Introduction to the Rasch Poisson Counts Model: An R Tutorial

Article *in* Psychological Reports · September 2018

DOI: 10.1177/0033294118797577

2 authors:

Purya Baghaei
Islamic Azad University Mashhad Branch
97 PUBLICATIONS **1,582** CITATIONS
SEE PROFILE

Philipp Doebler
Technische Universität Dortmund
104 PUBLICATIONS **2,246** CITATIONS
SEE PROFILE

Article

Psychological Reports
0(0) 1–28

DOI: 10.1177/0033294118797577
journals.sagepub.com/home/prx
SAGE

Introduction to the Rasch Poisson Counts Model: An R Tutorial

Purya Baghaei
English Department, Mashhad Branch, Islamic Azad University, Mashhad, Iran

Philipp Doebler
Department of Statistics, TU Dortmund University, Dortmund, Germany

Abstract
The Rasch Poisson Counts Model is the oldest Rasch model developed by the Danish mathematician Georg Rasch in 1952. Nevertheless, the model has had limited applications in psychoeducational assessment. With the rise of neurocognitive and psychomotor testing, there is more room for new applications of the model where other item response theory models cannot be applied. In this paper, we give a general introduction to the Rasch Poisson Counts Model and then using data of an attention test walk the reader through how to use the "lme4" package in R to estimate the model and interpret the outputs.

Keywords
Rasch Poisson Counts Model, psychomotor testing, attention, "lme4" package

Introduction

In most item response theory (IRT) models the unit of analysis is the individual item. In such models the probability that a person correctly answers an item or endorses certain categories is modeled. However, common IRT models need at least one parameter per item (any many more on polytomous IRT models), so they are relatively complex for situations where the same task or many simple tasks are given to examinees and aggregation of hits/misses is conducted. Such testing conditions arise in psychomotor testing (Spray, 1990), the testing of attention/processing

speed (Baghaei, Ravand, & Nadri, in press; Doebler & Holling, 2015), oral reading errors (Jansen, 1997b; Rasch, 1960; Verhelst & Kamphuis, 2009), reading comprehension (Verhelst, & Kamphuis, 2009), and divergent thinking (Forthmann, Gerwig, Holling, Çelik, Storme, & Lubart, 2016). In these tests, examinees usually have to solve an unlimited (or at least very large) number of relatively easy items within a fixed period of time. Another example is identifying correctly spelled words in a long list of words. In such testing situations the total scores (raw counts) or the total numbers of errors on the tasks are modeled instead of the individual attempts.

The Rasch Poisson Counts Model (RPCM, Rasch, 1960/1980) is a member of the family of Rasch models which was developed for tests where counts of errors or successes on a number of tasks are modeled instead of replies to individual items. Modeling the number of errors might be the only option when the number of potential successes is not well defined, say in classic oral reading tests where examinees are to read a passage aloud and the test administrator counts the number of errors (Rasch, 1960). In such conditions the total scores or the total number of errors on each block is assumed to be the realization of a Poisson process (e.g. Ross, 1983). That is, the number of correct checks (or errors) on each block for each person is assumed to be Poisson distributed and is the unit of analysis.[1]

RPCM is a unidimensional latent trait model (Jansen, 1994) and enjoys all the elegant properties of Rasch models including separate person and item parameters, sufficiency of raw scores, and specifically objective comparison of persons and items (Masters & Wright, 1984). Masters and Wright (1984) demonstrate that the RPCM, binomial trials, rating scale model, and

1 Andrich's (1978) Rating Scale Model or Master's (1982) Partial Credit Model could be considered alternatives for this situation, but (a) they have a lot more parameters and (b) hence require larger sample sizes than the RPCM for stable parameter estimation. Also, (c) the maximal number of errors would have to be known.

the partial credit model all originate from a logistic function and hence have the same algebraic basis. In all these models simple counts of correct replies or successes are the sufficient statistics to estimate ability and difficulty parameters.

The Poisson distribution

To appreciate the RPCM understanding the mathematical basis of the model is in order. The model rests on the Poisson probability function and its underlying assumptions. The Poisson probability function, named after the French mathematician Siméon Denis Poisson (1781-1840), expresses the probability of a number of events occurring in a fixed time period if the average number of events is known. The function is written as follows:

$$P(k) = \frac{\lambda^k e^{-\lambda}}{k!} \qquad (1)$$

where λ is the expectation or the average of events within a fixed interval of time, and k is the observed number of events. Note also, that the parameter λ coincides with the variance of a Poisson variable.

The Poisson probability function is adequate in the following situation: The number of many potential events is recorded, each event has a very small probability of occurring and the number of potential events might be unknown. In addition, the occurrence of one event does not have any impact on the probability of another event (stochastic independence). Suppose the average number of oral reading errors is 2.5 in a session (say this value has been computed using the past records of reading error frequency). Now we can compute the probability of making no errors (k=0) or 1, 2, 3 … errors in a given reading session. Using the Poisson probability function we have:

$$P(k = 0) = \frac{2.5^0 e^{-2.5}}{0!} = \frac{e^{-2.5}}{1} = 0.082$$

$$P(k=1) = \frac{2.5^1 e^{-2.5}}{1!} = \frac{2.5 e^{-2.5}}{1} = 0.205$$

$$P(k=2) = \frac{2.5^2 e^{-2.5}}{2!} = \frac{6.25 e^{-2.5}}{2} = 0.257$$

That is, the probability of making no errors in a session is .08, the probability of one error is .20, and the probability of making two is .25.

Rasch Poisson Counts Model

The RPCM is historically the first member of the family of Rasch models developed by the Danish mathematician and statistician Georg Rasch. Despite being the first Rasch model introduced it has had limited applications compared with other extensions of the model such as the dichotomous model or the polytomous models for rating data. In 1951 the Danish Ministry of Social Affairs assigned Rasch to the task of analyzing oral reading data collected over several years from a group of 125 students with reading problems. The complication that Rasch encountered was that the texts students had read in each testing administration differed in difficulty, making the monitoring of their reading development rather difficult.

> In a concrete formulation of this problem I imagined - in good statistical tradition - the possibility that the reading ability of a student at each stage, and in each of the two above-mentioned dimensions [accuracy and speed], could be characterized in a quantitative way - not through a more or less arbitrary grading scale, but by a positive real number defined as regularly as the measurement of a length (Rasch, 1977).

Preliminary analyses of the reading errors revealed that the numbers of reading errors students committed across different texts were “proportionate to each other, although with a wide margin for variations” (Rasch, 1977). To account for these variations he used a Poisson probability

function to model the oral misreadings. Rasch assumed that the distribution of reading errors across different texts (items) was independent. That is, it should be possible to estimate the probability of the number of errors made by each test taker with the Poisson function. Rasch's reading data fitted very well to the Poisson distribution leading to the introduction of the multiplicative Poisson model (Rasch, 1960/1980). "The outcome of the reading test experiment was beyond expectation: a statistically very satisfactory analysis on the basis of a new model which represented a genuine innovation in statistical techniques!" (Rasch, 1977). The development of the most well-known and widely used Rasch model, i.e., the dichotomous model, was based upon the multiplicative Poisson model.

Rasch (1960) used the Poisson probability function to model raw counts of errors on the reading tests. Here the total numbers of misreadings on individual tests were modeled. The RPCM assumes that the total raw scores or errors Y_{vi} of person v on part i *of a test* are independent and Poisson distributed with mean μ_{vi}. Note that in part of the RPCM literature a collection of simple items or tasks, i.e. a subtest, is referred to as an item and the total scores on these subtests are modeled and not responses to individual items. Item parameters refer to the subtests. Time limits on the subtests are optional (Doebler, Doebler, & Holling, 2014).

The parameter μ_{vi} is the expected number of successes of person v on item i. The probability that person v gets a raw scores of y on item i is given by the Poisson function which is the multiplicative form of the model:

$$P(Y_{vi} = y_{vi}) = \frac{exp(-\mu_{vi})\mu_{vi}^{y_{vi}}}{y_{vi}!} \qquad (2)$$

The probability to observe a score (or number of errors) y_{vi} in Equation 2 is obtained by replacing k in Equation 1 by y_{vi} and λ by μ_{vi}. The expectation μ_{vi} is a function of person's ability

and item's difficulty and is assumed to have a multiplicative composition, i.e., it is the product of person's ability and item's easiness:

$$\mu_{vi} = \theta_v \sigma_i \qquad (3)$$

where θ_v, and σ_i are person ability and item easiness parameters, respectively. Extending the model by a time limit is possible, i.e., $\mu_{vi} = \tau_i \theta_v \sigma_i$, where τ_i is the time limit for item i, which can be set to one if there is no time limit (e.g. Doebler, et al. 2014) . In some estimation approaches the following constraint is imposed for model identification (Jansen & van Dujn, 1992):

$$\sum_i \tau_i \sigma_i = 1 \qquad (4)$$

but constraints on the distribution of θ_v can also be used, as we will elaborate below.

Rasch (1960/1980) with some algebra demonstrated that it is possible to estimate the person parameter θ_v independently of the item parameter σ_i and vice versa (separability of parameters). Equivalent to the multiplicative form of the RPCM is the additive specification that uses the natural logarithm as a link function. Towards this, apply the log-function to Equation 3 and obtain:

$$\tilde{\mu}_{vi} = log(\mu_{vi}) = log(\theta_v \sigma_i) = log(\theta_v) + log(\sigma_i) = \tilde{\theta}_v + \tilde{\sigma}_i \qquad (5)$$

We use the tilde symbol to indicate that log-parameters are used. Note that we can always obtain the original form in Equation 3 by applying the exponential function, i.e., $exp(\tilde{\theta}_v) = \theta_v$ and $exp(\tilde{\sigma}_i) = \sigma_i$.

The additive specification is important because it allows to view the RPCM as a special case of a generalized linear mixed model, a large class of regression models, and to employ corresponding software.

Local independence and unidimensionality are assumed to hold for the RPCM like in other IRT models. That is, a test is supposed to measure a single trait and the trials should be independent of each other conditional on a fixed level of ability as are the scores for different examinees. This assumption may be violated as learning and fatigue can affect attempts, especially those close to each other in time (Spray, 1990).

There are several estimation methods for the RPCM (Kampuis & Verhelst, 2009). As there are sufficient statistics for parameter estimation, conditional maximum likelihood estimation (CML) is feasible. However, no distribution of the person parameters is obtained in CML, which might be interesting especially when structural models for latent traits are of interest (Jansen, 2003). Treating the ability parameter as a Gamma distributed random variable one can derive marginal maximum likelihood estimators for the person and item parameters (Jansen, 1997a).

Joint maximum likelihood estimation (JML) also exists to estimate the parameters of RPCM. However, formally each additional person adds a parameter to the model, which is statistically undesirable. In contrast, marginal maximum likelihood estimation (MML) imposes some distributional assumptions on the person parameters and as a consequence, the model's likelihood is a function of the item parameters only (and maybe the parameters of the marginal distribution; Doebler & Holling, 2015). For MML estimation the two-parameter Gamma distribution is typically specified for the person parameter with shape parameter c and scale parameter m. This is referred to as Gamma Poisson Counts Model (Jansen & van Duijn, 1992). Alternatively, a lognormal distribution is computationally feasible (e.g. Doebler & Holling, 2015) but analytically less tractable (Jansen, 1994). Since person parameters are non-negative the Gamma distribution which is conjugate to the Poisson is very convenient from an algebraic perspective. Jansen and van Duijn (1992) demonstrated that imposing a Gamma distribution on

the person parameters does not affect the point estimates of the item parameters, i.e. JML, MML with Gamma marginal distribution, and CML result in the same item parameter estimates. The drawback of imposing a parametric distribution, however, is that the ability distribution might be misspecified (Jansen, 1994) which, nevertheless, does not affect point estimates of item parameters. Therefore, if only the difficulty of the tests is of interest the distribution of the abilities can essentially be ignored. However, in other applications the person parameters and their distribution are important (Jansen, 1994, 1997a).

The fit of the model is assessed by checking the ability of the model to correctly predict total scores, as they are sufficient statistic for estimating model parameters. As the RPCM is a log-linear model (cf. Equation 3) it can be considered as a generalized linear model (GLM; McCullagh & Nelder, 1989) or (from the MML perspective) as a generalized linear mixed model (GLMM; Demidenko, 2013; Brown and Prescott, 2015). Therefore, methods for checking GL(M)Ms and the diagnosis of under or overdispersion is available for RPCM too (Doebler & Holling, 2015). Assuming a Gamma distribution for the person parameters implies that the distribution of the row total scores is negative binomial (Jansen & van Duijn, 1992). This property can be used to evaluate the correctness of the Gamma assumptions (Jansen & van Duijn, 1992). Model checking within MML estimation entails comparing the observed distribution of total scores with the one predicted by the model.

Empirical Example

In this section RPCM is estimated using the lme4 package (Bates, et al., 2017) in R (R Core Team, 2016). We employ the glmer function from the lme4 package. The additive form of the model from Equation 5 has to be used as glmer does not support the equivalent multiplicative form

directly. Also, glmer can only handle a lognormal distribution of the person parameter, i.e. θ_v from Equation 5 is assumed to follow a normal distribution and the parameters of this normal distribution are estimated, too.

The data of 228 examinees to a selective attention test is used for demonstration. The test is constructed by Beyzaee (2017) after the model of Ruff 2 and 7 Selective Attention Test (Ruff, Evans, & Light, 1986). In this test respondents have to cross out the digits 2 and 7 in three rows of randomly arranged digits and letters. The test contains 20 blocks each containing three lines. The time limit for each block is 15 seconds. Here the task is very simple and it is not reasonable to scale the test with the dichotomous Rasch model. An example block is given below.

2 G O X C 7 M J 7 H Z R N G A S 2 Y W Q 2 L H B Z G J N V 7 E T 2 P R V M J H S T Q 2 C 7 K L W C 7
X M T 7 K T R 2 A V P I W O C 2 G J 7 L S 2 B N V W 7 T O X R 2 P H 7 F D A B M 2 W H K A S T 2 O P
H W E D 2 T R N E Q X 2 P K L 7 P K 7 Z C V 7 2 Z 7 E T G H L K S D I N 7 S 2 W I S N 7 T B M O P W

Estimation with 'lme4'

A prerequisite for RPCM analysis with lme4 is that the data should be in long format. Perhaps the simplest way to do this is with SPSS though also R includes this functionality in the reshape function. The data structure for the current analysis after converting it to long format is shown in Figure 1. 'ID' is student identification number, 'Grade' indicates each student's grade at school, 'Item' refers to the block of letters and numbers students had to check, 'Hit' is the total number of correct checks on each item for each examinee, 'Miss' is the number of items that test takers have failed to check, and 'TL' is the hypothetical time limit (in seconds) set for each item. As explained before in RPCM the raw counts of correct answers or the counts of errors within a set of items are modeled. Here each block is considered an item and the total number of correct checks of 2's and 7's are recorded under 'Hit' and modeled as the unit of analysis.

Figure 1: The data structure for the attention test

```
ID Grade Item Hit Miss TL
1   3     1    20  10   15
1   3     2    21  9    15
1   3     3    16  14   12
1   3     4    13  17   10
1   3     5    16  14   12
1   3     6    13  17   10
1   3     7    14  16   11
1   3     8    14  16   11
1   3     9    20  10   15
1   3     10   20  10   15
1   3     11   14  16   11
1   3     12   12  18   10
1   3     13   16  14   12
1   3     14   13  17   10
1   3     15   21  9    15
1   3     16   14  16   11
1   3     17   18  12   14
1   3     18   11  19   10
1   3     19   17  13   13
1   3     20   18  12   14
2   3     1    25  5    15
2   3     2    25  5    15
2   3     3    29  1    12
2   3     4    20  10   10
2   3     5    25  5    12
2   3     6    28  2    10
2   3     7    28  2    11
2   3     8    28  2    11
...
```

Preliminary analyses

To perform the analyses first load the package:

```
library(lme4)
```

import the dataset (here from a text file, which is not the only option):

```
attention <- read.table("Att_Hit.txt", header=TRUE)
```

A summary of the variables in the dataset can be obtained with the following function:

```
summary(attention)
```

To make sure that R treats items as categorical variables run the following code:

```
attention$Item <- as.factor(attention$Item)
```

The boxplot of the raw scores can be obtained with:

```
boxplot(attention$Hit ~ attention$Item)
```

Test takers' raw scores, i.e. the sum of the hits, can be obtained with the following function.

```
rs <- tapply(attention$Hit, attention$ID, sum)
```

A histogram of the raw score is produced by:

```
hist(rs)
```

The basic function call to fit the RPCM is:

```
fit1 <- glmer(Hit ~ -1 + Item + (1|ID), data = attention,family = poisson)
```

Note two peculiarities in the models syntax: (1) the `-1` omits a regular intercept from the model. In combination with `+ Item`, this yields item-wise intercepts, which are the easiness parameters (or in regression model terms: cell means coding). (2) The syntax `+ (1|ID)` adds a random intercept on the person level, with mean 0 and unknown variance. The following function returns the item parameters, their standard errors, and information criteria:

```
summary(fit1)
```

Output 1: The output of RPCM analysis

```
Generalized linear mixed model fit by maximum likelihood (Laplace
Approximation) ['glmerMod']
Family: poisson  ( log )
Formula: Hit ~ -1 + (1 | ID) + Item
Data: attention

   AIC      BIC   logLik    deviance df.resid
```

```
 24945.9  25080.9 -12452.0  24903.9     4539

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.4315 -0.4723 -0.0100  0.4440  2.4441

Random effects:
 Groups Name        Variance Std.Dev.
 ID     (Intercept) 0.02276  0.1509
Number of obs: 4560, groups:  ID, 228

Fixed effects:
       Estimate Std. Error z value Pr(>|z|)
Item1   2.97307    0.01793   165.8   <2e-16 ***
Item2   2.86497    0.01863   153.8   <2e-16 ***
Item3   2.91227    0.01832   159.0   <2e-16 ***
Item4   2.88115    0.01852   155.6   <2e-16 ***
Item5   2.86324    0.01864   153.6   <2e-16 ***
Item6   2.89587    0.01842   157.2   <2e-16 ***
Item7   2.85303    0.01871   152.5   <2e-16 ***
Item8   2.95110    0.01807   163.3   <2e-16 ***
Item9   2.94519    0.01811   162.6   <2e-16 ***
Item10  3.04886    0.01748   174.4   <2e-16 ***
Item11  2.94973    0.01808   163.2   <2e-16 ***
Item12  2.95019    0.01808   163.2   <2e-16 ***
Item13  2.89922    0.01840   157.6   <2e-16 ***
Item14  2.93464    0.01818   161.5   <2e-16 ***
Item15  2.97130    0.01795   165.6   <2e-16 ***
Item16  2.93764    0.01816   161.8   <2e-16 ***
Item17  2.93418    0.01818   161.4   <2e-16 ***
Item18  2.91439    0.01830   159.2   <2e-16 ***
Item19  2.90707    0.01835   158.4   <2e-16 ***
Item20  2.90184    0.01839   157.8   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

In Output 1 the Akaike Information Criterion (AIC, Akaike, 1974) and Bayesian Information Criterion (BIC, Schwarz, 1978) and log item easiness parameters are given. The easiest item is Item 10 with easiness parameter 3.04 and the hardest item is Item 7 with an easiness estimate of 2.853. The standard deviation of the log person ability parameters is 0.15 and their mean is constrained to zero for model identification.

Confidence intervals for the item parameters can be obtained using the following code:

```
confint(fit1)
```

The confidence intervals are helpful to assess the amount of uncertainty in the estimation of item easiness parameters. Note that the first confidence interval is not for an easiness parameter, but for the standard deviation of the person parameters. Since the lower bound is clearly separated from zero there is variance in the person parameters which amounts to individual differences that the test is able to discern.

In attention or processing speed tests the contents of the items do not differ much. Usually the same items with little variation in content are repeated across the test. Since the structure and content of the blocks are the same we can expect equal difficulty for the items. For comparison purposes we can run a model that assumes equal difficulty for all the items but includes a person parameter (random intercept model):

```
fit0 <- glmer(Hit ~1+ (1|ID),data = attention, family = poisson)
```

Output 2: Output of RPCM analysis with equal item parameters

```
Generalized linear mixed model fit by maximum likelihood (Laplace
Approximation) ['glmerMod']
Family: poisson  ( log )
Formula: Hit ~ (1 | ID)
Data: attention
     AIC       BIC    logLik  deviance  df.resid
 25082.01  25094.86 -12539.01  25078.01      4558
Random effects:
Groups Name        Std.Dev.
ID     (Intercept) 0.1509
Number of obs: 4560, groups:  ID, 228
Fixed Effects:
(Intercept)
      2.925
```

Outputs 1 and 2 show that the AIC and BIC for the model named ‘fit1’ are smaller than those for ‘fit0’. That is, the model with equal item parameters does not fit as good as the model where

different difficulty parameters are assumed for the items. Hence, the item difficulties vary across the items. To compare the fit of the two models with a hypothesis test, run the following code:

```
anova(fit0, fit1)
```

which gives Output 3.

Output 3: Comparison of the fit of the two models

```
Data: attention
Models:
fit0: Hit ~ (1 | ID)
fit1: Hit ~ -1 + (1 | ID) + Item
     Df   AIC   BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
fit0  2 25082 25095 -12539    25078
fit1 21 24946 25081 -12452    24904 174.06     19  < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Output 3 depicts the information criteria, the log-likelihood, and the deviance statistic (-2*loglikelihood) for each model. A likelihood ratio test (LRT) with a chi square test statistic is computed to compare the fit of the two models. The value of the chi square (174.6) statistic is significant at $p<.001$, df =19. In other words, the model with different item parameters (fit1) significantly fits better than the model with equal item parameters (fit0).

To obtain predicted score for each item and person run the following code:

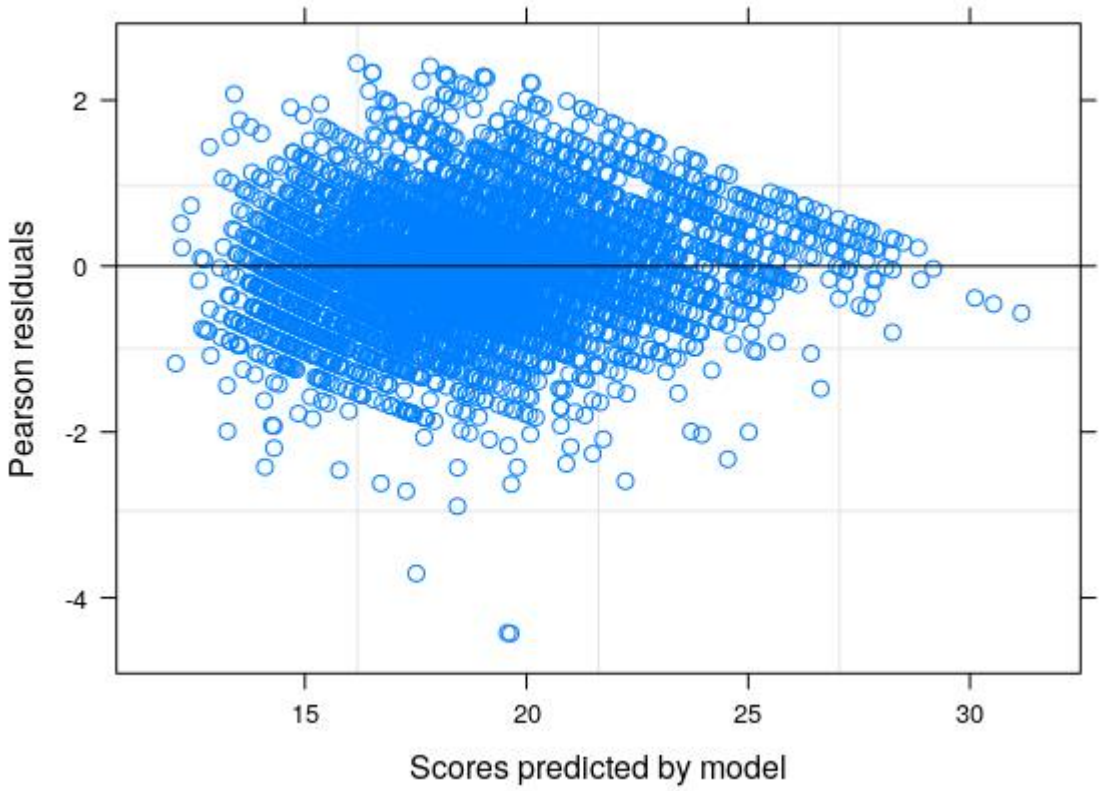
```
pre <- fitted(fit1)
```

For graphical check of the model run the following code:

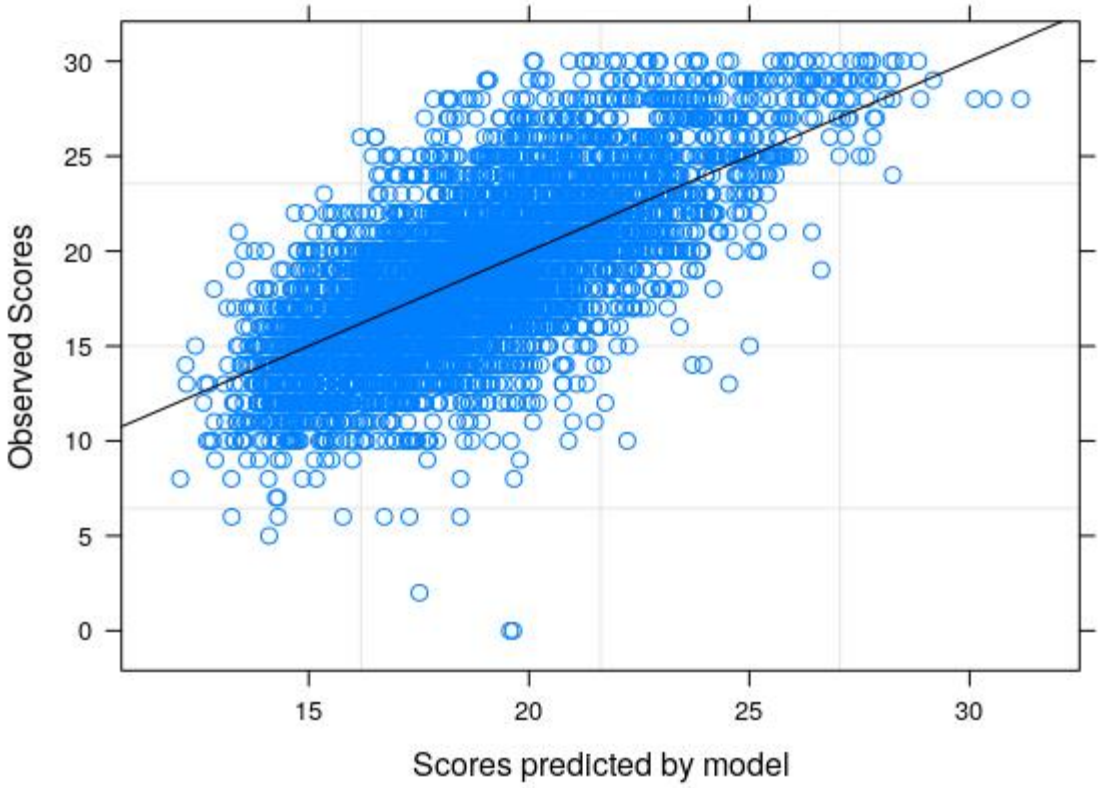
```
plot(fit1, xlab = "Scores predicted by model", ylab = "Pearson residuals")
```

Figure 2: Graphical overall model check

a. Predicted scores against their Pearson residuals



b. Predicted scores against observed scores



In Figure 2a we see the predicted values for each person on the *x*-axis and the Pearson-residuals on the *y*-axis. For good model fit, Pearson residuals have a mean of 0 and S.D. of 1.0. The more the points diverge from these values, the worse the fit. They are assumed to be roughly symmetrical (which is the case here) and they should (roughly) follow a normal distribution, at least when counts are not small. Here, they are too small for large values of the predicted values, i.e. there is less variance than the model predicts. This is not necessarily detrimental and just means that the Poisson distribution predicts too much variance here, hence our data is underdispersed, potentially leading to more conservative inference (Zeviani, Ribeiro, Bonat, Shimakura, & Muniz, 2014). Additionally, we can plot the predicted values against the observed values and obtain a regression line (Figure 2b) with the following code:

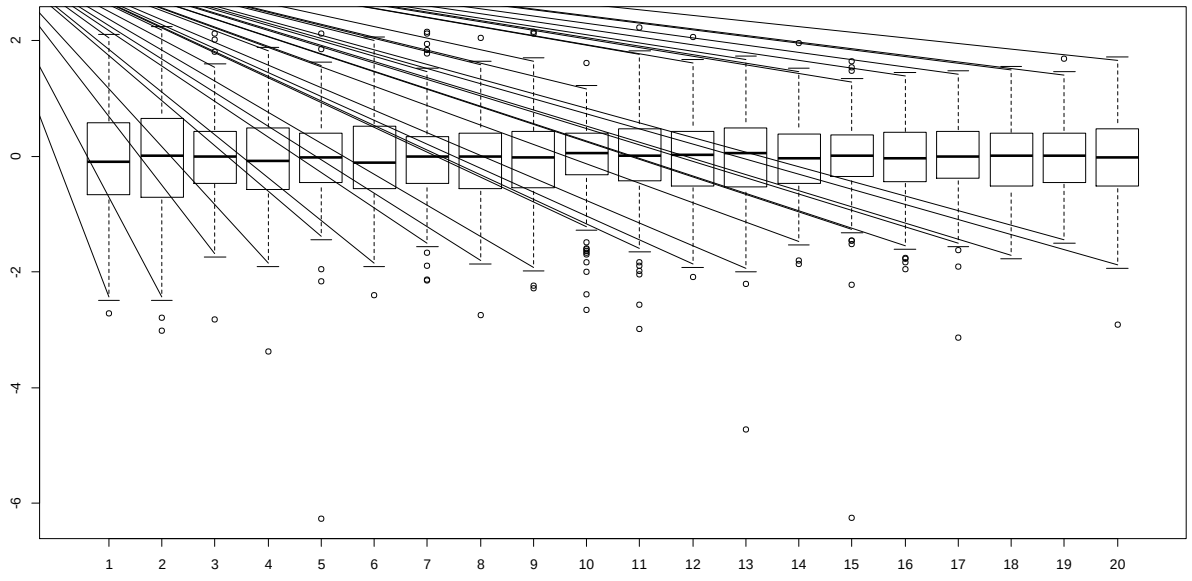
```
plot(fit1, Hit ~ fitted(.), abline = c(0,1), xlab = "Scores predicted by
model", ylab = "Observed Scores")
```

Fit of the individual items can be evaluated by graphical inspection of the residuals. The following code can be run:

```
boxplot(resid(fit1) ~ attention$Item)
```

which returns Figure 3. The residuals are the difference between the observed scores and the scores predicted by the Poisson model. The boxes and the whiskers indicate the range of the residuals. The residuals should roughly span from -2 to 2, i.e. approximately the 2.3%- and 97.7% quantiles of a standard normal distribution that approximates the distribution of the residuals. Some outliers are expected. As Figure 3 shows only Items 1 and 2 have a substantial amount of residuals which exceed ±2.

Figure 3: Graphical item fit check



The item parameters given in Output 1 are on the log-scale (additive parameterization from Equation 5). By exponentiation we can obtain the item parameters on the multiplicative scale (Equation 3):

```
exp(fixef(fit1))
```

The next function returns the person parameters:

```
ranef(fit1)
```

Output 4 depicts the person parameters for the first 10 respondents out of the total of 228 respondents.

Output 4: Person parameters

```
$ID
      (Intercept)
1   -0.1318555277
2    0.3244358819
3   -0.2692883165
4    0.0099088747
5   -0.0189995007
6    0.3902893258
7    0.0945963592
8   -0.0238910051
9    0.2409053514
10   0.0835225095
```

Likewise, we can obtain the person parameters on the multiplicative scale using the following code:

```
exp(ranef(fit1)$ID)
```

To obtain the summary of person parameters use the following functions, depending on which person parameters you obtained:

```
summary(ranef(fit1)$ID[,1])

summary(exp(ranef(fit1)$ID[,1]))
```

RPCM with separate time limits for items

In the analysis above all the 20 items had an equal time limit. However, in many such tests different time limits are imposed for each task. Let's assume that different time limits (between 10 to 15 seconds in this example) are set for the 20 attention items. Now we want to estimate RPCM considering these time limits. A new column in the dataset needs to be created to record the time limit for each item if it is not already part of the data in long format. First a vector for the time limits should be created in the order of the items:

```
timelimit <- c(15, 15, 12, 10, 12, # Items 1-5
               10, 11, 11, 15, 15, # Items 6-10
               11, 10, 12, 10, 15, # Items 11-15
               11, 14, 10, 13, 14) # Items 16-20
```

Then run the following lines of code:

```
attention$timelimit <- timelimit[as.numeric(attention$Item)]
```

The time limits in the multiplicative form define the units of the easiness parameter. As the model is fit on the log-scale, the time limits need to be log transformed and added as an offset in the additive form of the model, i.e., a known constant added to the regression equation. This amounts to the factor τ_i in the multiplicative parametrization ($\mu_{vi} = \tau_i \theta_v \sigma_i$). Consequently, one only needs to add an offset argument to the `glmer` call:

```
fit2<-glmer(Hit~-1+Item+(1|ID),data=attention,offset=
log(attention$timelimit), family = poisson)
```

Output 5: The output of RPCM with different time limits for the items

```
Generalized linear mixed model fit by maximum likelihood (Laplace
```

```
Approximation) [glmerMod]
Family: poisson  ( log )
Formula: Hit ~ 1 + (1 | ID) + Item
Data: attention
Offset: log(attention$timelimit)

     AIC      BIC   logLik deviance df.resid
 24945.9  25080.9 -12452.0  24903.9     4539

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.4316 -0.4723 -0.0100  0.4440  2.4441

Random effects:
 Groups Name        Variance Std.Dev.
 ID     (Intercept) 0.02276  0.1509
Number of obs: 4560, groups:  ID, 228

Fixed effects:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.265015   0.017935  14.776  < 2e-16 ***
Item2       -0.108091   0.021642  -4.994 5.90e-07 ***
Item3        0.162362   0.021377   7.595 3.07e-14 ***
Item4        0.313550   0.021550  14.550  < 2e-16 ***
Item5        0.113325   0.021652   5.234 1.66e-07 ***
Item6        0.328259   0.021468  15.291  < 2e-16 ***
Item7        0.190133   0.021710   8.758  < 2e-16 ***
Item8        0.288190   0.021166  13.616  < 2e-16 ***
Item9       -0.027883   0.021198  -1.315 0.188395
Item10       0.075801   0.020662   3.669 0.000244 ***
Item11       0.286820   0.021174  13.546  < 2e-16 ***
Item12       0.382594   0.021171  18.071  < 2e-16 ***
Item13       0.149296   0.021449   6.960 3.39e-12 ***
Item14       0.367035   0.021255  17.268  < 2e-16 ***
Item15      -0.001758   0.021059  -0.083 0.933464
Item16       0.274716   0.021239  12.935  < 2e-16 ***
Item17       0.030100   0.021257   1.416 0.156784
Item18       0.346778   0.021365  16.231  < 2e-16 ***
Item19       0.077103   0.021406   3.602 0.000316 ***
Item20      -0.002229   0.021435  -0.104 0.917166
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

As you can see in Output 5 the item parameters change but the information criteria are identical to those in ‘fit1’, because including the offset merely shifts parameters without a change in the log-likelihood. In other words, including the a priori known time limits does not change the model fit, but merely enhances interpretation of the easiness parameters: On the multiplicative scale, they

can now be interpreted as the expected number of points scored / errors made by a person of average ability (average ability being zero as person parameters are centered at zero) within the imposed time unit, i.e., a second in this study. The code given to produce the statistics for 'fit1' can be used to obtain the statistics for 'fit2'.

Investigating item fit

To investigate item fit, we propose a simple chi-square type statistic: Let the set G of person indices be subdivided into K subsets, $G_1, \ldots, G_K$, formed by ordering the persons by total score and binning them. In the application to the attention data $K = 5$ subsets will be used, corresponding to the quantiles of the total score distribution. The chi-square type statistic for item i is calculated by dividing the squared difference of the sums of the predictions $\hat{\mu}_{vi}$ of the model and the sum of the scores Y_{vi} by sums of the predictions for each group and summing over the groups:

$$X_i^2 = \sum_{k=1}^{K} \frac{\left(\sum_{v \in G_k} Y_{vi} - \hat{\mu}_{vi}\right)^2}{\sum_{v \in G_k} \hat{\mu}_{vi}}$$

If the model holds, then X_i^2 asymptotically follows a χ^2-distribution with K degrees of freedom, that is, for large samples the distribution of the statistic is well approximated by a χ^2-distribution. One can then proceed and investigate and/or remove misfitting items as in other IRT models by comparing X_i^2 to quantiles of the χ^2-distribution with K degrees of freedom or calculating (approximate) p-values. Defining and applying the function requires the following code:

```
chisquares <- function(glmerMod, theta_name, item_ids, person_ids,
                       quantiles = seq(0,1,by=0.2)){
  if(!class(glmerMod) == "glmerMod"){
    stop("expected an object of class glmerMod. Use an output of glmer!")
  }
  resp <- model.extract(model.frame(glmerMod), "response")
  pred <- predict(glmerMod, type = "response")
  theta <- ranef(glmerMod)[theta_name][[1]][[1]]
  groups <- cut(theta, quantile(theta, quantiles))[person_ids]
```

```
  d <- data.frame(resp, pred, groups)

  # define function to calculate itemwise chisquare statistic
  itemwise <- function(data){
    resp <- data$resp
    p <- data$pred
    groups <- data$groups

    expected <- tapply(p, groups, sum)
    observed <- tapply(resp, groups, sum)
    X2 <- (expected - observed)^2 / expected
    sum(X2)
  }
  # return itemwise statistic:
  by(d, item_ids, itemwise)
}

itemfit1 <- chisquares(fit1, theta_name = "ID", item_ids = attention$Item,
person_ids = attention$ID)
```

For the attention data, items 5 and 10 show misfit, i.e., the chi square-values are larger than the .99-quantile of a chi square distribution with five degrees of freedom.

```
which(1- pchisq(itemfit1, 5) < 0.01) # check which items have a small p-value;

df=5 and refers to the number of subsets of the persons
```

We omit a reanalysis with these items removed and mention in passing, that the asymptotic argument can be replaced by a parametric bootstrap in small samples.

Differential Item Functioning

When one or more groups are to be compared with IRT models, the comparison should be based on means of latent variables. The prerequisite for such a comparison is that the latent variables of both groups are on the same scale, otherwise one could merely be observing an artefactual difference (e.g. Holland & Thayer, 1988). As the scale is implied by the item difficulty parameters,

it is vital to check that item difficulty parameters are identical, i.e., the items do not function differentially or that measurement invariance (Millsap, 2011) holds.

We now discuss a Differential Item Functioning (DIF) detection method in the context of the RPCM, that builds on well-known approaches for existing IRT models. Similar to binary or ordinal IRT models, detecting DIF items is complicated by the fact, that latent means from the two groups cannot be assumed to be equal. We propose a modelling approach that first uses an LRT to investigate globally whether DIF is present. In the absence of DIF, group means can then be compared. The method also flags items as DIF items and by subsequently eliminating flagged items, a DIF-free item set can be obtained.

In the attention test example, 3rd and 4th graders have been tested (N_{3rd} = 174, N_{4th} = 54) and we investigate, whether there is a difference in latent means for these two groups. In a first step, we extend the model with different time limits (fit2) by group-specific intercepts:

```
# Grade is 0-1 coded with 0 = 3rd grade and 1=4th grade

fit3 <- glmer(Hit ~ -1 + Grade+Item+(1|ID), data =
attention,offset=log(attention$TL),family=poisson,control=glmerControl(optimi
zer="bobyqa",optCtrl=list(maxfun=2e5)))
```

Note that for numerical reasons, we have used a different optimizer (bobyqa) and a higher number of function evaluations (maxfun), which avoids mild convergence problems. The `glmer` function provides several alternative model fitting procedures (optimizers) and sometimes default values need to be adjusted to ensure model fitting proceeds smoothly.

Output 6: Differential Item Functioning: Baseline Model

```
Generalized linear mixed model fit by maximum likelihood (Laplace
Approximation) ['glmerMod']
```

```
Family: poisson  ( log )
Formula: Miss ~ -1 + Grade + (1 | ID) + Item
Data: attention
Offset: log(attention$TL)
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))

     AIC      BIC   logLik deviance df.resid
 24795.5  24936.8 -12375.7  24751.5     4538

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2387 -0.6029  0.0094  0.6030  6.4120

Random effects:
Groups Name        Variance Std.Dev.
ID     (Intercept) 0.1008   0.3175
Number of obs: 4560, groups:  ID, 228

Fixed effects:
Estimate Std. Error z value Pr(>|z|)
Grade  -0.03472    0.05077  -0.684  0.49402
Item1  -0.31344    0.16695  -1.878  0.06045 .
Item2  -0.13749    0.16674  -0.825  0.40961
Item3   0.01648    0.16682   0.099  0.92130
Item4   0.24139    0.16677   1.448  0.14776
Item5   0.09174    0.16673   0.550  0.58215
Item6   0.22540    0.16679   1.351  0.17655
Item7   0.19329    0.16672   1.159  0.24629
Item8   0.03662    0.16690   0.219  0.82635
Item9  -0.27107    0.16689  -1.624  0.10433
Item10 -0.48502    0.16719  -2.901  0.00372 **
Item11  0.03290    0.16690   0.197  0.84371
Item12  0.12945    0.16690   0.776  0.43796
Item13  0.01917    0.16681   0.115  0.90849
Item14  0.16114    0.16686   0.966  0.33419
Item15 -0.31215    0.16694  -1.870  0.06151 .
Item16  0.05214    0.16688   0.312  0.75471
Item17 -0.17454    0.16686  -1.046  0.29556
Item18  0.19533    0.16682   1.171  0.24163
Item19 -0.05513    0.16681  -0.330  0.74104
Item20 -0.11973    0.16680  -0.718  0.47286
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The parameter of the Grade variable is the difference in latent means on the log-scale. Here, 4th graders are slightly less skilled. The item parameters are assumed to be equal in this model for both groups. As we have introduced the difference in latent means, the item parameters deviate

slightly from this in fit1. We now add group by item interaction terms that represent the group-specific deviations in item difficulty:

```
fit44 <- glmer(Hit ~ -1 + Grade +Item+(1|ID) + Item*Grade, data = attention,
offset=log(attention$TL),family=poisson,control=glmerControl(optimizer="bobyq
a",optCtrl=list(maxfun=2e5)))
```

Output 7: Differential Item Functioning: Interaction Model

```
Generalized linear mixed model fit by maximum likelihood (Laplace
Approximation) ['glmerMod']
Family: poisson  ( log )
Formula: Miss ~ -1 + Grade + (1 | ID) + Item + Item * Grade
Data: attention
Offset: log(attention$TL)
Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))

     AIC      BIC   logLik deviance df.resid
 24806.4  25069.8 -12362.2  24724.4     4519



Scaled residuals:
Min      1Q  Median      3Q     Max
-3.2378 -0.6033  0.0045  0.6033  6.2416

Random effects:
 Groups Name        Variance Std.Dev.
 ID     (Intercept) 0.1008   0.3175
Number of obs: 4560, groups:  ID, 228

Fixed effects:
             Estimate Std. Error z value Pr(>|z|)
Grade         0.03749    0.06905   0.543   0.5872
Item1        -0.54725    0.22572  -2.425   0.0153 *
Item2        -0.06851    0.21853  -0.314   0.7539
Item3         0.33401    0.22384   1.492   0.1357
Item4         0.33312    0.22001   1.514   0.1300
Item5        -0.02096    0.21713  -0.097   0.9231
.                 .          .       .       .
.                 .          .       .       .
.                 .          .       .       .
Grade:Item2  -0.09356    0.06594  -1.419   0.1560
Grade:Item3  -0.17071    0.06774  -2.520   0.0117 *
Grade:Item4  -0.10060    0.06641  -1.515   0.1298
Grade:Item5  -0.03736    0.06539  -0.571   0.5677
```

```
.                  .          .       .       .
.                  .          .       .       .
.                  .          .       .       .
Grade:Item10 -0.24283    0.07382  -3.290   0.0010 **
.                  .          .       .       .
.                  .          .       .       .
.                  .          .       .       .


---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The interaction of Grade and the individual items (the lines starting with Grade:ItemX) are the difference in item difficulty on log-scale of the 4th graders relative to the 3rd graders. For example, item 2 is slightly more difficult for the 4th graders (-0.094), but the difference is not significant. The difference of -0.170 for Item 3 is significant. Note however, that with 20 items, a multiple testing problem exists and one should not interpret significant interaction terms before conducting a global DIF test:

By comparing the two models with an LRT, we test whether the interaction terms explain variability in the data, which amounts to a global DIF test. Here we find:

Output 8: LRT for global DIF test

```
> anova(fit3, fit4)

Data: attention
Models:
fit3: Miss ~ -1 + Grade + (1 | ID) + Item
fit4: Miss ~ -1 + Grade + (1 | ID) + Item + Item * Grade
     Df   AIC   BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
fit3 22 24796 24937 -12376    24752
fit4 41 24806 25070 -12362    24724 27.069     19     0.1031
```

A p-value of 0.10 results, indicating that the model with the interaction term does not explain more variability in the data, i.e., the LRT provides no evidence for global DIF. It is hence not necessary

to remove items to limit the amount of item level DIF. We can now proceed and test for group differences by comparing fit2 and fit3:

Output 9: Testing for Global DIF

```
> anova(fit2, fit3)

Data: attention
Models:
fit2: Miss ~ -1 + (1 | ID) + Item
fit3: Miss ~ -1 + Grade + (1 | ID) + Item
     Df   AIC   BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
fit2 21 24794 24929 -12376    24752
fit3 22 24796 24937 -12376    24752 0.4685      1     0.4937
```

We find no differences in latent means (p =0.49), i.e., 3rd and 4th graders have comparable latent attention ability. Alternatively, the z-test for the coefficient of Grade in fit3 can be used (which is also not significant, z = -0.69, p = 0.49).

In some scenarios one wants to study DIF on the item level. By studying the interaction terms in fit4 item-wise, we see which items have been flagged by the procedure. Here items 3 and 10 are flagged, and we could proceed to purify the item set by removing the flagged item with the smaller p-value first (item 10 here) and refitting the model of fit4 to the reduced item set to investigate the remaining item set. We caution, however, that this sequential procedure might be suboptimal in terms of the nominal alpha level.

Dispersion

In a Poisson model (and by implication in the RPCM) the assumption is that the variance of the dependent variable *Y* given covariates *X* is equal to its expectation, i.e.

$$\mathrm{Var}(Y|X) = \mathrm{E}(Y|X) \qquad (7)$$

In reality, this equidispersion assumption is frequently violated. The φ coefficient is essentially the ratio of model implied variance to predicted mean (McCullagh & Nelder, 1989). It is expected to be 1. Three scenarios may occur:

1. φ is roughly equal to 1; the assumption of equal mean and variance is met.
2. $\varphi < 1$ indicates underdispersion. This means that there is less variance in the data than the Poisson model predicts. This is a very commonly observed scenario in test data. This entails that confidence intervals calculated from a model fit are too wide. Also, in our context, the reliability is underestimated, i.e., the test seems less reliable than it really is. Clearly, this is undesirable and needs to be addressed. Unfortunately, the problem of underdispersion has been widely ignored in the literature and dispersion modeling normally focuses on overdispersion.
3. $\varphi > 1$ indicates overdispersion. This means that there is more variance in the data than the Poisson model predicts. Confidence intervals for model parameters are too narrow and reliability is overestimated. Again, this in undesirable.

There are some strategies to address under- and overdispersion: (1) deleting misfitting items or exclude persons (especially those without any mistakes might be the cause of an overly low variance), (2) employ a kind of ad-hoc statistical correction via a so-called quasi-Poisson-Regression. This has been advised for the case of overdispersion in the literature, but it is not a standard procedure in the random effects case, and (3) employ extensions of the Poisson-distribution, say the Conway-Maxwell-Poisson distribution (Boatwright, Borle, & Kadane, 2003; Shmueli, Minka, Kadane, Borle, & Boatwright, 2005) or the Gamma-count distribution (Zeviani et al., 2014). However, currently no standard procedures or software for these approaches are available.

Generally, overdispersion is considered to be worse than underdispersion, because if ignored statistical inference is anti-conservative: When overdispersion occurs, *SE*s from a Poisson model are spuriously too small and when undersidpersion occurs they are artificially too large. To estimate φ the following code should be run:

```
phi <- function(fit){
    y <- fit@resp$y
    pred <- exp(predict(fit))
    sum((y - pred)^2/pred)/length(y)
 }


phi(glmer(Hit~-1+(1|ID)+Item,data=attention, family = poisson))
[1] 0.531849
```

Which returns a φ equal to .53. The value of φ is way smaller than 1 which indicates underdispersion. The model is prone to underestimate reliability and inference on model parameters is conservative.

Reliability

As in other IRT models, reliability varies as a function of ability. Figure 4 shows reliability for different ability estimates. The horizontal axis depicts the ability continuum and the vertical axis depicts the reliability estimate. The plot implies that the precision of the test changes as a function of the ability. The reliability estimates for different locations on the log ability scale can be read from plot. For example the reliability for examinees with ability -1 on the log scale is .90 and for examinees with ability -.50 is. 94.

Note that we use a reliability estimate on log scale here, in contrast to Verhelst and

Kamphuis (2009) and Doebler and Holling (2015). Specifically, from an estimate $\hat{s}^2_{\hat{\theta}}$ of the latent variance of the person parameters on log scale, and a squared standard error s_v^2 for the *v*th person's ability, we define

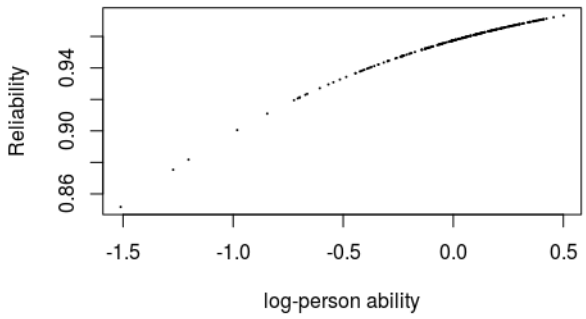
$$\hat{s}^2_{\hat{\theta}} / \left(\hat{s}^2_{\hat{\theta}} + s_v^2 \right)$$

as the person specific reliability estimate for person *v*.

Figure 4 was produced with the following lines of code:

```
theta_ests <- as.data.frame(ranef(fit1, condVar = TRUE))

var_log_theta <- as.numeric(VarCorr(fit1))

plot(theta_ests$condval, var_log_theta/(var_log_theta + theta_ests$condsd^2),

     ylab = "reliability estimate", # label of y-axis

     xlab = "log-person ability estimate", # label of x-axis

     pch = 19, # type of symbol (19 = small dot)

     cex = .1) # size of symbol (.1 = one tenth of default size)
```

Figure 4: Reliability graph



Conclusion

In this article the RPCM was reviewed and R functions were given to fit the model. The 'lme4' package (Bates, et al., 2017) in R was employed to estimate the model parameters. The data of 228 respondents to an attention test developed after the model of "Ruff 2 and 7" test (Ruff, et al., 1988) were analyzed and the outputs were interpreted. Item residuals and a chi-square type statistic showed that only two items misfit the model. Overall graphical and statistical model checks indicated that the attention data fit the RPCM, but the data was underdispersed, making inference conservative and biasing reliability estimates downwards.

Instead of modeling the counts of the correctly checked items (Hit) we could have modeled the counts of missed items (Miss). The analysis of the counts of missed items, which is not reported here due to space limitations, had a better fit to the data. In the original application of the model by Georg Rasch also the oral misreadings or errors were modeled instead of the correct words.

The MML-approach as implemented with the glmer function here has several advantages and some practical and statistical limitations: On the plus side, the GLMM framework is very flexible. One can add covariates as fixed effects and incorporate additional random effects, for example to model multilevel structures in the data that appear naturally in many (educational) applications, such has

persons nested in classrooms which are maybe even nested in schools (e.g. Aiken, Mistler, Coxe, & West, 2015). Adding fixed effects is interesting to explain variation in ability, say to predict ability by gender, age or experimental conditions. While we cannot cover all these techniques in this brief example, we refer the reader to the authoritative monograph on explanatory IRT models by de Boeck and Wilson (2004) and the tutorial by de Boeck et. al. (2011) for IRT-models for dichotomous data. However, by assuming log-normal person parameters, the MML-approach is a bit more restrictive than conditional maximum likelihood (CML). From our experience, the differences in item parameter estimates are minor, i.e., they are essentially only rescaled. Also, glmer can be computationally intensive for large datasets and complex models.

It is worth noting in this context, that count data items are typically more informative of person ability than binary items. This has two consequences: Person parameter estimates are reliable, even when only a few items are used. Depending on an item's mean and the variance of the person parameters, as few as three items will give reliable person parameter estimates. The second implication is that as a rule of thumb sample size requirements for RPCM are more modest compared to the dichotomous case, which includes the explanatory models mentions above.

References

Aiken, L. S., Mistler, S. A., Coxe, S., & West, S. G. (2015). Analyzing count variables in individuals and groups: Single level and multilevel models. *Group Processes & Intergroup Relations*, *18*, 290-314.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*, 716-723.

Baghaei, P., Ravand, H., & Nadri, M. (in press). Is the d2 test of attention Rasch scalable? Analysis with the Rasch Poisson Counts Model. *Perceptual and Motor Skills*.

Bates, D., Maechler, M., Bolker, B, Walker, S., Christensen, R., Singmann, H., et al. (2017). *lme4: Linear Mixed-Effects Models using 'Eigen' and S4.* R package, version 1.1-14. https://cran.r-project.org/web/packages/lme4/index.html

Beyzaee, S. Z. (2017). *A latent variable modeling of verbal reasoning, cognitive flexibility, processing speed, sustained attention, and reading comprehension among Iranian EFL learners*. Unpublished master's thesis. Mashhad: Islamic Azad University.

Boatwright, P., Borle, S., & Kadane, J. B. (2003). A model of the joint distribution of purchase quantity and timing. *Journal of the American Statistical Association, 98*, 564-572.

Brown, H. & Prescott, R. (2015*). Applied mixed models in medicine* (3rd Ed.). Hoboken, NJ: John Wiley & Sons. doi:10.1002/9781118778210

de Boeck, P. & Wilson, M. (2004, Eds.). *Explanatory Item Response Models: A Generalized Linear and Nonlinear Approach*. New York: Springer.

de Boeck, P., Bakker, M., Zwitser, R., Nivard, M., Hofman, A., Tuerlinckx, F., & Partchev, I. (2011). The estimation of item response models with the lmer function from the lme4 package in R. *Journal of Statistical Software*, *39*, 1-28.

Demidenko, E. (2013). *Mixed models: theory and applications*. Hoboken, NJ: John Wiley & Sons.

Doebler, A., Doebler, P., & Holling, H. (2014). A latent ability model for count data and application to processing speed. *Applied Psychological Measurement, 38*, 587–598.

Doebler, A., & Holling, H. (2016). A processing speed test based on rule-based item generation: An analysis with the Rasch Poisson Counts model. *Learning and Individual Differences*, *52*, 121-128. doi: 10.1016/j.lindif.2015.01.013.

Forthmann, B., Gerwig, A., Holling, H., Çelik, P., Storme, M., & Lubart, T. (2016). The be-creative effect in divergent thinking: The interplay of instruction and object frequency. *Intelligence, 57*, 25-32.

Holland, P.W., & Thayer, D.T. (1988). Differential item performance and the Mantel-Haenszel procedure. In H. Wainer & H. Braun (Eds.), *Test validity* (pp. 129–145). Hillsdale, NJ: Lawrence Erlbaum Associates.

Jansen, M.G.H. (1997a). Rasch's model for reading speed with manifest exploratory variables. *Psychometrika, 62*, 393–409.

Jansen, M.G.H. (1997b). Applications of Rasch's Poisson counts model to longitudinal count data. In Langeheine, R., & Rost, J. (Eds.), *Applications of latent trait and latent class models in the social sciences* (pp. 380-388). Münster: Waxmann.

Jansen, M. G., & van Duijn, M. A. (1992). Extensions of Rasch's multiplicative Poisson model. *Psychometrika, 57*, 405-414.

Jansen, M. G. H. (1994). Parameters of the latent distribution in Rasch's Poisson counts model. In G. Fischer & D. Laming (Eds.), *Contributions to mathematical psychology, psychometrics, and methodology* (pp. 319-326). New York, NY: Springer.

Jansen, M. G. H. (1995). The Rasch Poisson counts model for incomplete data: An application of the EM algorithm. *Applied Psychological Measurement, 19,* 291-302.

Jansen, M. G. H., & van Duijn, M. (1992). Extensions of Rasch's multiplicative Poisson model. *Psychometrika, 57*, 405-414.

Jansen, M. G. H. (2003). Estimating the parameters of a structural model for the latent traits in Rasch's model for speed tests. *Applied Psychological Measurement, 27*, 138–151. doi: 10.1177/0146621602250536

Jansen, M. G. H., & Glas, C. A. W. (2005). Checking the assumptions of Rasch's model for speed tests. *Psychometrika, 70,* 671–684. Doi: 10.1007/s11336-001-0929-2

McCullagh, P., & Nelder, J. A. (1989). *Generalized linear models*. New York: Chapman & Hall/CRC.

Masters, G.N. & Wright, B.D. (1984). The essential process in a family of measurement models. *Psychometrika, 49*, 529-544.

Masters, G. N. (1988). Measurement models for ordered response categories. In Langeheine, R. & Rost, J. (Eds.), *Latent trait and latent class models* (pp. 11-29). New York: Springer.

Millsap, R.E. (2011). *Statistical approaches to measurement invariance*. New York: Routledge.

Rasch, G. (1960/1980). *Probabilistic models for some intelligence and attainment tests* (Expanded Ed.). Chicago: University of Chicago Press.

Rasch, G. (1977). On specific objectivity: An attempt at formalizing the request for generality and validity of scientific statements. *Danish Yearbook of Philosophy, 14*, 58-93.

R Development Core Team (2016). *R: A language and environment for statistical computing.* Vienna, Austria: R Foundation for Statistical Computing. http: //www.R-project.org/).

Ross, S. M. (1983). *Stochastic processes*. New York: Wiley.

Ruff, R. M., Evans, R. W., & Light, R. H. (1986). Automatic detection vs. controlled search: a paper-and-pencil approach. *Perceptual & Motor Skills, 62*, 407-416.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics, 6*, 461-464.

Shmueli, G., Minka, T. P., Kadane, J. B., Borle, S., & Boatwright, P. (2005). A useful distribution for fitting discrete data: revival of the Conway–Maxwell–Poisson distribution. *Journal of the Royal Statistical Society: Series C (Applied Statistics), 54*, 127-142.

Spray, J. A. (1990). One-parameter item response theory models for psychomotor tests involving repeated independent attempts. *Research Quarterly for Exercise and Sport, 61*, 162-168.

Verhelst, N.D., Kamphuis, F.H. (2009). *A Poisson-Gamma model for speed tests*. Measurement and Research Department Reports 2009-2. Cito, Arnhem.

Zeviani, W. M., Ribeiro Jr, P. J., Bonat, W. H., Shimakura, S. E., & Muniz, J. A. (2014). The Gamma-count distribution in the analysis of experimental underdispersed data. *Journal of Applied Statistics, 41*, 2616-2626.