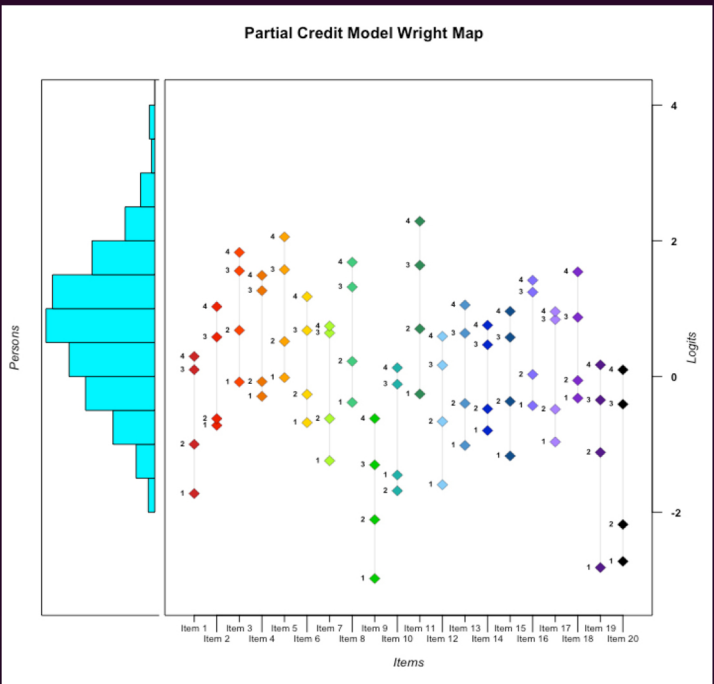
The R Series

Rasch Measurement Theory Analysis in R



Stefanie Wind
Cheng Hua

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Rasch Measurement Theory Analysis in R

Chapman & Hall/CRC
The R Series

Recently Published Titles

Using R for Modelling and Quantitative Methods in Fisheries
Malcolm Haddon

R For Political Data Science: A Practical Guide
Francisco Urdinez and Andres Cruz

R Markdown Cookbook
Yihui Xie, Christophe Dervieux, and Emily Riederer

Learning Microeconometrics with R
Christopher P. Adams

R for Conservation and Development Projects: A Primer for Practitioners
Nathan Whitmore

Using R for Bayesian Spatial and Spatio-Temporal Health Modeling
Andrew B. Lawson

Engineering Production-Grade Shiny Apps
Colin Fay, Sébastien Rochette, Vincent Guyader, and Cervan Girard

Javascript for R
John Coene

Advanced R Solutions
Malte Grosser, Henning Bumann, and Hadley Wickham

Event History Analysis with R, Second Edition
Göran Broström

Behavior Analysis with Machine Learning Using R
Enrique Garcia Ceja

Rasch Measurement Theory Analysis in R
Stefanie A. Wind and Cheng Hua

Spatial Sampling with R
Dick R. Brus

For more information about this series, please visit: https://www.crcpress.com/Chapman--Hall-CRC-The-R-Series/book-series/CRCTHERSER

Rasch Measurement Theory Analysis in R

Stefanie A. Wind and Cheng Hua

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
A CHAPMAN & HALL BOOK

First edition published 2022
by CRC Press
6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742

and by CRC Press
4 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

CRC Press is an imprint of Taylor & Francis Group, LLC

Library of Congress Cataloging-in-Publication Data

Names: Wind, Stefanie A. (Anne), author. | Hua, Cheng, author.
Title: Rasch measurement theory analysis in R / Stefanie Wind and Cheng Hua.
Description: First edition. | Boca Raton : CRC Press, [2022] | Series: Chapman and Hall/CRC the R series | Includes bibliographical references and index.
Identifiers: LCCN 2021058803 (print) | LCCN 2021058804 (ebook) | ISBN 9780367776398 (pbk) | ISBN 9781032005607 (hbk) | ISBN 9781003174660 (ebk)
Subjects: LCSH: Psychometrics. | Rasch models. | Psychology--Statistical methods. | Social sciences--Statistical methods.
Classification: LCC BF39 .W564 2022 (print) | LCC BF39 (ebook) | DDC 150.1/5195--dc23/eng/20220303
LC record available at https://lccn.loc.gov/2021058803
LC ebook record available at https://lccn.loc.gov/2021058804

ISBN: 978-1-032-00560-7 (hbk)
ISBN: 978-0-367-77639-8 (pbk)
ISBN: 978-1-003-17466-0 (ebk)

DOI: 10.1201/9781003174660

Typeset in Latin Modern
by KnowledgeWorks Global Ltd.

Publisher's note: This book has been prepared from camera-ready copy provided by the authors.

Contents

1

Introduction

The purpose of this book is to illustrate techniques for conducting Rasch measurement theory (Rasch, 1960) analyses using existing R packages. The book includes some background information about Rasch models, and the primary objectives are to demonstrate how to apply the models to data using R packages and how to interpret the results.

The primary audience for this book is graduate students or professionals who are familiar with Rasch measurement theory at a basic level, and who want to use the R software program (R Core Team, 2021) to conduct their Rasch analyses. We provide a brief overview of several key features of Rasch measurement theory in this chapter, and we provide descriptions of basic characteristics of the models and analytic techniques in each of the following chapters. Accordingly, we encourage readers who are new to Rasch measurement theory to use this book as a supplement to other excellent introductory texts on the subject that include a detailed theoretical and statistical introduction to Rasch measurement. For example, interested readers may find the following texts useful to begin learning about Rasch measurement theory:

> Andrich, David, and Ida Marais. 2019. *A Course in Rasch Measurement Theory: Measuring in the Educational, Social and Health Sciences.* Singapore: Springer.
>
> Bond, Trevor G., Zi Yan, and Moritz Heene. 2019. *Applying the Rasch Model: Fundamental Measurement in the Human Sciences (4th Ed.).* New York: Routledge, Taylor & Francis Group.
>
> Engelhard, George, and Jue Wang. *Rasch Models for solving measurement problems: Invariant Measurement in the Social Sciences.* Vol.187. SAGE, 2020. https://us.sagepub.com/en-us/nam/rasch-models-for-solving-measurement-problems/book267292

This book also assumes a basic working knowledge of the R software and programming language. To use this book, readers will need to know how to run existing code in R or R Studio, and how to make basic edits to existing code in order to adapt it for use with their own data. Readers who are new

DOI: 10.1201/9781003174660-1

to R may find the following resources helpful for learning how to use this program:

- What is R & R Studio: https://libguides.library.kent.edu/statconsulting/r
- Install R & R-Studio: https://rstudio-education.github.io/hopr/starting.html
- R Tutorial for beginners: https://rstudio-education.github.io/hopr/starting.html

In addition, readers should note that our descriptions of the R code generally assume that the analyses are being conducted in R Studio. However, all of the R code will work in both R and R Studio.

1.1 Overview of Rasch Measurement Theory

Georg Rasch was a Danish psychometrician who introduced a theory and approach to social science measurement in his classic text entitled *Probabilistic Models for Some Intelligence and Attainment Tests* (Rasch, 1960). This approach to measurement involves transforming ordinal item responses, such as the data that are collected in a multiple-choice educational assessment of middle school students' understanding of engineering design (Alemdar et al., 2017), a survey designed to measure self-efficacy for making career decisions (Nam et al., 2011), or a diagnostic scale used to identify individuals with depression (Shea et al., 2009), to interval-level measures for examinees and items. Now called Rasch measurement theory, this approach is based on principles and requirements that reflect measurement in the physical sciences.

Chief among the defining features of Rasch measurement theory is the emphasis on *invariance* in measurement. In the context of Rasch measurement theory, invariance occurs when the following properties are observed in item response data (Rasch, 1961):

• The comparison between two stimuli should be independent of which particular individuals were instrumental for the comparison;

• and it should also be independent of which stimuli within the considered class were or might also have been compared.

• Symmetrically, a comparison between two individuals should be independent of which particular stimuli with the class considered were instrumental for the comparison;

- and it should also be independent of which other individuals were also compared on the same or on some other occasion (pp. 331-332)

Rasch used the term *specific objectivity* (Rasch, 1977) to describe the importance of identifying specific situations in which the requirements for invariant measurement are approximated. In emphasizing invariance, Rasch noted that meaningful interpretation and use of social science measurement instruments is not possible unless invariance is approximated.

Rather than assuming that data will adhere perfectly to the model requirements, researchers who use Rasch models do so in order to identify deviations from these requirements when they occur. Information about departures from model requirements can help analysts identify areas for additional research, including qualitative investigations of persons and items, as well as guidance for improving the quality of a measurement procedure. This perspective, in which the *measurement theory* (i.e., the model) is emphasized as a guide for understanding the quality of the data by comparing it to strict requirements, is a key distinguishing feature between Rasch measurement theory and other item response theory (IRT) approaches. In typical IRT analyses, analysts attempt to identify a model that is the most accurate representation of the characteristics of the *data* (Embretson and Reise, 2000). For example, many researchers select the three-parameter logistic model (Birnbaum, 1968) when analyzing responses to multiple-choice educational assessments because the model directly incorporates instances of guessing and differences in item discrimination. However, researchers guided by a Rasch perspective would instead use the Rasch model to identify unexpected observations that could alert them to potential guessing and inconsistent item ordering over examinee achievement levels (as reflected by differences in item discrimination). These unexpected observations could then lead to additional exploration and the improvement of the assessment procedure. Although there are many situations in which reproducing the characteristics of item response data may be useful or necessary, the general perspective that characterizes Rasch measurement theory is that the theory (as reflected in the model) provides a framework for evaluating data according to its adherence to fundamental measurement properties. Rasch measurement theory scholars argue that evidence of adherence to model requirements is necessary before data can be used to make inferences about persons and items (e.g., in statistical analyses). As Bond and Fox (2019) noted, "researchers should spend more time investigating their scales than investigating with their scales" (p. 4).

In addition to providing useful information about adherence to fundamental measurement properties, Rasch models have several other theoretical and practical features that have contributed to their widespread popularity across disciplines in the social, behavioral, and health sciences (please see Rasch, 1977 for a review). Wright and Mok (2004) summarized the key theoretical and practical features of the Rasch measurement approach as follows:

> In order to construct inference from observation, the measurement model must: (a) produce linear measures, (b) overcome missing data, (c) give estimates of precision, (d) have devices for detecting misfit, and (e) the parameters of the object being measured and of the measurement instrument must be separable. Only the family of Rasch measurement models solves these problems (p. 4).

To help researchers take advantage of these useful features in a practical way, our book provides an overview of several key models within the family of Rasch models, offers basic guidance on the estimation of the models using available R packages, and provides suggestions and advice for interpreting the results from the analyses.

1.2 Online Resources

This book includes several supplemental resources that are available online, including copies of the R code, example data sets, and data sets for the challenge exercises at the end of some of the chapters. All of the materials used in this book, including the R software, R packages, and data sets, are free to download. Table 1.1 provides details about how to download all the relevant learning materials.

TABLE 1.1
Online Resources

Title	Version	Download Link
R Programming Language	4.0.3 (latest)	cran.r-project.org
R Studio	1.3	rstudio.com
Rasch Book Online Version	Beta 0.2	https://beta.rstudioconnect.com/connect/#/apps/e5fb8a2a-e6d7-4ede-8e53-888cf0129c9e/access
Source Code for this Book	GitHub	https://github.com/huacheng1985/Bookdown_CRC_Rasch
Data set	Beta	See in Each Chapter

https://www.routledge.com/Rasch-Measurement-Theory-Analysis-in-R/Wind-Hua/p/book/9781032005607

2

Dichotomous Rasch Model

This chapter provides a basic overview of the dichotomous Rasch model, along with guidance for analyzing data with the dichotomous Rasch model using R. We use data from a transitive reasoning assessment presented by Sijtsma and Molenaar (2002) to illustrate the analysis using Conditional Maximum Likelihood Estimation (CMLE) with the Extended Rasch Modeling (eRm) package (Mair et al., 2021). Then, we illustrate the application of the dichotomous Rasch model using Marginal Maximum Likelihood Estimation (MMLE) and Joint Maximum Likelihood Estimation (JMLE) with the Test Analysis Modules (TAM) package (Robitzsch et al., 2021). After the analyses are complete, we present an example description of the results. The chapter concludes with a challenge exercise and resources for further study.

Overview of the Dichotomous Rasch Model

The *dichotomous Rasch model* (Rasch, 1960) is the simplest model in the Rasch family of models (Wright and Mok, 2004). It was designed for use with ordinal data that are scored in two categories (usually 0 or 1). The dichotomous Rasch model uses sum scores from these ordinal responses to calculate interval-level estimates that represent person locations (i.e., person ability or person achievement) and item locations (i.e., the difficulty to provide a correct or positive response) on a linear scale that represents the latent variable (the log-odds or "logit" scale). The difference between person and item locations can be used to calculate the probability for a correct or positive response ($x = 1$), rather than an incorrect or negative response ($x = 0$).

The equation for the dichotomous Rasch model can be expressed in log-odds form as follows:

$$\ln\left[\frac{\phi_{ni1}}{\phi_{ni0}}\right] = \theta_n - \delta_i \tag{2.1}$$

The Rasch model predicts the probability that person n on item i provides a correct or positive ($x = 1$) response, rather than an incorrect or negative ($x = 0$) response, given person locations (i.e., ability, achievement, θ_n) and item locations (i.e., difficulty, δ_i), as expressed on the logit scale.

DOI: 10.1201/9781003174660-2

Rasch Model Requirements

Estimates that are calculated using the dichotomous Rasch model can only be meaningfully interpreted if there is evidence that the data approximate the requirements for the model. Key among dichotomous Rasch model requirements are the following:

- *Unidimensionality*: A single latent variable is sufficient to explain most of the variation in item responses.
- *Local independence*: After controlling for the latent variable, there is no substantial association between the responses to individual items.
- *Person-invariant item estimates*: Item locations do not depend on (i.e., are independent from) the persons whose responses are used to estimate them.
- *Item-invariant person estimates*: Person locations do not depend on (i.e., are independent from) the items used to estimate them.

Evidence that data approximate these requirements provides support for the meaningful interpretation and use of item and person estimates on the logit scale as indicators of item and person locations on the latent variable. In practice, many analysts evaluate some or all of these requirements using various indicators of model-data fit for the facets in a Rasch model (in this case, items and persons). In the current chapter, we provide some basic code for calculating some popular residual-based fit indices for items and persons. We explore issues related to model requirements and evaluating model-data fit in more detail in Chapter 3.

2.1 Example Data: Transitive Reasoning Test

In this chapter, we will be working with data from a transitive reasoning test designed to measure students' ability to reason about the relationships among physical objects. The data were collected from a one-on-one interactive assessment in which an experimenter presented students with a set of objects, such as sticks, balls, cubes, and discs. The following description is given in Sijtsma and Molenaar (2002), pp. 31–32:

> The items for transitive reasoning had the following structure. A typical item used three sticks, here denoted A, B, and C, of different length, denoted Y, such that YA $<$ YB $<$ YC. The actual test taking had the form of a conversation between experimenter and child in which the sticks were identified by their colors rather than letters. First, sticks A and B were presented to a child, who was allowed to pick them up and compare their lengths, for example, by placing them next to each other on a table.

> Next, sticks B and C were presented and compared. Then all three sticks were displayed in a random order at large mutual distances so that their length differences were imperceptible, and the child was asked to infer the relation between sticks A and C from his or her knowledge of the relationship in the other two pairs.

The transitive reasoning items varied in terms of the property students were asked to reason about (length, weight, area). The tasks also varied in terms of the number of physical objects that students were asked to reason about, and whether the comparison tasks involved equalities, inequalities, or a mixture of equalities and inequalities. The characteristics of the transitive reasoning data are summarized in the following table:

Task	Property	Format	Objects	Measures
1	Length	YA > YB > YC	Sticks	12, 11.5, 11 (cm)
2	Length	YA = YB = YC = YD	Tubes	12 (cm)
3	Weight	YA > YB > YC	Tubes	45, 25, 18 (g)
4	Weight	YA = YB = YC = YD	Cubes	65 (g)
5	Weight	YA < YB < YC	Balls	40, 50, 70 (g)
6	Area	YA > YB> YC	Discs	2.5, 7, 6.5 (diameter; cm)
7	Length	YA > YB = YC	Sticks	28.5, 27.5, 27.5 (cm)
8	Weight	YA >YB = YC	Balls	65, 40, 40 (g)
9	Length	YA = YB = YC = YD	Sticks	12.5, 12.5, 13, 13 (cm)
10	Weight	YA = YB < YC = YD	Balls	60, 60, 100, 100 (g)

2.2 Dichotomous Rasch Model Analysis with CMLE in eRm

In the next section, we provide a step-by-step demonstration of a dichotomous Rasch model analysis using the *eRm* package (Mair et al., 2021), which uses CMLE. We encourage readers to use the example data set that is provided in the online supplement to conduct the analyses along with us.

Prepare for the Analyses

We selected eRm for the first illustration in the current chapter because it includes functions for applying the dichotomous Rasch model that are relatively straightforward to use and interpret. Please note that the eRm package uses the CMLE method to estimate Rasch model parameters. As a result, estimates from the eRm package are not directly comparable to estimates obtained using other estimation methods. Later in this chapter, we have included illustrations

of dichotomous Rasch model analyses with the TAM package (Robitzsch et al., 2021) with MMLE. We also provide an illustration with TAM using JMLE, which produces comparable estimates to JMLE estimates from some popular standalone Rasch software programs, such as Winsteps (Winsteps, 2020b) and Facets (Facets, 2020a).

To get started with the eRm package, install and load it into your R environment using the following code.

```
#install.packages("eRm")
library("eRm")
```

Now that we have installed and loaded the package to our R session, we are ready to import the data.

In this book, we use the function `read.csv()` to import data that are stored using comma separated values. We encourage readers to use their preferred method for importing data files into R or R Studio. Please note that if you use `read.csv()` you will need to first specify the file path to the location at which the data file is stored on your computer or set your working directory to the folder in which you have saved the data.

First, we will import the data using `read.csv()` and store it in an object called `transreas`.

```
transreas <- read.csv("transreas.csv")
```

Next, we will explore the data using descriptive statistics using the `summary()` function.

```
summary(transreas)
```

```
##      Student         Grade           task_01
##  Min.   :  1   Min.   :2.000   Min.   :0.0000
##  1st Qu.:107   1st Qu.:3.000   1st Qu.:1.0000
##  Median :213   Median :4.000   Median :1.0000
##  Mean   :213   Mean   :4.005   Mean   :0.9412
##  3rd Qu.:319   3rd Qu.:5.000   3rd Qu.:1.0000
##  Max.   :425   Max.   :6.000   Max.   :1.0000
##     task_02          task_03
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:1.0000   1st Qu.:1.0000
##  Median :1.0000   Median :1.0000
##  Mean   :0.8094   Mean   :0.8847
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
##     task_04          task_05
```

```
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:1.0000   1st Qu.:1.0000
##  Median :1.0000   Median :1.0000
##  Mean   :0.7835   Mean   :0.8024
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
##     task_06          task_07
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:1.0000   1st Qu.:1.0000
##  Median :1.0000   Median :1.0000
##  Mean   :0.9741   Mean   :0.8447
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
##     task_08          task_09          task_10
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00
##  1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:0.00
##  Median :1.0000   Median :0.0000   Median :1.00
##  Mean   :0.9671   Mean   :0.3012   Mean   :0.52
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.00
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00
```

From the summary of `transreas`, we can see there are no missing data. We can also get a general sense of the scales, range, and distribution of each variable in the data set.

Specifically, we can see that Student ID numbers range from 1 to 425, student grade levels range from 2 to 6, and all tasks have scores in both of the dichotomous categories (0 and 1). We can also get a sense for the range of item difficulty by examining the mean for each task, which is the proportion-correct statistic (item difficulty estimate for Classical Test Theory).

Run the Dichotomous Rasch Model

To run the dichotomous Rasch Model using the eRm package, need to isolate the item response matrix from the other variables in the data (student IDs and grade level). To do this, we will create an object made up of only the item responses by removing the first two variables from the data. We will remove the descriptive variables using the `subset()` function with the `select=` option. We will save the response matrix in a new object called `transreas.responses`.

```
transreas.responses <- subset(transreas,select=-c(
   Student,Grade))
```

Next, we will use `summary()` to calculate descriptive statistics for the `transreas.responses` object to check our work and ensure that the responses are ready for analysis.

```
summary(transreas.responses)
```

```
##      task_01          task_02
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:1.0000   1st Qu.:1.0000
##  Median :1.0000   Median :1.0000
##  Mean   :0.9412   Mean   :0.8094
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
##      task_03          task_04
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:1.0000   1st Qu.:1.0000
##  Median :1.0000   Median :1.0000
##  Mean   :0.8847   Mean   :0.7835
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
##      task_05          task_06
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:1.0000   1st Qu.:1.0000
##  Median :1.0000   Median :1.0000
##  Mean   :0.8024   Mean   :0.9741
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
##      task_07          task_08
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:1.0000   1st Qu.:1.0000
##  Median :1.0000   Median :1.0000
##  Mean   :0.8447   Mean   :0.9671
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
##      task_09          task_10
##  Min.   :0.0000   Min.   :0.00
##  1st Qu.:0.0000   1st Qu.:0.00
##  Median :0.0000   Median :1.00
##  Mean   :0.3012   Mean   :0.52
##  3rd Qu.:1.0000   3rd Qu.:1.00
##  Max.   :1.0000   Max.   :1.00
```

Now, we are ready to run the dichotomous Rasch model on the transitive reasoning response data. We will use the `RM()` function to run the model and store the results in an object called `dichot.transreas`.

```
dichot.transreas <- RM(transreas.responses)
```

Overall Model Summary

The next step is to request a summary of the model estimation results in order to begin to understand the results from the analysis. We can do so by applying the `summary()` function to the model object.

```
summary(dichot.transreas)
```

```
##
## Results of RM estimation:
##
## Call:  RM(X = transreas.responses)
##
## Conditional log-likelihood: -921.3465
## Number of iterations: 18
## Number of parameters: 9
##
## Item (Category) Difficulty Parameters (eta): with
   0.95 CI:
##          Estimate Std. Error lower CI upper CI
## task_02     0.258      0.133   -0.003    0.518
## task_03    -0.416      0.157   -0.723   -0.109
## task_04     0.441      0.128    0.190    0.692
## task_05     0.309      0.131    0.052    0.567
## task_06    -2.175      0.292   -2.747   -1.604
## task_07    -0.025      0.141   -0.302    0.252
## task_08    -1.909      0.262   -2.423   -1.395
## task_09     2.923      0.130    2.668    3.179
## task_10     1.836      0.115    1.610    2.062
##
## Item Easiness Parameters (beta) with 0.95 CI:
##               Estimate Std. Error lower CI
## beta task_01     1.243      0.204    0.842
## beta task_02    -0.258      0.133   -0.518
## beta task_03     0.416      0.157    0.109
## beta task_04    -0.441      0.128   -0.692
## beta task_05    -0.309      0.131   -0.567
## beta task_06     2.175      0.292    1.604
## beta task_07     0.025      0.141   -0.252
## beta task_08     1.909      0.262    1.395
## beta task_09    -2.923      0.130   -3.179
## beta task_10    -1.836      0.115   -2.062
##               upper CI
## beta task_01     1.643
## beta task_02     0.003
## beta task_03     0.723
```

```
## beta task_04    -0.190
## beta task_05    -0.052
## beta task_06     2.747
## beta task_07     0.302
## beta task_08     2.423
## beta task_09    -2.668
## beta task_10    -1.610
```

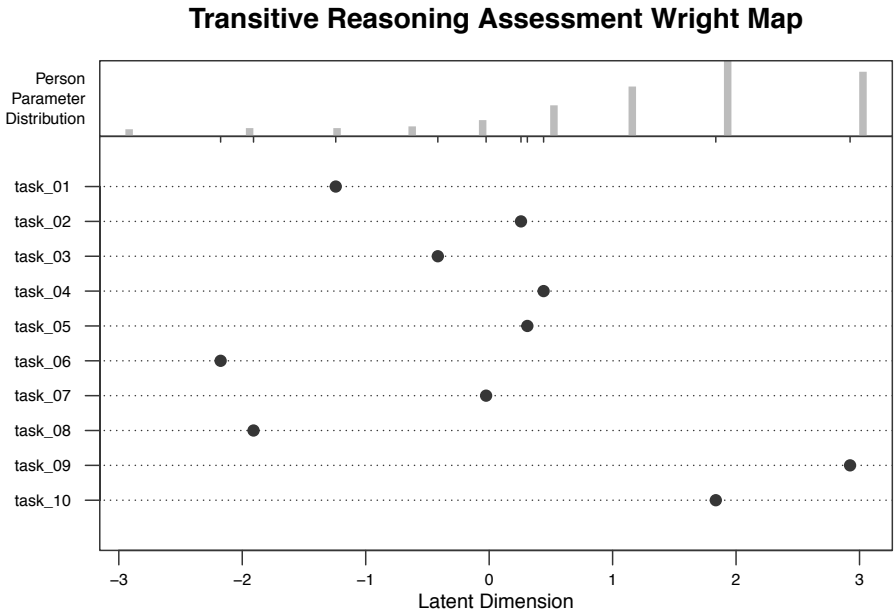
The summary of the dichotomous Rasch model output includes the Conditional Log-likelihood statistic, details about the number of iterations and model parameters, and a table with item parameters, their standard errors, and confidence intervals. It is important to note that the item parameters included in this preliminary output are *item easiness* parameters– *not* item difficulty parameters. We will examine item difficulty parameters in detail later in our analysis.

Wright Map

A useful feature of Rasch models is that when there is acceptable fit between the model and the data (discussed in detail in Chapter 3), it is possible to visualize and compare item and person locations on a single linear continuum. Professor Benjamin D. Wright popularized an approach to displaying Rasch model results on a linear continuum, and Rasch measurement researchers across disciplines have adopted this technique. In his honor, many researchers refer to these displays as *Wright Maps*. Researchers also refer to these displays as *Variable Maps*. Please see Wilson (2011) for a discussion of the term *Wright Map*.

As the next step in our analysis, we will create a Wright Map from our model results. We will create the plot using the function `PlotPImap()` on the model object (`dichot.transreas`). We will set the option for displaying threshold labels as `FALSE`, because we are working with dichotomous data. We also used the `main.title=`option to customize the title of the plot.

```
plotPImap(dichot.transreas, main = "Transitive
   Reasoning Assessment Wright Map")
```



In this *Wright Map* display, the results from the dichotomous Rasch model analysis of the Transitive Reasoning data are summarized graphically. The figure is organized as follows:

Starting at the bottom of the figure, the horizontal axis (labeled *Latent Dimension*) is the logit scale that represents the latent variable. In the application of the Transitive Reasoning data, lower numbers indicate less transitive reasoning ability, and higher numbers indicate more transitive reasoning ability.

The central panel of the figure shows item difficulty locations on the logit scale for the 10 transitive reasoning tasks that were included in the analysis; the y-axis for this panel shows the item labels. By default in eRm, the items are ordered according to their original order in the response matrix. The items can be ordered by difficulty by adding `sorted = TRUE` as an argument in the `plotPImap()` call. For each item, a solid circle plotting symbol shows the location estimate.

The upper panel of the figure shows a histogram of person (in this case, student) location estimates on the logit scale. Small vertical lines on the x-axis of this histogram show the points on the logit scale at which information (variance) is maximized for the sample of persons and items in the analysis. These lines can be omitted by adding `irug = FALSE` as an argument in the `plotPImap()` call.

Even though it is not appropriate to fully interpret item and person locations on the logit scale until there is evidence of acceptable model-data fit, we recommend examining the Wright Map during the preliminary stages of an

item analysis to get a general sense of the model results and to identify any potential scoring or data entry errors.

A quick glance at the Wright Map suggests that, on average, the persons (students) are located higher on the logit scale compared to the average item (task) locations. In addition, there appears to be a relatively wide spread of person and item locations on the logit scale, such that the transitive reasoning test appears to be a useful tool for identifying differences in person locations and item locations on the latent variable. We will return to this display for more exploration after we check for acceptable model-data fit, among other psychometric properties.

Item Parameters

As the next step in our analysis, we will examine the item parameters in detail. The eRm package provides several options with which analysts can find and examine item location parameters. For example, one way to obtain the overall item location parameters is to extract the *eta parameters* from the model object using the `$` operator. We extract these parameters, print them to the console, and calculate summary statistics for them with the following code.

```
difficulty <- dichot.transreas$etapar
difficulty
```

```
##      task_02      task_03      task_04      task_05
##  0.25775001 -0.41581384  0.44093780  0.30941151
##      task_06      task_07      task_08      task_09
## -2.17531698 -0.02481129 -1.90884851  2.92343872
##      task_10
##  1.83581566
```

Because of the nature of the estimation process used in eRm, the item.locations object that we just created does not include the location estimate for the first item. One can calculate the location for item 1 by subtracting the sum of the item locations from zero. In the following code, we find the location for item 1, and then create a new object with all 10 item locations.

```
n.items <- ncol(transreas.responses)
i1 <- 0 - sum(difficulty[1:(n.items - 1)])
difficulty.all <- c(i1, difficulty[c(1:(n.items - 1))
   ])
difficulty.all
```

```
##                   task_02      task_03      task_04
## -1.24256308  0.25775001 -0.41581384  0.44093780
##      task_05      task_06      task_07      task_08
##  0.30941151 -2.17531698 -0.02481129 -1.90884851
##      task_09      task_10
##  2.92343872  1.83581566
```

Alternatively, we could find the item difficulty parameters by extracting the item *easiness* parameters from the model object (*beta parameters*) and multiplying them by −1 to find item difficulty.

```
difficulty2 <- dichot.transreas$betapar * -1
difficulty2
```

```
## beta task_01 beta task_02 beta task_03
##  -1.24256308   0.25775001  -0.41581384
## beta task_04 beta task_05 beta task_06
##   0.44093780   0.30941151  -2.17531698
## beta task_07 beta task_08 beta task_09
##  -0.02481129  -1.90884851   2.92343872
## beta task_10
##   1.83581566
```

The item difficulty parameters (`xsi`) are the item location estimates on the logit scale that represents the latent variable. Assuming that the responses are scored such that lower scores ($x = 0$) indicate lower locations on the latent variable (e.g., incorrect, negative, or absent responses), *higher* item estimates on the logit scale indicate items that are *more difficult* or require persons to have relatively *higher locations* on the construct to provide a correct response. On the other hand, *lower* item estimates on the logit scale indicate items that are *easier* or require persons to have relatively *lower locations* on the construct to provide a correct response. In our analysis, Task 9 is the most difficult item ($\delta = 2.92$), whereas Task 6 is the easiest item ($\delta = -2.18$).

Next, we will examine item standard errors. The standard errors are reported for all of the items as part of the beta (item easiness) parameters.

```
dichot.transreas$se.beta
```

```
##  [1] 0.2043488 0.1327967 0.1566695 0.1282289
##  [5] 0.1314316 0.2916385 0.1413876 0.2622114
##  [9] 0.1302767 0.1152173
```

The standard error for each item is an estimate of the precision of the item estimates, where larger standard errors indicate less-precise estimates. Standard errors are reported on the same logit scale as item locations. In our analysis, the standard errors range from 0.11 for Task 9 and Task 10, which were the items with the most precise estimates, to 0.31 for Task 6, which was the item with the least precise estimate. These differences largely reflect differences in item targeting to the person locations (see the Wright Map above).

Next, let's calculate descriptive statistics to better understand the distribution of the item locations and standard errors. We will do so using the `summary()` function and the `sd()` function.

```
summary(difficulty.all)
```

```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -2.1753 -1.0359  0.1165  0.0000  0.4081  2.9234
```

```
sd(difficulty.all)
```

```
## [1] 1.576441
```

```
summary(dichot.transreas$se.beta)
```

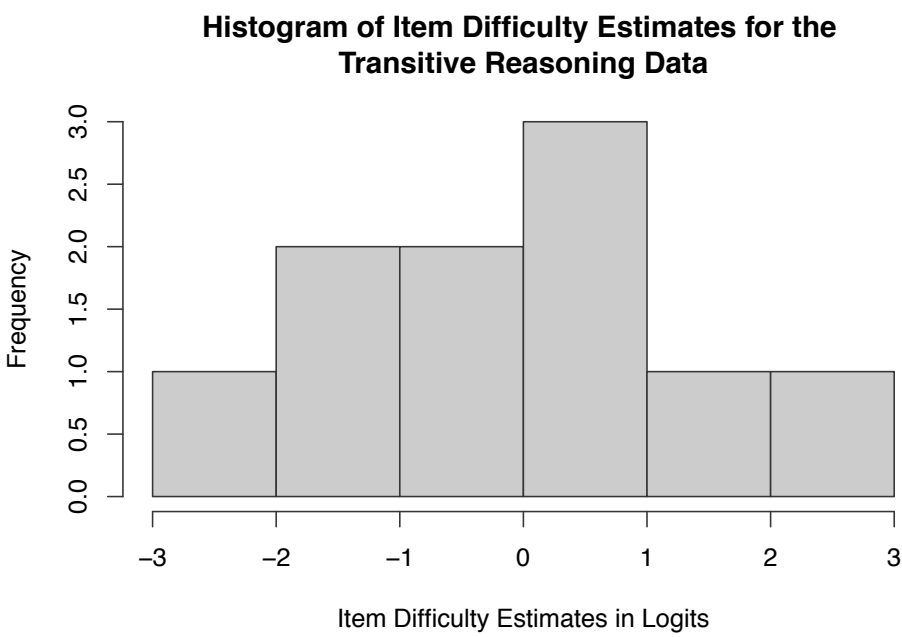
```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.1152  0.1306  0.1371  0.1694  0.1924  0.2916
```

```
sd(dichot.transreas$se.beta)
```

```
## [1] 0.06206381
```

We can also visualize the item difficulty estimates using a simple histogram by using the `hist()` function. In our plot, we specified a custom title using `main =` and a custom x-axis label using `xlab =`:

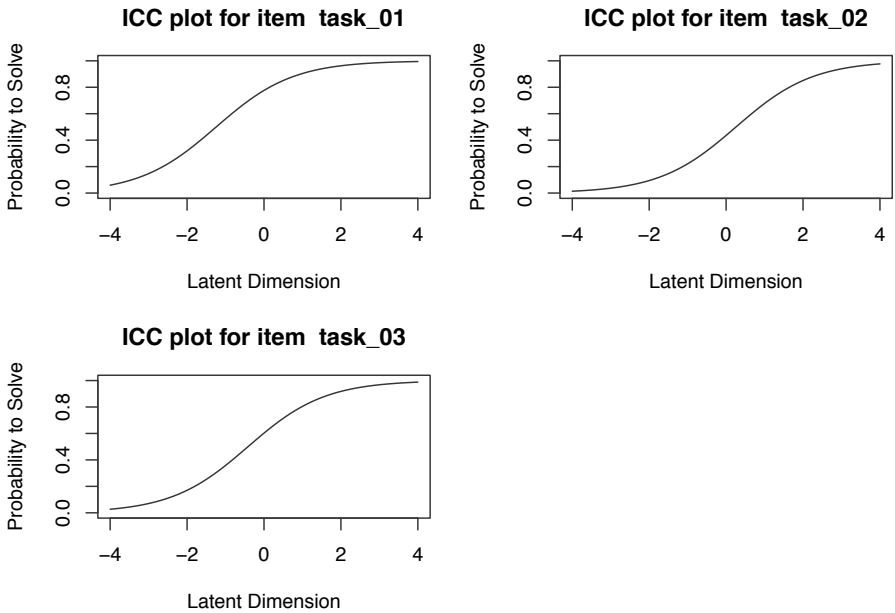
```
hist(difficulty.all, main = "Histogram of Item
    Difficulty Estimates for the \nTransitive
    Reasoning Data",
      xlab = "Item Difficulty Estimates Logits")
```



Item Response Functions

We will examine graphical displays of item difficulty using item response functions (IRFs). With the dichotomous Rasch model, the eRm package creates plots of the probability for a correct or positive response ($x = 1$), conditional on person locations on the latent variable. In the following code, we use `plotICC()` from eRm to create IRF plots for the items in our analysis. We included `ask = FALSE` in our function call to generate all of the plots at once. For brevity, we have only included plots for the first three items in this book. The specific items to be plotted can be controlled by changing the items included in the `items.to.plot` object.

```
items.to.plot <- c(1:3)
plotICC(dichot.transreas, ask = FALSE, item.subset =
    items.to.plot)
```



This code generates IRFs for the first three items in our analysis. In each item-specific plot, the x-axis is the logit scale that represents the latent variable; this scale represents transitive reasoning ability in our example. The y-axis is the probability for a correct or positive response, conditional on person locations on the latent variable.

Person Parameters and Item Fit

In the eRm package, it is necessary to calculate person parameters *before* item fit statistics can be calculated. Accordingly, we will proceed with a brief examination of person parameters before we conduct item fit analyses. In

practice, we recommend examining item fit before examining and interpreting item nad person locations in detail.

Person Parameters

In the following code, we use the `person.parameter()` function to estimate student locations and save the results in an object called `student.locations`. Then, we save the theta estimates (i.e., achievement estimates) in a data.frame object called `achievement`, and add student identification numbers for reference.

```
student.locations <- person.parameter(dichot.
   transreas)
achievement <- student.locations$theta.table
achievement$id <- rownames(achievement)
```

The estimation procedure in eRm does not directly produce parameter estimates for persons with extreme scores. In our example, 51 students earned extreme scores on the transitive reasoning assessment, so achievement estimates are reported for the remaining 374 students with non-extreme scores. The achievement estimates for the 51 students with extreme scores are extrapolated, and standard errors are not calculated.

We can add standard errors for the student achievement estimates to our `achievement` object as follows.

```
se <- as.data.frame(student.locations$se.theta$
   NAgroup1)
se$id <- rownames(se)
names(se) <- c("person_se", "id")
achievement.with.se <- merge(achievement, se, by = "
   id" )
```

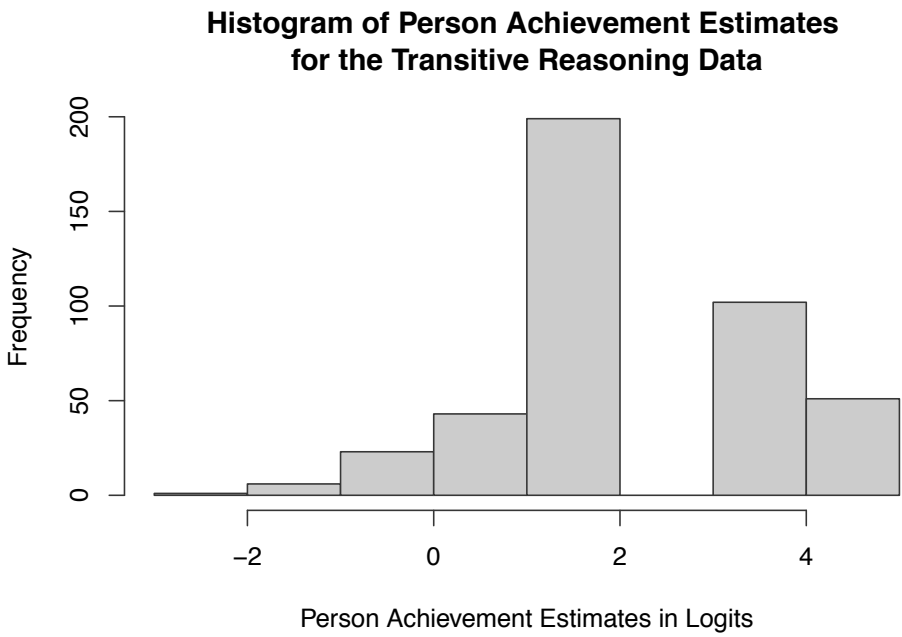
As a final step in examining the person parameters, we will calculate descriptive statistics and use a histogram to examine the distribution of student locations on the logit scale.

```
summary(achievement)
```

```
##  Person Parameter    NAgroup  Interpolated
##  Min.   :-2.916   Min.   :1   Mode :logical
##  1st Qu.: 1.160   1st Qu.:1   FALSE:374
##  Median : 1.932   Median :1   TRUE :51
##  Mean   : 2.010   Mean   :1
##  3rd Qu.: 3.028   3rd Qu.:1
##  Max.   : 4.201   Max.   :1
##       id
##  Length:425
##  Class :character
```

```
##   Mode   :character
##
##
##
```

```
hist(achievement$`Person Parameter`, main = "
    Histogram of Person Achievement Estimates \nfor
    the Transitive Reasoning Data",
      xlab = "Person Achievement Estimates Logits")
```



Item Fit

Next, we will conduct a brief exploration of item fit statistics. We explore item fit in more detail in Chapter 3.

To calculate numeric item fit statistics, we will use the function `itemfit()` from eRm on the person parameter object (`person.locations.estimate`). This function produces several item fit statistics, including infit mean square error (MSE), outfit MSE, and standardized infit and outfit MSE statistics. We will store the item fit results in a new object called `item.fit`, and then format this object as a data.frame object for easy manipulation and exporting.

```
student.locations <- person.parameter(dichot.
    transreas)
item.fit <- itemfit(student.locations)
item.fit
```

```
##
## Itemfit Statistics:
##            Chisq  df p-value Outfit MSQ Infit MSQ
## task_01 245.550 373   1.000       0.657      0.761
## task_02 421.904 373   0.041       1.128      1.087
## task_03 218.155 373   1.000       0.583      0.745
## task_04 478.894 373   0.000       1.280      1.158
## task_05 346.015 373   0.839       0.925      0.991
## task_06  65.613 373   1.000       0.175      0.611
## task_07 244.494 373   1.000       0.654      0.804
## task_08 125.478 373   1.000       0.336      0.687
## task_09 394.207 373   0.216       1.054      0.904
## task_10 326.534 373   0.960       0.873      0.898
##         Outfit t Infit t Discrim
## task_01   -1.327  -1.622   0.455
## task_02    1.071   1.187   0.054
## task_03   -2.742  -2.688   0.583
## task_04    2.442   2.268  -0.014
## task_05   -0.625  -0.109   0.184
## task_06   -2.789  -1.781   0.620
## task_07   -2.768  -2.475   0.493
## task_08   -2.205  -1.569   0.563
## task_09    0.462  -1.727   0.022
## task_10   -1.822  -2.408   0.226
```

Next, we will request a summary of the numeric fit statistics using the `summary()` function.

```
summary(item.fit$i.infitMSQ)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.6114  0.7490  0.8506  0.8645  0.9689  1.1575
```

```
summary(item.fit$i.infitZ)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -2.6885 -2.2509 -1.6743 -1.0924 -0.4742  2.2680
```

```
summary(item.fit$i.outfitMSQ)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1754  0.6009  0.7648  0.7665  1.0218  1.2805
```

```
summary(item.fit$i.outfitZ)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -2.7885 -2.6080 -1.5748 -1.0304  0.1904  2.4418
```

The `item.fit` object includes mean square error (MSE) and standardized (Z) versions of the outfit and infit statistics for each item included in the analysis. These statistics are summaries of the residuals associated with each item. When data fit Rasch model expectations, the MSE versions of outfit and infit are expected to be close to 1.00 and the standardized versions of outfit and infit are expected to be around 0.00. Please refer to Chapter 3 for a more-detailed discussion of item fit.

As a final step in examining item fit, we will calculate the *reliability of separation statistic for items* using the following procedure.

```
# Get Item scores
ItemScores <- colSums(transreas.responses)

# Get Item Standard Deviation (SD)
ItemSD <- apply(transreas.responses,2,sd)

# Calculate the Standard Error (SE) for the Items
ItemSE <- ItemSD/sqrt(length(ItemSD))

# Calculate the Observed Variance (also known as
    Total Person Variability or Squared Standard
    Deviation)
SSD.ItemScores <- var(ItemScores)

# Calculate the Mean Square Measurement error (also
    known as Model Error variance)
Item.MSE <- sum((ItemSE)^2) / length(ItemSE)

# Calculate the Item Separation Reliability
item.separation.reliability <- (SSD.ItemScores-Item.
    MSE) / SSD.ItemScores
item.separation.reliability

## [1] 0.9999984
```

Briefly, the *item reliability of separation statistic* describes the degree to which items have unique locations on the logit-scale. Please refer to Chapter 3 for more details about item reliability analysis.

Person Fit

Next, we will conduct a brief exploration of person fit. To calculate numeric person fit statistics, we will use the function `personfit()` from eRm on the person parameter object (`person.locations.estimate`). This function produces several person fit statistics, including infit MSE, outfit MSE, and standardized infit and outfit MSE statistics. We will store the person fit

results in a new object called `person.fit`, and then format this object as a data.frame for easy manipulation and exporting. Then, we request summaries of the infit MSE and outfit MSE statistics using `summary()`.

```
person.fit <- personfit(student.locations)
summary(person.fit$p.infitMSQ)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4060  0.4910  0.7487  0.9102  1.1740  2.2483
```

```
summary(person.fit$p.outfitMSQ)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1723  0.2399  0.5468  0.7665  0.9592  7.3060
```

We can calculate the *reliability of person separation statistic* using eRm. This value is interpreted similarly to Cronbach's alpha (Cronbach, 1951) when there is good fit between the data and the Rasch model (Andrich, 1982). However, it is important to note that these coefficients are not equivalent because alpha is based on an assumption of linearity and the Rasch reliability of separation statistic is based on a linear, interval-level scale when there is evidence of good model-data fit (discussed in Chapter 3).

```
person_rel <- SepRel(student.locations)
person_rel$sep.rel
```

```
## [1] 0.2061478
```

Summarize the Results in Tables

As a final step, we will create tables that summarize the calibrations of the items and persons from our dichotomous Rasch model analysis with eRm.

Table 1 is an overall model summary table that provides an overview of the logit-scale locations, standard errors, fit statistics, and reliability statistics for items and persons. This type of table is useful for reporting the results from Rasch model analyses because it provides a concise overview of the location estimates and numeric model-data fit statistics for the items and persons in the analysis.

```
summary.table.statistics <-
  c("Logit Scale Location Mean",
  "Logit Scale Location SD",
  "Standard Error Mean",
  "Standard Error SD",
  "Outfit MSE Mean",
  "Outfit MSE SD",
  "Infit MSE Mean",
```

```
  "Infit MSE SD",
  "Std. Outfit Mean",
  "Std. Outfit SD",
  "Std. Infit Mean",
  "Std. Infit SD",
  "Reliability of Separation")

item.summary.results <-
  rbind(mean(difficulty.all),
        sd(difficulty.all),
        mean(dichot.transreas$se.beta),
        sd(dichot.transreas$se.beta),
        mean(item.fit$i.outfitMSQ),
        sd(item.fit$i.outfitMSQ),
        mean(item.fit$i.infitMSQ),
        sd(item.fit$i.infitMSQ),
        mean(item.fit$i.outfitZ),
        sd(item.fit$i.outfitZ),
        mean(item.fit$i.infitMSQ),
        sd(item.fit$i.infitZ),
        item.separation.reliability)


person.summary.results <-
  rbind(mean(achievement.with.se$`Person Parameter`),
        sd(achievement.with.se$`Person Parameter`),
        mean(achievement.with.se$person_se, na.rm =
           TRUE),
        sd(achievement.with.se$person_se, na.rm =
           TRUE),
        mean(person.fit$p.outfitMSQ),
        sd(person.fit$p.outfitMSQ),
        mean(person.fit$p.infitMSQ),
        sd(person.fit$p.infitMSQ),
        mean(person.fit$p.outfitZ),
        sd(person.fit$p.outfitZ),
        mean(person.fit$p.infitZ),
        sd(person.fit$p.infitZ),
        person_rel$sep.rel)

# Round the values for presentation in a table:
item.summary.results_rounded <- round(item.summary.
   results, digits = 2)
```

```
person.summary.results_rounded <- round(person.
    summary.results, digits = 2)

Table1 <- cbind.data.frame(summary.table.statistics,
                           item.summary.results_
                              rounded,
                           person.summary.results_
                              rounded)

# Add descriptive column labels:
names(Table1) <- c("Statistic", "Items", "Persons")
print.data.frame(Table1, row.names = FALSE)
```

```
##                   Statistic Items Persons
##  Logit Scale Location Mean  0.00    1.71
##    Logit Scale Location SD  1.58    1.09
##        Standard Error Mean  0.17    0.95
##          Standard Error SD  0.06    0.16
##            Outfit MSE Mean  0.77    0.77
##              Outfit MSE SD  0.35    0.80
##             Infit MSE Mean  0.86    0.91
##               Infit MSE SD  0.18    0.47
##           Std. Outfit Mean -1.03    0.08
##             Std. Outfit SD  1.83    0.64
##            Std. Infit Mean  0.86   -0.12
##              Std. Infit SD  1.67    0.91
##  Reliability of Separation  1.00    0.21
```

Table 2 summarizes the calibrations of individual items. For data sets with manageable item sample sizes such as the transitive reasoning data example in this chapter, we recommend reporting details about each item in a table similar to this one.

```
# Calculate the proportion correct for each task:
PropCorrect <- apply(transreas.responses, 2, mean)

# Combine item calibration results in a table:
Table2 <-
  cbind.data.frame(colnames(transreas.responses),
                   PropCorrect,
                   difficulty.all,
                   dichot.transreas$se.beta,
                   item.fit$i.outfitMSQ,
                   item.fit$i.outfitZ,
                   item.fit$i.infitMSQ,
                   item.fit$i.infitZ)
```

```
names(Table2) <- c("Task ID", "Proportion Correct", "
    Item Location","Item SE","Outfit MSE","Std. Outfit
    ", "Infit MSE","Std. Infit")

# Sort Table 2 by Item difficulty:
Table2 <- Table2[order(-Table2$`Item Location`),]

# Round the numeric values (all columns except the
    first one) to 2 digits:
Table2[, -1] <- round(Table2[,-1], digits = 2)
print.data.frame(Table2, row.names = FALSE)

##  Task ID Proportion Correct Item Location Item SE
##  task_09               0.30          2.92    0.13
##  task_10               0.52          1.84    0.12
##  task_04               0.78          0.44    0.13
##  task_05               0.80          0.31    0.13
##  task_02               0.81          0.26    0.13
##  task_07               0.84         -0.02    0.14
##  task_03               0.88         -0.42    0.16
##  task_01               0.94         -1.24    0.20
##  task_08               0.97         -1.91    0.26
##  task_06               0.97         -2.18    0.29
##  Outfit MSE Std. Outfit Infit MSE Std. Infit
##        1.05         0.46      0.90       -1.73
##        0.87        -1.82      0.90       -2.41
##        1.28         2.44      1.16        2.27
##        0.93        -0.62      0.99       -0.11
##        1.13         1.07      1.09        1.19
##        0.65        -2.77      0.80       -2.47
##        0.58        -2.74      0.75       -2.69
##        0.66        -1.33      0.76       -1.62
##        0.34        -2.21      0.69       -1.57
##        0.18        -2.79      0.61       -1.78
```

Finally, Table 3 provides a summary of the person calibrations. When there is a relatively large person sample size, it may be more useful to present the results as they relate to individual persons or subsets of the person sample as they are relevant to the purpose of the analysis. In the following code, we create Table 3 and print the first 6 lines to the console.

```
# Calculate proportion correct for persons:
PersonPropCorrect <- apply(student.locations$X.ex, 1,
    mean)
```

```
# Combine person calibration results in a table:
Table3 <-
  cbind.data.frame(achievement.with.se$id,
                   PersonPropCorrect,
                   achievement.with.se$`Person
                      Parameter`,
                   achievement.with.se$person_se,
                   person.fit$p.outfitMSQ,
                   person.fit$p.outfitZ,
                   person.fit$p.infitMSQ,
                   person.fit$p.infitZ)

names(Table3) <- c("Person ID", "Proportion Correct",
    "Person Location","Person SE","Outfit MSE","Std.
   Outfit", "Infit MSE","Std. Infit")

# Round the numeric values (all columns except the
   first one) to 2 digits:
Table3[, -1] <- round(Table3[,-1], digits = 2)
print.data.frame(Table3[1:6, ], row.names = FALSE)

##  Person ID Proportion Correct Person Location
##          1                0.8            1.93
##         10                0.6            1.16
##        100                0.4            1.16
##        101                0.7            3.03
##        102                0.9            0.52
##        103                0.7            3.03
##  Person SE Outfit MSE Std. Outfit Infit MSE
##       0.94       0.24       -0.52      0.41
##       0.83       1.10        0.36      1.13
##       0.83       5.62        3.50      2.10
##       1.19       0.56       -0.37      0.71
##       0.77       0.17        0.12      0.49
##       1.19       0.73       -0.09      0.80
##  Std. Infit
##       -1.19
##        0.47
##        2.55
##       -0.57
##       -0.61
##       -0.34
```