)
datoutsav(datout500)

#########################################################################
############################datenlöschung################################
#########################################################################
if(r > 1){
  behalten <- c("a", "trueitempardif", "trueitempar", "passung", "datcheck", "datimp", "datsav", "minimaxi",
                "zufallsvor", "j", "k", "m", "n", "spitze", "namen", "z", "r", "rmoutput", "rmcompdif", "rmcomp",
                "datoutsav", "anfang", "difvor", "probab", "difvor", "passungvoll")
  del <- ls()[-which(ls() %in% behalten)]
  rm(list = del)
```

```
982         rm(del)
983       }
984     }
```

IV. LEBENSLAUF

Ausbildung und berufliche Tätigkeiten:

Seit Sep. 2015	Mitarbeit an dem Projekt „Thrombektomie bei ischämischen Schlaganfall“, Auftraggeber Ludwig Boltzmann Institut, Health Technology Assessment
Okt. 2014 - Dez. 2014	Mitarbeit an der Studie „Health effects of cow’s milk consumption in infants up to 3 years of age – a systematic review and meta-analysis”, Auftraggeber Donau-Universität-Krems (DUK)
Feb. 2014 - Dez. 2014	Mitarbeit an dem Projekt „Interventionen zur Adipositasprävention bei Kindern und Jugendlichen“, Auftraggeber DUK
Feb. 2013 - Sep. 2013	Mitarbeit an dem Projekt „Jugendlichenuntersuchung Neu“, Auftraggeber DUK
Juli u. August 2012	Mitarbeit an der Evaluationsstudie „Gesunde Schule is(s)t“, Auftraggeber DUK
Nov. 2011 - Feb. 2012	Praktikum an der DUK, Department für Evidenzbasierte Medizin und klinische Epidemilogie, Mitarbeit an dem Projekt „Ausgewählte Indikatoren zur Beschreibung der gesundheitlichen Lage von Kindern und Jugendlichen“
April 2009 - Nov. 2010	Freiberufliche Tätigkeit als Dipl. Gesundheits- u. Krankenpfleger in der Hauskrankenpflege
seit Oktober 2008	Studium der Psychologie an der Universität Wien, 1010 Wien. Studienschwerpunkte: Forschungsmethoden, Evaluation und psychoglische Diagnostik.
Juni 2008	Ablegung der Berufsreifeprüfung
Sept. 2007 - Juni 2008	Vorbereitung zur Berufsreifeprüfung am WIFI, 5020 Salzburg

März 2007 - Aug. 2007	Arbeit als Dipl. Gesundheits- u. Krankenpfleger in der Seniorenwohnanlage der Österreichischen Jungarbeiterbewegung, 5026 Salzburg
2006 - 2007	Zivildienst bei Pro Juventute (Kinderbetreuung), 5020 Salzburg
2003 - 2006	Ausbildung zum Dipl. Gesundheits- u. Krankenpfleger Sozialmedizinischem Zentrum Süd, 1100 Wien
2002 - 2003	Freiwilliges Soziales Jahr im Pflegeheim Ried, 4910 Ried im Innkreis

Publikationen:

Griebler U., Bruckmüller M. U., Kien C., Dieminger B., Meidlinger B., Seper K., Hitthaler A., Emprechtinger R., Wolf A., Gartlehner G., (2015) *Health effects of cow's milk consumption in infants up to 3 years of age: a systematic review and meta-analysis.* Public Health Nutrition, available on CJO2015. doi:10.1017/S1368980015001354.

Kien C., Emprechtinger R., Grillich L., (2014) *Primär- und Sekundärprävention von Übergewicht und Adipositas. Overview of Reviews.* Department für Evidenzbasierte Medizin und Klinische Epidemiologie, Donau-Universität-Krems

Kien C., Emprechtinger R., Strobelberger M., (2013) *Evaluation „Gesunde Schule is(s)t“,* Department für Evidenzbasierte Medizin und Klinische Epidemiologie, Donau-Universität-Krems

Kien C., Kaminski-Hartenthaler A., Emprechtinger R., Rohleder S., Mahlknecht P., (2013) *Evidenzreport „Jugendlichenuntersuchung NEU“,* Department für Evidenzbasierte Medizin und Klinische Epidemiologie, Donau-Universität-Krems