# Getting Started with the Graded Response Model (GRM): A gentle introduction and tutorial in R

Only half of published findings include evidence of measurement validity (Flake et al., 2017) and a large proportion of published studies suffer from hidden invalidity (Hussey & Hughes, 2020). Lack of evidence for measurement validity can raise serious concerns about the replicability of research findings, as it directly contributes to the invalidity of the findings (Flake & Fried, 2020). Furthermore, the lack of evidence for measurement validity can hamper consensus on the best way to measure a construct, which becomes a major obstacle to the establishment of cumulative psychological science (Elson et al., 2023). Given these circumstances, it is critical to provide applied psychological researchers with a set of tools and practical steps to help them assess measurement validity so that they can ensure the credibility of their findings.

The purpose of this article is to provide a brief introduction to the Graded Response Model (GRM), a family of item response models specifically designed to assess the measurement precision of a polytomous (Likert-style) scale. In this paper, we aim to keep our tutorial concise, so we do not cover all IRT concepts in detail. Interested readers are referred to available didactic texts on IRT, such as Embretson and Reise (2000) for the introductory level, and Baker and Seock-Ho (2017) and de Ayala (2022) for a more technical and comprehensive overview.

First, we briefly explain what GRM is and how it is specified. In this part, we highlight how model and item parameters are specified and interpreted as well as visualizations derived from a GRM analysis. We also introduce three key assumptions underlying a GRM and how to test these assumptions. We then provide an illustrative example of testing measurement precision using a GRM analysis, and in this part, we show how to practically implement a GRM analysis and interpret its results. We complement the illustrative example with code in the open-source programming language R (R Core Team, 2023) so that the readers can implement and reproduce the example presented in this paper. Readers with some experience in R and RStudio can easily follow this tutorial, but for those who are not yet familiar with R, we refer them to excellent, easy-to-follow materials developed by Navarro (2018). Finally, we present some concluding remarks to encourage researchers to implement a GRM analysis as a part of their routine.

# Graded Response Model

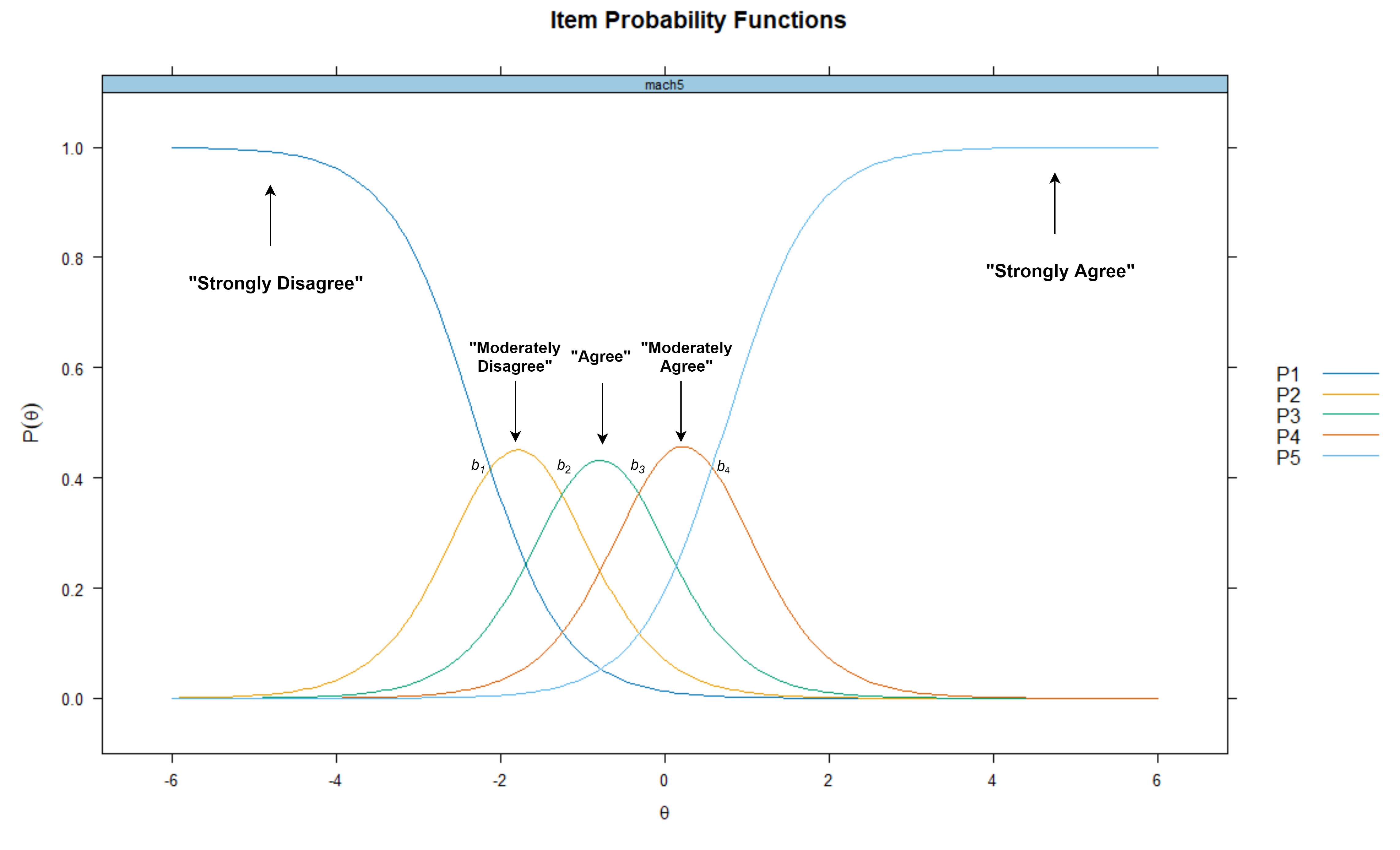
GRM is a family of IRT models specifically designed to analyze ordered polytomous (Likert-style) data (Samejima, 1997, 2010, 2016). The fundamental idea of GRM is to extend the logic of simpler dichotomous IRT models to polytomous scales by applying a probabilistic function to each response category. In dichotomous two-parameter logistic (2PL) IRT models, the probability of answering a correct response is modeled as a function of , item difficulty (*b*), and item discrimination (*a*). GRM extends this idea by modeling the probability of selecting certain response category or higher on an item (i.e., *step function*) given to item difficulty (*b*) and discrimination (*a*). For each response category (e.g., answering “agree” on a five-point Likert scale), GRM calculates the cumulative probability of a participant answering “agree” or above, given to their . Therefore, the notion of item difficulty in dichotomous IRT models is extended to the step function (i.e., item threshold - *b*) to handle ordered polytomous data.

To illustrate the step function, consider a scale measuring sadistic personality (e.g., “*watching a fist-fight excites me*,” etc.) with response categories ranging from 1 (*strongly disagree*) to 4 (*strongly agree*). Imagine these categories are hierarchically ordered from the lowest category (*strongly disagree*) to the highest (*strongly agree*), like a staircase. GRM aims to estimate the level of sadistic personality () required for “stepping” from one response category to another (e.g., *moderately agree* to *strongly agree*). Therefore, a GRM model of a five-point Likert scale calculates four item threshold parameters (*b*): the location of level where individuals are equally likely to respond 1 or 2 (*b*1), 2 or 3 (*b*2), 3 or 4 (*b*3), and 4 or 5 (*b*4). Each threshold is exactly the point where a participant is equally likely to respond to either of the adjacent response categories. For example, if *b*1 = -1.23 for the item “*watching a fist-fight excites me*”, it implies that participants with sadism level 1.23 below the mean are equally likely to either answer *strongly disagree* or *moderately disagree*.

Figure 1 shows an example of an item probability function (IPF) from an item on a scale with five response categories. In general, IPF describes a relationship between (x-axis) and the probability of endorsing a response category (y-axis - *P*). Figure 1 consists of five category probability curves, each of which represents the probability of endorsing a response category given . Item threshold is exactly the location of where two adjacent category probability curves cross each other.

Figure 1

An Item Probability Function from a GRM Model



## Assumptions

The first key assumption of a GRM is that is monotonically correlated with the probability of endorsing a response category. More sharply, individuals with higher levels of should have a higher probability of endorsing response category that represents greater intensity of the traits (Hambleton et al., 2010). For instance, people with a higher tendency of sadistic personality have a greater chance of answering “*moderately agree*” or “*strongly agree*” on a sadism scale compared to those with lower levels of sadism.

The second key assumption pertains to the unidimensionality of . This implies that the item pool tested using a GRM analysis should represent only one latent trait (). Although it is indeed possible to account for multiple in an IRT model (multidimensional IRT) we limit the scope of our tutorial to the unidimensional GRM. A unidimensional IRT model can still be robustly applied to multidimensional data if multiple latent traits are moderately intercorrelated or when there is a strong general factor (*g*) underlying the data (Reise et al., 2015). To test whether an item pool represents only one , researchers can apply a parallel analysis or a factor analysis, before specifying their GRM models.

Relatedly, a GRM model assumes that the item pool is locally independent, as the third assumption. Local independence suggests that participants’ responses to one item do not influence their response to another. It is important to note that evidence for unidimensionality does not always warrant local independence. As an eyeball example - consider a scale measuring trust in science, which contains: “*I trust scientists working in natural science*” and “*Most biologists are trustworthy*.” While both items might measure the same latent trait (i.e., trust in science), those who trust scientists working in natural science are very likely to also trust biologists. To test whether items are locally independent, researchers can examine the relationships between item responses after accounting for by performing residual correlation analysis (Chen & Thissen, 1997).

# An Illustrative Example of Graded Response Model: The Right Wing Authoritarianism (RWA) Scale

## A Brief Overview of the Altemeyer’s RWA Scale

To illustrate the application of GRM for analyzing measurement precision of an ordered polytomous scale, we provide an example of a GRM analysis of the RWA scale data (Altemeyer, 2006). Altemeyer (2006) defines RWA as a personality propensity to blindly abide by established authorities (i.e., authoritarian submission), to act aggressively toward individuals who are perceived to be punished by these established authorities (i.e., authoritarian aggression), and to uphold traditional values promoted by the authorities (i.e., conventionalism). Individuals with higher levels of RWA tend to view the world as a dangerous place (Duckitt & Sibley, 2017), and thus, to ensure stability and safety, they are motivated to preserve social order by deferring to those whom they perceive as legal, social, or moral authorities; for instance, the government, religious institutions, or political or religious or military leaders (Saunders & Ngo, 2017). The RWA scale remains one of the most important measures in the field of social and political psychology, and has helped researchers to conceptualize the role of RWA in shaping various social psychological outcomes, such as prejudice, political behavior, and various antisocial behaviors (Osborne et al., 2023).

In this paper, we use the most recent version of the RWA scale (Altemeyer, 2006), which consists of 22 items in which participants are asked to indicate their agreement with the items on a nine-point scale ranging from “*strongly disagree*” (-4) to “*strongly agree*” (+4). While the RWA scale consists of three sub-dimensions, i.e., Submissiveness, Aggression, and Conventionalism, the RWA scale is theoretically assumed to be unidimensional because (Altemeyer, 2006) shows that these sub-dimensions are strongly intercorrelated. While there is some evidence to the contrary (Duckitt & Bizumic, 2013), for the purposes of this article we will assume that the RWA scale is unidimensional.

# Disclosure and Data Availability Statements

To maximize the reproducibility of our illustrative example, we wrote the article as a Quarto (.qmd) document, where we integrate the R codes used in the analysis as well as its outputs. We also include a longer, annotated R script file (.R) for the example we used as a supplementary document, which we highly recommend to use for a didactic purpose. The Quarto file (and its corresponding .docx and .pdf output) and R script file are publicly available on a Github repository. The dataset we used in this tutorial paper was obtained from a publicly accessible [Open-Source Psychometrics Project](http://openpsychometrics.org/_rawdata/).

## Step 1: Data Preparation

Before running the analysis, we need to install the necessary R packages, which are, *mirt* (Chalmers, 2012; Chalmers et al., 2023), *psych* (Revelle, 2023), *ggmirt* (Masur, 2023), and *caret* (Kuhn, 2024) with these following commands:

install.packages("tidyverse", # for data wrangling  
 "psych", # for descriptive statistics and unidimensionality test  
 "devtools", # for installing R package that is not available on CRAN  
 "mirt", # for conducting the main GRM analysis  
 "caret", # for helping us detecting large residuals correlation (local independence test)  
 dependencies=TRUE)  
devtools::install\_github("masurp/ggmirt") # remote ggmirt installation through GitHub repository

The above command also automatically requests to install dependency packages. Note that we install *ggmirt* directly from its GitHub repository because *ggmirt* is not yet available in the Comprehensive R Archive Network (CRAN). Next, we should activate the R packages by typing a command as follows:

library(mirt); library(ggmirt); library(psych); library(tidyverse); library(caret)

Now after the packages are activated, we need to import the dataset to our R environment. The dataset (data.csv) and codebook (codebook.txt) are available in data folder in our repository. Readers can download them from our repository and then simply import the dataset with the following command:

ds <- read.csv("data/data.csv")

However, since the dataset is openly accessible in the Open Psychometrics webpage, readers can also directly import the dataset from the Open Psychometrics webpage to their R environment. For interested readers, we show how to do this in detail in our annotated R script file.

Next, it is important to examine the structure of the data frame (ds) to get a glimpse of its structure. By typing str() command, readers can examine the variables’ name, number of cases, types of variables (integer or character), and the values of each variable. Then readers can match the information provided in their R console to the codebook.

str(ds)

According to the information provided in the codebook, the responses to the RWA scale are denoted in Q1 to Q22, each representing the score of each item of the RWA scale. Readers can also see the information of Q1 to Q22 variables in their R console: these variables are integer and have values ranging from 1 to 9. In the codebook, there is no information regarding what these values (i.e., 1-9) mean. However, we could reasonably assume that participants who chose “*strongly disagree*” (-4) were scored 1, and those who opted for “*strongly agree*” (+4) were scored 9.

Please note that we are only interested in the responses of the RWA scale, so we should subset the dataset to contain only the variables we are interested in, and then, examine the structure again. To this end, readers can type these commands:

rwa <- subset(ds, select = Q1:Q22)  
str(rwa)

Here, we have a new data frame namely rwa, which contains only the responses to the RWA scale (Q1:Q22). In the data frame rwa, we have 22 variables with 9,881 participants. Before proceeding to the analysis stage, we recommend readers to inspect descriptive statistics, which we explain the procedure in the next sub-section.

## Step 2: Inspecting Key Descriptive Statistics

To efficiently examine descriptive statistics, readers can make use of describe() function from *psych* package by typing a simple command:

describe(rwa)

As readers can see in their R console, all variables have a minimum value of 0, while the minimum score should be 1. Again, there is no information provided in the codebook regarding what 0 actually represents. Therefore, before deciding what we should do next, it is better to find out how many 0 we have in the data frame. To this end, we can calculate the percentage of 0 in each item by running these commands:

zero <- colSums(rwa == 0) / nrow(rwa) \* 100 # Computing the percentage of "0" in each column.  
print(zero) # Showing the percentage of "0" for each item.

After executing these commands, readers can see the percentage of 0 in each item in their R console, which overall are very small, ranging from 0.09% to 0.32% of the total cases. Since the percentages are very small, 0 is likely a code for a missing value. It is important to note that *mirt* package uses the Expected-Maximization (EM) algorithm to estimate model parameters, and the EM algorithm can handle missing values. If we decide to eliminate cases with missing values, we still have a sizable number of cases, nonetheless. Therefore, to simplify our analysis, we only keep cases with complete responses by executing these commands:

rwa <- rwa %>%  
 mutate\_all(~na\_if(., 0)) %>% # Recoding "0" with "NA" in all columns.  
 drop\_na() # Removing cases with any "NA" values.

If readers prefer to keep cases with missing values, especially when the sample size is low, readers can exclude drop\_na() command.

Interestingly, if we look at mean scores of the items, they seem to differ drastically. Some items have high mean scores, ranging from 6 to 7, but some of those are comparatively low. The RWA scale indeed has some unfavorable items but, again, there is no information in the codebook whether these unfavorable items have been reversely coded. However, if we look closely at ten items with high mean ( > 6), these are the items that are supposed to be unfavorable. Therefore, we can reasonably assume that unfavorable items have not been reversed in the original dataset, and we should do so by running these commands:

unfav <- c("Q4","Q6","Q8","Q9","Q11","Q13","Q15","Q18","Q20","Q21") # Now we create a vector defining which items will be coded reversely.  
rwa <- rwa %>%   
 mutate(across(all\_of(unfav), ~ 10 - .))# We simply subtract the scores from 9 (the maximum) + 1 to reverse code the unfavorable items.

First, we define a vector namely unfav to identify which items should be reversed, and then we use mutate() function to reverse the score. This is simply done by subtracting the values of the unfavorable items from the maximum score plus one. We can examine descriptive statistics of our modified data frame by running describe() function again, and as readers can see in Table 1, now mean score across items does not drastically vary and has 1 as its minimum value.

Table 1

Descriptive Statistics of RWA Scale

| Item | Mean | SD | Minimum | Maximum | Range | n |
| --- | --- | --- | --- | --- | --- | --- |
| Q1 | 3.38 | 2.19 | 1 | 9 | 8 | 9,680 |
| Q2 | 2.10 | 2.08 | 1 | 9 | 8 | 9,680 |
| Q3 | 2.73 | 2.54 | 1 | 9 | 8 | 9,680 |
| Q4 | 7.57 | 2.38 | 1 | 9 | 8 | 9,680 |
| Q5 | 2.96 | 2.12 | 1 | 9 | 8 | 9,680 |
| Q6 | 7.46 | 2.28 | 1 | 9 | 8 | 9,680 |
| Q7 | 2.87 | 2.54 | 1 | 9 | 8 | 9,680 |
| Q8 | 6.69 | 2.41 | 1 | 9 | 8 | 9,680 |
| Q9 | 7.53 | 1.99 | 1 | 9 | 8 | 9,680 |
| Q10 | 3.20 | 2.63 | 1 | 9 | 8 | 9,680 |
| Q11 | 7.67 | 2.03 | 1 | 9 | 8 | 9,680 |
| Q12 | 3.61 | 2.46 | 1 | 9 | 8 | 9,680 |
| Q13 | 7.05 | 2.47 | 1 | 9 | 8 | 9,680 |
| Q14 | 3.55 | 2.74 | 1 | 9 | 8 | 9,680 |
| Q15 | 6.85 | 2.31 | 1 | 9 | 8 | 9,680 |
| Q16 | 2.14 | 2.09 | 1 | 9 | 8 | 9,680 |
| Q17 | 3.00 | 2.55 | 1 | 9 | 8 | 9,680 |
| Q18 | 7.87 | 2.09 | 1 | 9 | 8 | 9,680 |
| Q19 | 2.80 | 2.31 | 1 | 9 | 8 | 9,680 |
| Q20 | 7.56 | 2.04 | 1 | 9 | 8 | 9,680 |
| Q21 | 6.28 | 2.67 | 1 | 9 | 8 | 9,680 |
| Q22 | 2.90 | 2.50 | 1 | 9 | 8 | 9,680 |

## Step 3: Examining Dimensionality

Next, we show the readers the ways of testing unidimensionality. To test the number of latent factors of an item pool, there are three alternatives, which are multidimensional scaling (MDS), EFA, and CFA. In this tutorial, we demonstrate the use of EFA, and *psych* package provides a very efficient way to do this by calling this command:

irt.fa(rwa, nfactors = 1, fm = "minres")

irt.fa function is to run an EFA with a polychoric correlation matrix as an input, not a Pearson’s correlation matrix, because our data is ordinal. The option fm = "minres" implies that we choose minimum residual as a factoring method due to its effectiveness in handling non-normally distributed data. Since we have more than eight response categories, we should instead run an EFA with Pearson’s correlation matrix as an input. Therefore, we first need to create a Pearson’s correlation matrix from our data frame, and then, run an EFA by calling these following commands:

cor <- cor(rwa, method = "pearson") # First, creating a Pearson's correlation matrix.  
efa <- fa(rwa, nfactors = 1, fm = "minres") # Now, running EFA.  
print(efa) # Print the results.

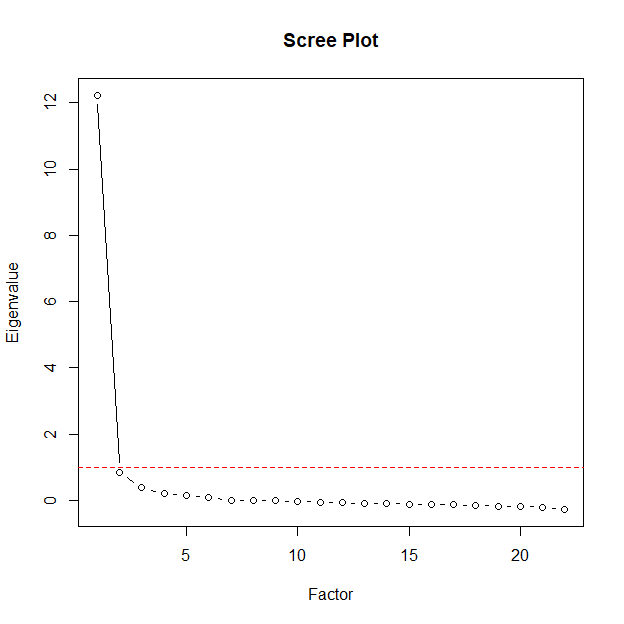
After calling the commands, readers can find Proportion Var in the output, which reflects the amount of variance explained by a latent factor. The variance proportion of our EFA model is 0.56, which means that 56% of the total variance is explained by a latent factor. Since the latent factor can account for more than 50% of the variance, we can reasonably assume that the latent factor underlying our data is unidimensional. To better visualize the Eigenvalues of the EFA, we can draw a scree plot using the following commands:

plot(efa$values, type = "b", main = "Scree Plot", xlab = "Factor", ylab = "Eigenvalue") # Scree plot  
abline(h = 1, col = "red", lty = 2) # Add new line to eigenvalue = 1.

The first line of code is a function to plot the Eigenvalues (efa$values, x-axis) given the number of latent factors (y-axis). The option type = "b" implies that we ask R to display the Eigenvalues exact points and a line connecting these points along the number of factors. The second line of code aims to display a horizontal red line (i.e., a reference line) to see which factors have eigenvalues greater than one, which is a common criterion for factor retention. Figure 2 shows that there is a clear break or an “elbow” that displays the location where the Eigenvalues start to level off. This implies that factors beyond the first may not be meaningful and could be only noise. Therefore, the scree plot strengthens our assumption that the item pool is, indeed, unidimensional.

Figure 2

Scree Plot



## Step 4: Model Estimation, Parameters, and Fit Statistics

In this step, we start to specify a vector representing our model, such as:

model <- 'rwa = 1-22'

This code implies that we want to estimate a model with namely rwa, and the items representing the are columns 1 to 22 of the data frame that we are going to use in the analysis. Then, we can continue to model fitting by typing the following command:

fit <- mirt(data = rwa, 1, model = model, itemtype = "graded", SE = TRUE, verbose = FALSE)

The code above implies that we are fitting an IRT model using data from a data frame namely rwa (data = rwa). The number 1 indicates that we assume that the model contains only one , i.e., unidimensional. We also set the model specification to model = model, which we have previously determined. We set itemtype to "graded" because we want to run a GRM model. The SE option is set to TRUE because we ask R to also estimate the standard error. At last, we set the verbose option to FALSE to avoid R displaying unnecessary information during the model fitting process, so the mirt() function can run quietly.

Next, we can ask the package to calculate item discrimination and thresholds by calling these functions below:

coefs <- coef(fit, IRTpars = TRUE, simplify = TRUE) # Saving model parameters (item discrimination (a) and threshold (b)) in a data frame.  
print(coefs) # Calling the data frame.  
summary(fit) # Displaying factor loadings and commonality.

The first line of code indicates that we are calling a coef() function to calculate item parameters from fit model, and then, store them in a new data frame called coefs. The IRTpars and simplify option is set to TRUE because we want to display the traditional IRT parameters (*a* and *b*) in a simple, readable format without showing the standard errors.

The output from this procedure is presented in Table 2, which shows item discrimination (*a*) of each item and item thresholds (*b1-b8*) of each response category within an item. To interpret *a*, readers can use a rule of thumb suggested by Baker and Seock-Ho (2017), which categorizes item discrimination exceeding 0.00, 0.35, 0.65, 1.35, and 1.70, as very low, low, moderate, high, and very high. As we see in {apatb-table4}, item discrimination parameters of the RWA scale range from high to very high (1.57 (Q1) to 3.32 (Q7)), suggesting efficient differentiation among individuals at different RWA levels. All threshold parameters show a consistent increase, implying that those with higher RWA levels tend towards a higher response categories. However, *b1* and *b2* for all items are very close to zero, which indicates that individuals with the RWA level around the mean are likely to opt for “*strongly disagree*” or stepping to its adjacent response category. This might suggest that the RWA scale is less sensitive to individuals with a low or very low RWA level.

The third line of code is a function to calculate a factor loading () and commonality (h2) of each item, which are also provided in Table 2. Before estimating a graded response model, *mirt* ran an EFA, and now as we can see in the console, we are looking at the EFA results. The results are slightly different from our previous EFA analysis, because *mirt* runs EFA using a quasi-polychoric correlation matrix, while the one we ran earlier to test unidimensionality used a Pearson’s correlation matrix as an input. However, most importantly, we see that all items are significantly loaded to one factor, and the factor now substantially accounts for 65.1% of the variance in the data, which strengthens our assumption that the RWA scale is unidimensional.

Table 2

Item Parameters

| Item | a | b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | λ | h2 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Q1 | 1.57 | -1.08 | -0.23 | 0.48 | 0.78 | 1.21 | 1.67 | 2.41 | 3.19 | 0.68 | 0.46 |
| Q2 | 2.45 | 0.47 | 0.94 | 1.19 | 1.33 | 1.58 | 1.86 | 2.14 | 2.47 | 0.82 | 0.67 |
| Q3 | 3.04 | 0.03 | 0.50 | 0.74 | 0.87 | 1.04 | 1.30 | 1.54 | 1.85 | 0.87 | 0.76 |
| Q4 | -2.50 | 2.10 | 1.82 | 1.53 | 1.30 | 1.08 | 0.98 | 0.74 | 0.22 | -0.83 | 0.68 |
| Q5 | 1.92 | -0.62 | 0.13 | 0.68 | 1.01 | 1.38 | 1.82 | 2.30 | 2.90 | 0.75 | 0.56 |
| Q6 | -1.92 | 2.52 | 2.13 | 1.81 | 1.53 | 1.17 | 0.98 | 0.62 | 0.05 | -0.75 | 0.56 |
| Q7 | 3.32 | -0.14 | 0.38 | 0.66 | 0.84 | 0.97 | 1.23 | 1.51 | 1.83 | 0.89 | 0.79 |
| Q8 | -1.58 | 2.44 | 2.04 | 1.64 | 1.28 | 0.70 | 0.47 | -0.06 | -0.74 | -0.68 | 0.46 |
| Q9 | -1.89 | 2.74 | 2.37 | 2.08 | 1.83 | 1.54 | 1.07 | 0.54 | -0.23 | -0.74 | 0.55 |
| Q10 | 3.30 | -0.30 | 0.20 | 0.47 | 0.63 | 0.82 | 1.13 | 1.40 | 1.75 | 0.89 | 0.79 |
| Q11 | -2.03 | 2.74 | 2.35 | 1.99 | 1.71 | 1.42 | 1.13 | 0.69 | 0.04 | -0.77 | 0.59 |
| Q12 | 2.61 | -0.77 | -0.20 | 0.25 | 0.46 | 0.79 | 1.19 | 1.60 | 2.02 | 0.84 | 0.70 |
| Q13 | -2.60 | 1.89 | 1.58 | 1.35 | 1.16 | 0.91 | 0.65 | 0.29 | -0.31 | -0.84 | 0.70 |
| Q14 | 2.30 | -0.46 | 0.04 | 0.32 | 0.47 | 0.75 | 1.09 | 1.42 | 1.79 | 0.80 | 0.65 |
| Q15 | -2.32 | 2.21 | 1.77 | 1.44 | 1.19 | 0.87 | 0.50 | 0.04 | -0.62 | -0.81 | 0.65 |
| Q16 | 2.59 | 0.43 | 0.85 | 1.13 | 1.30 | 1.54 | 1.82 | 2.11 | 2.42 | 0.84 | 0.70 |
| Q17 | 2.56 | -0.15 | 0.35 | 0.61 | 0.77 | 1.00 | 1.35 | 1.66 | 2.01 | 0.83 | 0.69 |
| Q18 | -2.32 | 2.45 | 2.14 | 1.87 | 1.61 | 1.44 | 1.26 | 0.98 | 0.40 | -0.81 | 0.65 |
| Q19 | 3.14 | -0.24 | 0.32 | 0.67 | 0.90 | 1.12 | 1.48 | 1.78 | 2.11 | 0.88 | 0.77 |
| Q20 | -1.75 | 2.89 | 2.46 | 2.07 | 1.74 | 1.52 | 1.14 | 0.61 | -0.14 | -0.72 | 0.51 |
| Q21 | -2.31 | 1.61 | 1.32 | 1.09 | 0.85 | 0.43 | 0.17 | -0.19 | -0.69 | -0.81 | 0.65 |
| Q22 | 3.07 | -0.16 | 0.35 | 0.63 | 0.82 | 1.02 | 1.32 | 1.59 | 1.88 | 0.87 | 0.76 |

*Note*. a = Discrimination, b1-b8 = Response specific threshold parameters, λ = Standardized Factor Loadings, h2 = Commonality.

Next, we can estimate the model goodness-of-fit (GOF) statistics (*M*2) by calling this command below:

M2(fit, type = "C2")

*mirt* calculates the *M*2 statistics (Maydeu-Olivares & Joe, 2006) and other GOF indices, such as Root Mean Square Error of Approximation (RMSEA), Comparative Fit Index (CFI), Tucker-Lewis Index (TLI), and Standardized Root-Mean-Square Residual (SRMSR) after running the code above. Here, we set the type option to "C2" because it is suitable for computing *M*2 statistics in polytomous models. The interpretation of GOF statistics is similar to a general guide for interpreting GOF in CFA or SEM models (Schermelleh-Engel et al., 2003), which are *RMSEA* 0.05, *SRMSR* < 0.05, *TLI* and *CFI* 0.97.

According to our analysis, the model overall does not fit the data well (*M*2(209) = 1.9808942^{4}, *p* < .05, *RMSEA* = 0.098, *SRMSR* = 0.055, TLI = 0.969, CFI = 0.972), because there is a significant discrepancy between response patterns predicted by the model with the data, as shown by significant *M*2 statistics. All GOF statistics, except for CFI, do not indicate a good fit. Model misfit, in this case, may be caused by local dependency between the items. We will explore this issue further in the next part.

Next, we calculate item-level fit statistics, i.e., the signed $\chi$2 statistics, by running the following function:

item.fit <- itemfit(fit) # Estimating item fit statistics and saving them in a data frame.  
print(item.fit) # Calling the data frame.

The output of this function is provided in Table 3, which shows fit statistics for an individual item. All items have very good (low) *RMSEA* values but normally, we want the *p* values of the signed $\chi$2 to be nonsignificant (*p* > .05), implying that there is no discrepancy between the observed response pattern with what the model predicts. In {apatb-table5}, only three items here are nonsignificant (Q2, Q4, and Q12), which strengthens our suspicion that the RWA items might not be locally independent.

Table 3

Item Fit Statistics

| Item | Sχ2 | df | RMSEA | p |
| --- | --- | --- | --- | --- |
| Q1 | 912.42 | 482 | 0.010 | 0.000 |
| Q2 | 1,726.93 | 447 | 0.017 | 0.000 |
| Q3 | 812.08 | 458 | 0.009 | 0.000 |
| Q4 | 1,477.36 | 526 | 0.014 | 0.000 |
| Q5 | 937.43 | 474 | 0.010 | 0.000 |
| Q6 | 1,037.64 | 552 | 0.010 | 0.000 |
| Q7 | 887.73 | 456 | 0.010 | 0.000 |
| Q8 | 1,159.88 | 526 | 0.011 | 0.000 |
| Q9 | 1,133.73 | 533 | 0.011 | 0.000 |
| Q10 | 1,424.25 | 472 | 0.014 | 0.000 |
| Q11 | 1,704.65 | 539 | 0.015 | 0.000 |
| Q12 | 1,710.42 | 472 | 0.016 | 0.000 |
| Q13 | 1,595.25 | 512 | 0.015 | 0.000 |
| Q14 | 674.07 | 469 | 0.007 | 0.000 |
| Q15 | 1,105.33 | 475 | 0.012 | 0.000 |
| Q16 | 2,424.64 | 448 | 0.021 | 0.000 |
| Q17 | 857.94 | 465 | 0.009 | 0.000 |
| Q18 | 1,006.98 | 524 | 0.010 | 0.000 |
| Q19 | 834.28 | 465 | 0.009 | 0.000 |
| Q20 | 1,417.14 | 536 | 0.013 | 0.000 |
| Q21 | 1,530.07 | 539 | 0.014 | 0.000 |
| Q22 | 946.23 | 465 | 0.010 | 0.000 |

*Note*. Signed χ2 Statistics. RMSEA = Root Mean Square Error of Approximation, CFI = Comparative Fit Index, TLI = Tucker-Lewis Index, SRMR = Standardized Root Mean Square Residual.

## 

## Step 5: Model Residuals

To ensure that the items are locally independent, *mirt* offers several alternatives to examine the behavior of residuals. First, we demonstrate how to run Local Dependency (LD) $\chi$2 statistics (Chen & Thissen, 1997), which looks at the covariance between item pairs after accounting for . In this tutorial, we demonstrate the use of Yen’s *Q*3 statistics (Yen, 1984), which is considered more powerful for detecting underlying local dependency than LD $\chi$2 statistics (Chen & Thissen, 1997). To run Yen’s *Q*3, readers can type and run this command:

q3 <- residuals(fit, type = "Q3") # Running Yen's Q3 statistics.

This code asks R to run residuals() function using "Q3" as the type, and then, keep the value as a new matrix, namely q3. Again, we are only interested in item pairwise correlations above |0.2|, and to help us flag problematic pairs, we can make use of findCorrelation() function from *caret* package (Kuhn, 2024) as follows:

findCorrelation(q3, cutoff = 0.2, verbose = T) # Flagging problematic item pairwise correlations.

After running the code, readers may see in their R console that residual correlations between item Q3, Q4, Q5, Q7, Q11, Q13, Q14, Q18, Q19, and Q21 are above |0.2|. Interestingly, all of those items, except for Q4, have significant *S*-$\chi$2 statistics (see {apatb-table5}) as well, which tells us that we have to look at these items closely and then decide whether these items have shared characteristics beyond what is explained by the model.

## Step 6: IRT Plots

*mirt* offers features to visualize model parameters but the options for customizing these plots are rather limited. *ggmirt* package fills this gap by combining *mirt* and *ggplot2* functions. In this tutorial, we demonstrate the use of *ggmirt* to visualize Item Probability Function (IPF), Item Information Function (IIF), Test Information Function (TIF), and conditional probability plot.

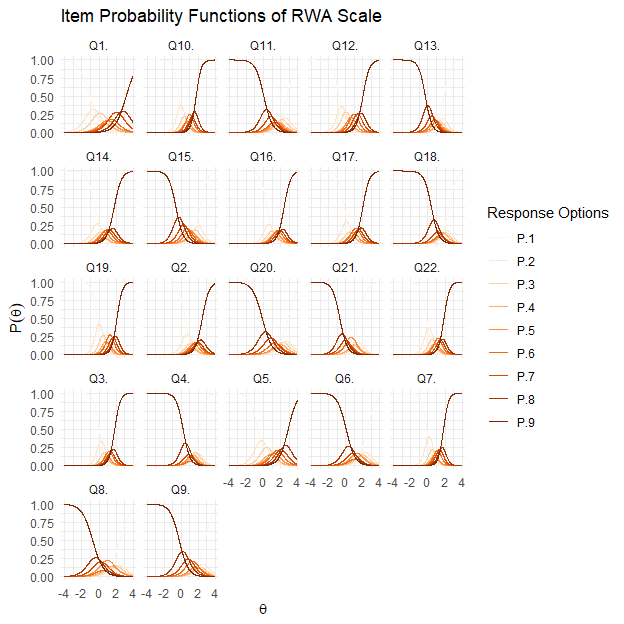
To visualize IPF, readers can type and run this function:

tracePlot(fit, title = "Item Probability Functions of RWA Scale") + labs(color="Response Categories")

The labs(color) function is to assign different colors to each curve representing a response category. The output of this function is Figure 3. In Figure 3, we see that as increases, respondents are more likely to choose higher response categories. The gradual transition from one response option to the next across the spectrum of values indicates that the items are capturing incremental increases in the RWA level. However, all IPFs of RWA items seem to be significantly overlapping and tend to peak on a value close to or higher than the mean. This implies that RWA items are more sensitive to differentiate participants with high levels of RWA.

Figure 3

Item Probability Functions of RWA Scale



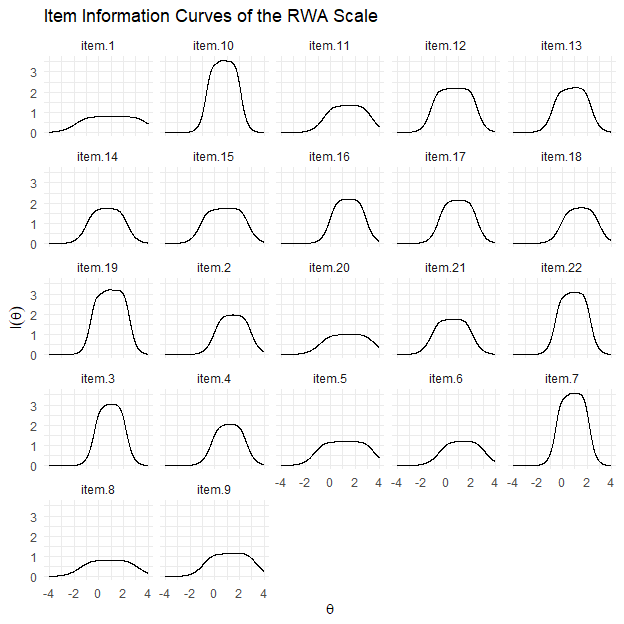
To evaluate the performance of each item in measuring the RWA trait, we can visualize the amount of information explained by each item with this simple line of code:

itemInfoPlot(fit, facet = TRUE, title = "Item Information Functions of the RWA Scale")

The facet option is set to TRUE so that R displays IIF for each item. The output for this function is Figure 4 where the x-axis represents the range of , and the y-axis indicates the amount of information provided by each item. Peaks that are higher and narrower indicate items that are very informative for specific levels of . In our case, most items provide the highest amount of information near the center to the right side of the distribution, indicating they are most useful for individuals with average to high levels of RWA.

Figure 4

Item Information Curves of the RWA Scale



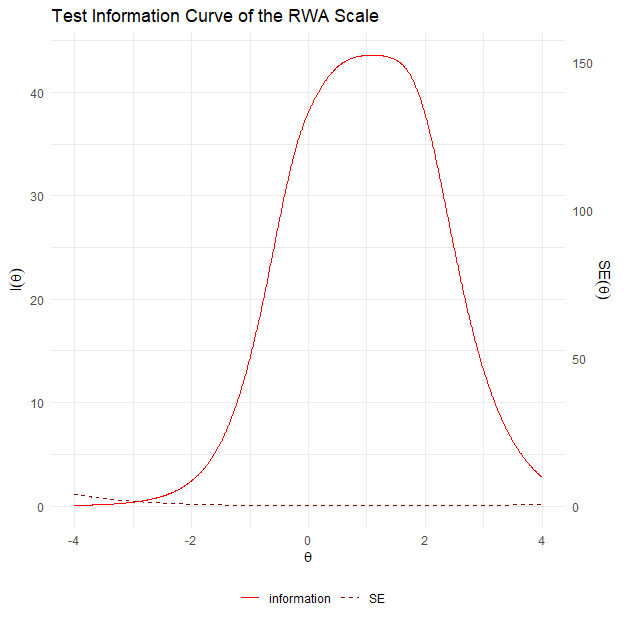
It is also possible to evaluate the overall performance of the RWA scale by plotting a TIF, which can be done by running the following code:

testInfoPlot(fit, title = "Test Information Function of the RWA Scale")

The output of this code is Figure 5, and according to this, the RWA scale as a whole is informative for measuring a group of individuals with a wide range of RWA levels, i.e., between -2*SD* to +4*SD*. However, the RWA scale is not optimal for measuring individuals with RWA levels below or beyond this range.

Figure 5

Test Information Curve of the RWA Scale



## Step 7: Computing Reliability

The heart of an IRT analysis is to estimate the precision of the measure, which can vary given different levels of . *mirt* provides two strategies to compute reliability that differ in their assumption regarding the distribution of . First, readers may calculate marginal reliability by running this simple command:

marginal\_rxx(fit)

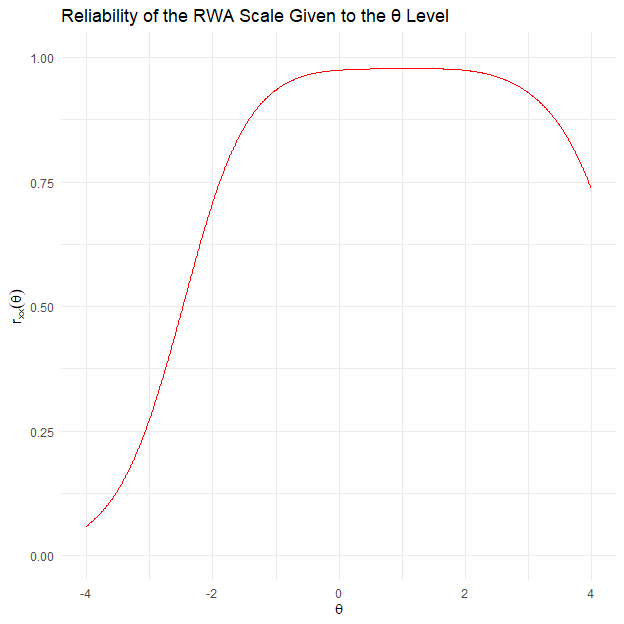
The output of the above code is 0.948, which indicates that the overall RWA scale is reliable, assuming that the underlying distribution follows the Gaussian or normal distribution. Since reliability can vary across different levels of , readers can also visualize the scale’s reliability given the levels of by running the function of *ggmirt* package, as follows:

conRelPlot(fit, title = "Reliability of the RWA Scale Given to the θ Level")

The output of this code is Figure 6, which shows that the RWA scale can measure individuals with levels between -2*SD* and +4*SD* with sufficient reliability, i.e., *r*xx 0.75.

Figure 6

Reliability of the RWA Scale Given to the θ Level



*mirt* can also calculate the RWA scale overall reliability using predicted by the model. To compute empirical reliability, one needs to calculate the estimated value of each participant first, and then, calculate reliability based on these values. To do this, readers can run these two lines of codes:

theta\_se <- fscores(fit, full.scores.SE = TRUE) # Extracting the estimated theta score of each participant.  
empirical\_rxx(theta\_se) # Then use the estimated theta to calculate empirical reliability.

The fscores() function is used to compute the estimated value for each participant predicted by the fit model. The full.scores.SE is set to TRUE because the package uses the value and standard error of each participant (*SE*) to compute the empirical reliability. We store the estimated and its *SE* in a new data frame, namely theta\_se, and then run empirical\_rxx() function to calculate empirical reliability. The output of these codes is 0.953, which is very close to marginal reliability we estimated before.

# Discussion

Construct validation is an ongoing process (Messick, 1995), so reporting evidence of validity should always be an important part of the routine. In practice, however, researchers are less interested in examining the performance of the measures they use in their study, making it difficult to assess the credibility of their research findings (Flake et al., 2017; Hussey & Hughes, 2020). In this tutorial, we demonstrate the applicability of GRM analysis as part of the IRT family to help applied psychology researchers assess their measurement quality. We provide a non-technical guide to implementing a GRM analysis through a simple 7-step process using a real, openly available dataset. To maximize the effectiveness of this tutorial, we show how to perform a GRM analysis using R, an open-source statistical software, and make all materials publicly available.

We begin the tutorial with a theoretical overview of IRT, which underlies GRM, so that readers can relate the practical steps to the theory behind the analysis. Since IRT is less popular than CTT, we also briefly explain how IRT differs from CTT in its assumptions. However, we want to emphasize that our goal here is not to argue for the superiority of CTT over IRT, since the choice of analytic tool depends largely on the specific research question at hand. It is important to highlight that researchers should be aware of the merits and limitations of their chosen methods. Therefore, researchers should always justify why they choose a particular method over many available alternatives.

Note that the model we estimated has a poor fit, perhaps due to local dependency, as we showed in steps 4 and 5. In practice, we recommend addressing model fitting issues before interpreting measurement precision or drawing substantive conclusions based on these estimates. One solution is to rephrase or to combine the content of the problematic items that indicate local dependency. Alternatively, readers may need to reconsider the structure. If the data is unidimensional but some of the items are locally dependent, this may indicate the existence of multiple within the data structure. In this sense, modeling the test data as a correlated multidimensional model or a bifactor model (i.e., multidimensional models with a *g* factor) may be a solution, given that a previous study shows that the RWA scale is multidimensional (Duckitt & Bizumic, 2013). Removing locally dependent items should be a last resort, as it may improve model fit but may also compromise measurement precision.

Construct validation is a critical step in the iterative process of theorizing a psychological phenomenon (Grahek et al., 2021). Therefore, we encourage applied psychology researchers to report evidence of measurement quality to ensure the validity of their findings and contribute to cumulative psychological science. One way to do this is to conduct and report a GRM analysis of their measures. We hope that this tutorial can be seen as a small step towards the larger goal of building robust psychological science.

# References

Altemeyer, B. (2006). *The Authoritarians*.

Baker, F. B., & Seock-Ho, K. (2017). *The basics of item response theory using R*. Springer Science+Business Media.

Chalmers, R. P. (2012). Mirt: A Multidimensional Item Response Theory Package for the R Environment. *Journal of Statistical Software*, *48*, 1–29. <https://doi.org/10.18637/jss.v048.i06>

Chalmers, R. P., Pritikin, J., Robitzsch, A., Zoltak, M., Kim, K., Falk, C. F., Meade, A., Schneider, L., King, D., Liu, C.-W., & Oguzhan, O. (2023). *Mirt: Multidimensional Item Response Theory*.

Chen, W.-H., & Thissen, D. (1997). Local Dependence Indexes for Item Pairs Using Item Response Theory. *Journal of Educational and Behavioral Statistics*, *22*(3), 265–289. <https://doi.org/10.3102/10769986022003265>

de Ayala, R. J. (2022). *The Theory and Practice of Item Response Theory*. The Guilford Press.

Duckitt, J., & Bizumic, B. (2013). Multidimensionality of Right-Wing Authoritarian Attitudes: Authoritarianism-Conservatism-Traditionalism: Authoritarianism-Conservatism-Traditionalism. *Political Psychology*, *34*(6), 841–862. <https://doi.org/10.1111/pops.12022>

Duckitt, J., & Sibley, C. G. (2017). The dual process motivational model of ideology and prejudice. In *The Cambridge handbook of the psychology of prejudice* (pp. 188–221). Cambridge University Press. <https://doi.org/10.1017/9781316161579.009>

Elson, M., Hussey, I., Alsalti, T., & Arslan, R. C. (2023). Psychological measures aren’t toothbrushes. *Commun Psychol*, *1*(1), 1–4. <https://doi.org/10.1038/s44271-023-00026-9>

Embretson, S. E., & Reise, S. P. (2000). *Item response theory for psychologists*. L. Erlbaum Associates.

Flake, J. K., & Fried, E. I. (2020). Measurement Schmeasurement: Questionable Measurement Practices and How to Avoid Them. *Advances in Methods and Practices in Psychological Science*, *3*(4), 456–465. <https://doi.org/10.1177/2515245920952393>

Flake, J. K., Pek, J., & Hehman, E. (2017). Construct Validation in Social and Personality Research: Current Practice and Recommendations. *Social Psychological and Personality Science*. <https://doi.org/10.1177/1948550617693063>

Grahek, I., Schaller, M., & Tackett, J. L. (2021). Anatomy of a Psychological Theory: Integrating Construct-Validation and Computational-Modeling Methods to Advance Theorizing. *Perspect Psychol Sci*, 1745691620966794. <https://doi.org/ghtczb>

Hambleton, R. K., Van Der Linden, W. J., & Wells, C. S. (2010). IRT models for the Analysis of Polytomously Scored Data: Brief and Selected History of Model Building Advances. In M. L. Nering & R. Ostini (Eds.), *Handbook of polytomous item response theory models* (pp. 21–42). Routledge.

Hussey, I., & Hughes, S. (2020). Hidden Invalidity Among 15 Commonly Used Measures in Social and Personality Psychology. *Advances in Methods and Practices in Psychological Science*, *3*(2), 166–184. <https://doi.org/10.1177/2515245919882903>

Kuhn, M. (2024). *Caret: Classification and Regression Training*.

Masur, P. K. (2023). *Ggmirt*.

Maydeu-Olivares, A., & Joe, H. (2006). Limited Information Goodness-of-fit Testing in Multidimensional Contingency Tables. *Psychometrika*, *71*(4), 713–732. <https://doi.org/10.1007/s11336-005-1295-9>

Messick, S. (1995). Validity of Psychological Assessment. *American Psychologist*, 9.

Navarro, D. J. (2018). R for Psychological Science. In *UNSW Computational Cognitive Science*. https://psyr.djnavarro.net/.

Osborne, D., Costello, T. H., Duckitt, J., & Sibley, C. G. (2023). The psychological causes and societal consequences of authoritarianism. *Nat Rev Psychol*, *2*(4), 220–232. <https://doi.org/10.1038/s44159-023-00161-4>

R Core Team. (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing.

Reise, S. P., Cook, K. F., & Moore, T. M. (2015). Evaluating the Impact of Multidimensionality on Unidimensional Item Response Theory Model Parameters. In S. P. Reise & D. A. Revicki (Eds.), *Handbook of Item Response Theory Modeling: Applications to Typical Performance Assessment* (pp. 13–40). Routledge.

Revelle, W. (2023). *Psych: Procedures for Psychological, Psychometric, and Personality Research*.

Samejima, F. (1997). Graded Response Model. In W. J. van der Linden & R. K. Hambleton (Eds.), *Handbook of Modern Item Response Theory* (pp. 85–100). Springer. <https://doi.org/10.1007/978-1-4757-2691-6_5>

Samejima, F. (2010). The General Graded Response Model. In *Handbook of Polytomous Item Response Theory Models*. Routledge.

Samejima, F. (2016). Graded Response Models. In *Handbook of Item Response Theory*. Chapman and Hall/CRC.

Saunders, B. A., & Ngo, J. (2017). The Right-Wing Authoritarianism Scale. In V. Zeigler-Hill & T. K. Shackelford (Eds.), *Encyclopedia of Personality and Individual Differences* (pp. 1–4). Springer International Publishing. <https://doi.org/10.1007/978-3-319-28099-8_1262-1>

Schermelleh-Engel, K., Moosbrugger, H., & Müller, H. (2003). Evaluating the Fit of Structural Equation Models: Tests of Significance and Descriptive Goodness-of-Fit Measures. *Methods of Psychological Research Online*, *8*(2), 23–74.

Yen, W. M. (1984). Effects of Local Item Dependence on the Fit and Equating Performance of the Three-Parameter Logistic Model. *Applied Psychological Measurement*, *8*(2), 125–145. <https://doi.org/10.1177/014662168400800201>