Item Response Theory - Final Essay

Marius Keute

September 18, 2022

Contents

submitted to:

Dr. Stefano Noventa

University of Tübingen

submitted by:

Marius Keute (QDS, 5991873)

Statutory Declaration: I hereby declare that I composed the present paper independently and that I have used no other resources than those indicated. The text passages which are taken from other works in wording or meaning have been identified as such. I also declare that this work has not been partly or completely used in another examination.

1 Introduction

Understanding sexual habits and behavior can be important for, e.g., improving sex education for adolescents, preventing sexually transmitted diseases (STDs), and identifying high-risk populations for sexual misconduct. The Sexual Compulsivity Scale (SCS) is a 10-item questionnaire constructed to measure hypersexuality and high libido in a given person (Kalichman and Rompa (1995),Kalichman and Rompa (2001)). Each of the 10 items is a statement about sexual habits, feelings, or experiences, and the test-taker can indicate how much they can relate to each statement on a four-level scale ranging from 1 (Not at all like me) to 4 (Very much like me).

The 10 items are (Kalichman and Rompa (2001)):

1. My sexual appetite has gotten in the way of my relationships.
2. My sexual thoughts and behaviors are causing problems in my life.
3. My desires to have sex have disrupted my daily life.
4. I sometimes fail to meet my commitments and responsibilities because of my sexual behaviors.
5. I sometimes get so horny I could lose control.
6. I find myself thinking about sex while at work.
7. I feel that sexual thoughts and feelings are stronger than I am.
8. I have to struggle to control my sexual thoughts and behavior.
9. I think about sex more than I would like to.
10. It has been difficult for me to find sex partners who desire having sex as much as I want to.

In this essay, using data from the original validation cohort (Kalichman and Rompa (2001)), I will provide a thorough analysis of the SCS, using methods derived from Item Response Theory (IRT), and to a lesser extent from Classical Test Theory (CTT). In the final section, I will give an overview over both theories and their key differences.

2 Preparing the Data

The dataset (Kalichman and Rompa (1995)) consists of 3376 observations, the variables being the ten items of the SCS, the sum score, gender and age. From the age variable, three cases where the reported age was 100 years or higher appeared implausible and therefore set to missing values. The remaining cases had a mean age of 30.9 years (median 28 years, range [14, 85]). From the gender variable, 13 values were missing and 15 cases where the reported gender was "3" (other) were set to missing values. Of the remaining cases, 2295 (68.5%) reported male gender ("1") and 1053 (31.4%) reported female gender ("2"). In the dataset, 133 cases contained at least one missing value.

The pattern of missing SCS items is shown in Figure 1. It can be seen that item Q9 was missing most often, though not by a large margin (Q9: 27 missing values, Q5: 13 missing values). It can be seen that the majority of cases with missing values (118 cases / 88.7%) had only a single missing item, while there were no prominent patterns of items that tended to be jointly missing. Eight cases where more than two SCS items were missing were excluded from all further analyses. For the remaining 3368 cases, the probability of missing values at each SCS variable was modeled as a function of the values in *all other* SCS variables using a logistic regression model:

$$P(M_{i,q} = 1|X_{i,\not{q}}) = \sigma(X_{i,\not{q}}\hat{\beta}),$$

where $M_{i,q}$ is 1 if the i^{th} person has a missing value at item $q \in \{Q1, Q2, ...Q10\}$, $X_{i,\not{q}}$ denotes the item values of all other items except item q, σ is the logistic function $\sigma(x) = \frac{1}{1-e^{-x}}$, and $\hat{\beta}$ are the estimated regression weights (Guan and Yusoff (2011)). Note that each variable's pattern of missing values could only be predicted based on the observations without missing values in any other variable, since those cases were excluded by the logistic model by default of the implementation. Since the majority of cases had either no or only one variable missing, however, this should not bias the overall picture very much.

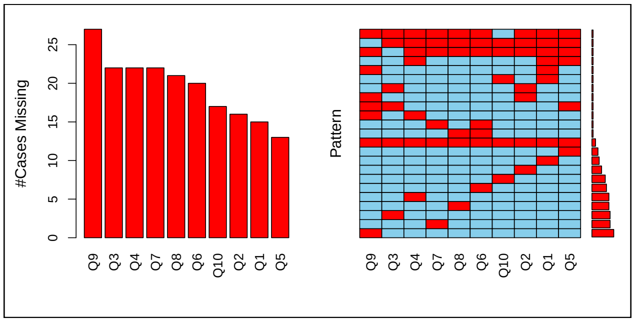


Figure 1: Pattern of missing SCS values.

3 Descriptive Analyses and Dichotomization

The distribution of responses for each item before dichotomization can be seen in Figure 2. All item categories show reasonable coverage of the range of responses (1-4), and there are no obvious flooring or ceiling effects, except for a slight tendency to a flooring effect with item Q6 (few cases with response 1).

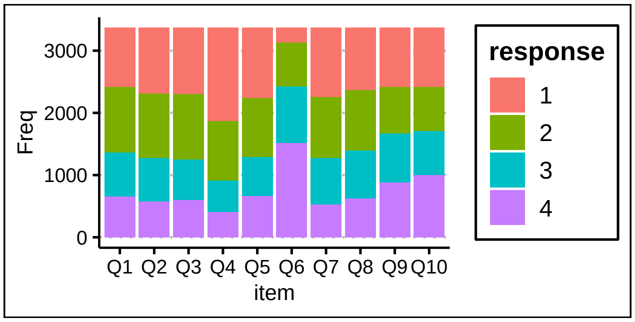


Figure 2: Distribution of non-dichotomized responses per item

For dichotomization of the item data, I considered two options, namely, thresholding each of the 10 items at its own median, to ensure an even distribution of observations into both categories for each item, or finding a common threshold for all items. Since the items have only four levels each, a median split would not necessarily lead to a very balanced dichotomization. Furthermore, the item levels are designed to have the same meaning across all items, therefore I decided to dichotomize at a common threshold of 2, i.e., the dichotomous items $D_q \in \{D_1, D_2, ..., D_{10}\}$ were defined such that

$$D_{i,q} = \begin{cases} 0 \text{ if } Q_{i,q} \in \{1,2\}, \\ 1 \text{ if } Q_{i,q} \in \{3,4\}, \end{cases}$$

Of note, simple models in IRT such as the Rasch model (see below) assume that all item responses are either correct or incorrect (or solved / unsolved, respectively). Since a personality test such as the SCS does not have right or wrong responses, it is common to dichotomize the values, as described above, and henceforth treat one of the dichotomous response options as the 'correct' one, in this case, responses greater than 2. This is, however, purely for compliance with IRT terminology and does not imply that the 'correct' dichotomous responses are better than the 'incorrect' ones in any way.

Descriptive characteristics of the 10 SCS items are shown in Table 1, the proportions of correct responses are shown in Figure 2. Since most variables' median was 2, this was not much different from an item-wise median threshold (see Table 1).

Subsequently, I calculated biserial correlations between all pairs of dichotomized items. Moreover, I calculated item discrimination, i.e., each items ability to discriminate between high- and low-scoring individuals, using the adjusted item-total correlation method (Reynolds and Livingston (2021)), i.e., by calculating biserial correlation coefficients between each (dichotomized) item's scores and the sum of all other (dichotomized) items.

Table 1: Descriptive item statistics (mean, median and range *before* dichotomization)

stat	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
max	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
mean	2.3	2.2	2.2	1.9	2.2	3.1	2.2	2.3	2.5	2.5
median	2.0	2.0	2.0	2.0	2.0	3.0	2.0	2.0	2.0	3.0
min	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 2: Distribution and discrimination of dichotomized items

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
percent in category 1	40.5	37.9	37.3	27	38.2	71.9	37.7	41.3	49.4	50.9
number of cases in category 1	1365	1275	1257	911	1285	2422	1269	1390	1663	1713
discrimination	0.45	0.45	0.44	0.34	0.29	0.26	0.42	0.37	0.31	0.36

Item intercorrelations are shown in Figure 3. It can be seen that all pairs of items show moderate to high positive correlations, indicating that all items measure similar information yet are not redundant. Item easiness (i.e., proportion of correct responses) was between 27% (item Q4) and 71.9% (item Q6), item discrimination was between .26 (item Q6) and .45 (items Q1, Q2), i.e., there was no item with a trivial response pattern (e.g., all or no responses correct), and no item was a good representation of the entire scale, since all item discriminations were only moderate in size.

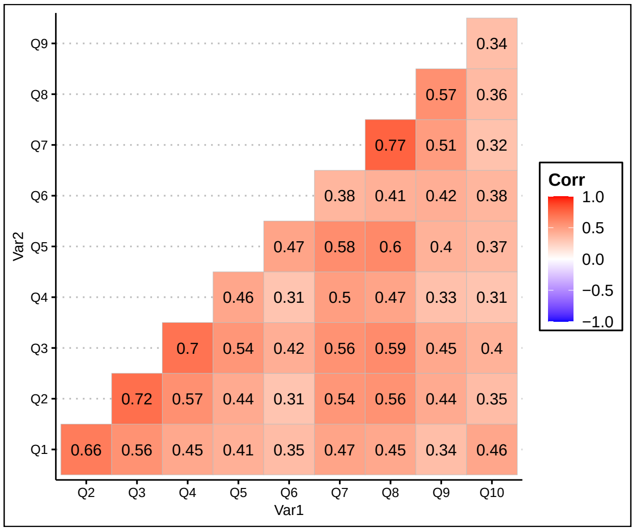


Figure 3: Pattern of missing SCS values.

4 IRT modeling

After analyzing the SCS data using descriptive statistics and concepts derived from CTT, in the following I will fit and discuss different IRT models to the data.

4.1 Rasch model estimation

Next, I estimated a *Rasch* model for the SCS data, also known as either the one-parameter logistic model or one-parameter normal ogive model, depending on the parameterization.

It models a given person's chances of solving a given item as a logistic function of the difference between the q^{th} item's difficulty β_q and the i^{th} person's ability θ_i, where β_q and θ_i are latent (unobserved) quantities that are estimated from the dichotomous (solved vs. not solved) item data.

The probability for a given person can then be expressed by the logistic function: $P(D_{i,q} = 1|\beta_q, \theta_i) = \sigma(\theta_i - \beta_q)$, where σ is the logistic function as specified above. That is to say, it is purely the difference between item difficulty and person ability that explains the correctness of item responses within the model.

Crucially, the model assumes that this relationship is identical for all items, i.e., the logistic function can only be shifted but not changed in slope across items with different difficulty. Item difficulty is, therefore, the only free parameter of the *Rasch* model, whereas alternative models (see below) also estimate additional parameters.

To obtain a comprehensive picture, I fitted *Rasch* models using three different software implementations in `R` 4.1.

The first method was the one implemented in the R package `eRm` (Mair and Hatzinger (2007)). The `eRm::RM` function estimates a *Rasch* model using conditional maximum likelihood estimation. To make the model identifiable, the user can choose between two model constraints, namely that the model parameters must sum to 0 or that the first item's parameter is fixed to 0. I chose the first (default) option, i.e., forcing item difficulties to sum to 0. Item discriminativity, i.e., the steepest slope of the logistic functions (at $\beta_q = \theta_i$), is fixed to 1 for all items in this implementation.

The second method was the one implemented in the R package `ltm` (Rizopoulos (2006)). The `ltm::rasch` function estimates a *Rasch* model using approximate marginal maximum likelihood estimation. This package provides the user with more flexibility to impose constraints on the model than `eRm`, I fixed item discriminativity to 1 for all items, to maximize comparability with the `eRm` parameters.

The third method was a structural equation model as implemented in `lavaan` (Rosseel (2012)). Unlike the two previous implementations, `lavaan` requires a more explicitly user-defined model specification, as it does not provide any ready-made function or syntax for *Rasch* models. I used a modified copy of the syntax presented in Templin (2022):

```
SCS =~ 1*Q1 + 1*Q2 + 1*Q3 + 1*Q4 + 1*Q5 + 1*Q6 + 1*Q7 + 1*Q8 + 1*Q9 + 1*Q10
```

```
Q1 | t1; Q2 | t1; Q3 | t1; Q4 | t1; Q5 | t1; Q6 | t1; Q7 | t1;
Q8 | t1; Q9 | t1;Q10 | t1;

SCS ~ 0;

SCS ~~ 1*SCS
```

Again, I fixed item discriminativities to 1 for all items. The item parameters `Q1`, . . . , `Q10` were subjected to a common threshold `t1`, and the sum of all item parameters (corresponding to the latent variable `SCS`) was fixed to 0, as with `eRm`. Moreover, its variance was fixed to unit.

4.2 Model analysis

The item difficulty parameters of the three models are shown in Figure 4, along with the item difficulty derived from CTT (i.e., the proportion of incorrect responses per item). While the parameters differed between the different models, it is important to note that the parameters from all four models (including CTT) were perfectly correlated for all pairs of models (all r > .999), which indicates that the parameters of one model are simply affine linear transformations of the parameters of any other model, i.e., while numerically different, the models incorporated identical information about the items. The corresponding item-characteristic curves (ICC) are shown in Figure 5. ICCs are generated by calculating the function graph of the item-wise logistic functions parameterized by item difficulty, across a range of possible person ability values on the x-axis.

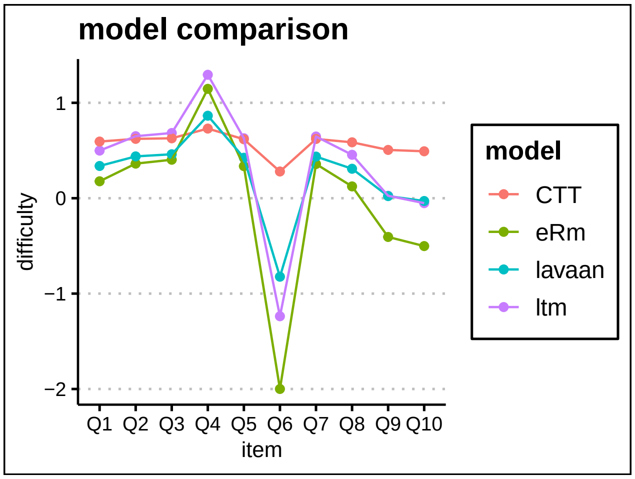


Figure 4: Item difficulties in comparison.

#TODO insert table with fits, discuss the implications

4.3 DIF

I tested for differential item functioning (DIF) using the package `difR` and the procedure outlined in the companion paper (Magis et al. (2010)). DIF is a disadvantageous property of a *Rasch* model, meaning that item responses differ between subjects from different participant

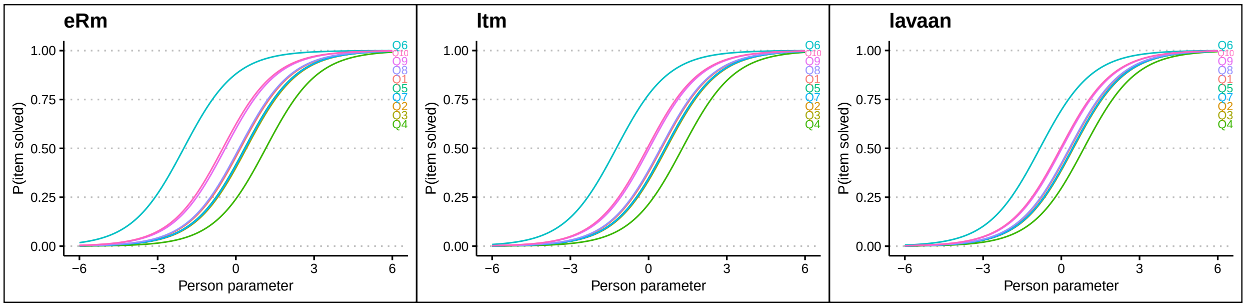


Figure 5: Item-characteristic curves for the three Rasch models.

groups, even given the same estimated ability level. The presence of DIF indicates a lack of measurement invariance of the model. The results are displayed in 6. I used the `difLord` method to investigate DIF, but obtained essentially the same results with the `difRaju` method. I discarded `difLRT`, the third recommended IRT-related method, due to its high computational demand. I tested for DIF across genders (male vs. female) and age groups (above median age vs. smaller or equal to median age). Significant DIF (FDR-corrected p-value $< .05$) across the gender groups was detected for items Q5 and Q10, and across the age groups for item Q1, Q5, and Q10. However, the effect sizes were in the negligible range for all but item Q5 in the gender group comparison, where a moderate effect was detected ($\Delta_{Lord} = -1.14$). Since I have only been considering Rasch (one-parameter) models so far, this analysis only tested for uniform DIF.

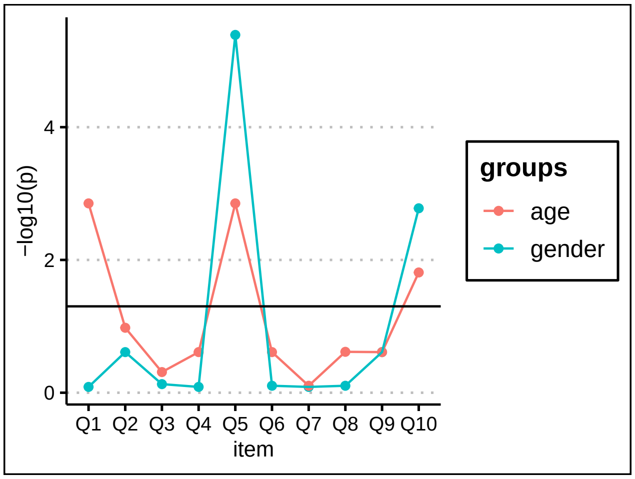


Figure 6: p-values (FDR corrected, negative log-transformed) from DIF analysis. Black line: significance threshold (p $< .05$)

5 Alternative IRT models

There are several extensions and alternatives to the *Rasch* model with its restrictive assumptions that differences between items can be described by just one parameter, namely item difficulty, while the *Birnbaum* model or 2-parameters logistic model also takes into account item discriminativity (corresponding to varying slopes of the item-characteristic curves of different items), and other possible models additionally include a guessing probability term (corresponding to various vertical offsets of the item-characteristic curves of different items) or ceiling probability term (corresponding to clipping the item-characteristic curves from above). For the given dataset, it appears reasonable to estimate a *Birnbaum* model, whereas 3-PL or 4-PL models seem difficult, since guessing and ceiling probabilities are not easy to operationalize for the dataset, considering that there is no ground truth to the items and we found no prominent ceiling or flooring in any item.

6 ICCs

7 fit

8 model comparison

9 Factor models

original categorical data bifactor, ...

9.1 Polytomous model

Procedure outlined in Smyth (2022)

10 Data Dimensionality

11 Reliability and Measurement Invariance

12 Theoretical Part: Key differences between IRT and CTT

12.1 Introduction

Unlike some physical quantities, many of the variables of interest in psychology, economics, and other human-centric fields, are latent, i.e., not directly observable. Researchers often try to reconstruct such latent variables by combining several observable variables. In particular, for psychological concepts such as personality traits, a person's score will often be estimated as a combination of item responses in a psychological test. Even though forerunners of psychological tests have been around for centuries, a comprehensive theory of psychological testing only emerged roughly half a century ago. Commonly, Melvin Novick is considered the first author to present a comprehensive account of Classical Test Theory (CTT) (Novick (1965)). Around the same time, a probabilistic view of psychological testing began to emerge, which are now referred to as Item Response Theory (IRT) (Rasch (1960)). It is interesting to note that *Classical* Test Theory does, therefore, not refer to the theory itself being older, but that it rather describes the ‘classical’ way authors thought about psychological testing from the early 20^{th} century onward, whereas probabilistic approaches became popular only later, when increasing computational capacity made them practical. In the following, I will describe some of the core ideas underlying CTT and ITT, their respective strengths and limitations, and practical applications.

12.2 Core Ideas

12.2.1 CTT

The traditional view of psychological tests (Classical Test Theory, CTT) conceived a given person’s total score across all items of a test as an additive combination of the person’s true score and a testing error: $X_i = \tau_i + \epsilon_i$ (Van der Linden and Hambleton (1997)). (Crocker and Algina (2008))

12.2.2 IRT

12.3 Strenghts

12.4 Limitations

12.5 Conclusion and Application

13 Analysis code

In the following, the complete analysis code and its output are shown.

```
require(ggplot2)
require(ggthemes)
require(reshape2)
require(readxl)
require(VIM)
require(mice)
require(dplyr)
require(tidyr)
require(psych)
require(ggcorrplot)
require(eRm)
require(ltm)
require(lavaan)
require(patchwork)
require(difR)

#####
#part 1: data preparation, descriptive analyses
#####
{
df = read_xlsx("SCS_data.xlsx")
SCS_vars = names(df)[1:10]
#set missing values
print(table(df$gender))
df$gender[df$gender == 3] = NA
df[df==0] = NA

print(unique(df$age))
df$age[df$age >= 100] = NA
mean(df$age,na.rm=T)
median(df$age,na.rm=T)
min(df$age,na.rm=T)
max(df$age,na.rm=T)

sprintf("%i cases are incomplete",sum(!complete.cases(df)))
sprintf("%i cases have incomplete SCS data",sum(!complete.cases(df[,SCS_vars])))


#missing data motifs
# and missing proportion per item
pdf("missingplot.pdf",width = 8, height = 4)
```

```
aggr(df[!complete.cases(df[,SCS_vars]),SCS_vars],
     numbers=TRUE, sortVars=TRUE,prop=FALSE,
     labels=SCS_vars,
     ylab=c("#Cases Missing","Pattern"))
box(which = "figure",lwd=2)
dev.off()

nmissing = rowSums(is.na(df[,SCS_vars]))
table(nmissing[nmissing!=0])
prop.table(table(nmissing[nmissing!=0]))

#missing-at-random analysis
#(check whether missing data points in each variable
#can be jointly predicted by all the other variables)
pvals = data.frame(matrix(ncol = length(SCS_vars), nrow=0))
colnames(pvals) = SCS_vars
for (var in SCS_vars){
  formula = sprintf("I(is.na(%s)) ~ .", var)
  formula0 = sprintf("I(is.na(%s)) ~ 1", var)
  m = summary(glm(formula, data=df[,1:10]))$coefficients
  pvals[var,rownames(m)[2:10]] = m[2:10,"Pr(>|t|)"]
}
min(p.adjust(unlist(pvals), method="fdr"),na.rm=T)


#-> missing at random can be assumed
#remove cases where more than two SCS variables are missing

#15 cases removed
df_clean = df[rowSums(is.na(df[,SCS_vars])) <= 2,]

#use multiple imputation for remaining data
df_clean = complete(mice(df_clean))


#descriptives
df_clean[,1:10] %>% summarise_all(list(mean=mean, median = median,
                                       min = min, max = max)) %>%
  round(1) %>%
  gather(variable, value) %>%
  separate(variable, c("var", "stat"), sep = "\\_") %>%
  spread(var, value) -> descriptives

#fix order of columns in descriptives table
```

```
descriptives = descriptives[,c("stat",SCS_vars)]

#re-calculate sum score
df_clean$score = rowSums(df_clean[,1:10])

#distribution plot before dichotomization
tmp = melt(
          cbind(data.frame(id=1:nrow(df_clean)),df_clean[,SCS_vars]),
          id.vars="id")
tmp2 = data.frame(table(tmp$variable,tmp$value))
colnames(tmp2) = c("item","response","Freq")
ggplot(tmp2,aes(x=item, y=Freq, fill=response))+geom_col()+theme_clean()
ggsave("distroplot.pdf",width = 4,height = 2)

#dichotomization
dich = df_clean
dich[,1:10] = data.frame(lapply(df_clean[,1:10],
                                function (x) as.numeric(x > 2)))
dich$score = rowSums(dich[,1:10])

}
```

```
## 
##    0    1    2    3 
##   13 2295 1053   15 
##  [1]  41  50  23  42  36  29  24  35  26  43  21  39  37  64  28  46  34  31  47
## [20]  22  61  16  40  33  30  56  49  51  18  20  45  32  15  27  25  59  58  19
## [39]  14  38  48  44  55 100  65  17  77  57  60  52  53  62  71  78  54  63  67
## [58]  68  72 999  85  69  70  66  84 123  73
```

```
## Warning in plot.aggr(res, ...): not enough vertical space to display frequencies
## (too many combinations)
```

```
## 
##  Variables sorted by number of missings: 
##  Variable Count
##        Q9    27
##        Q3    22
##        Q4    22
##        Q7    22
##        Q8    21
##        Q6    20
##       Q10    17
##        Q2    16
##        Q1    15
##        Q5    13
```

```
## 
##  iter imp variable
##   1   1  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   1   2  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   1   3  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   1   4  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   1   5  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   2   1  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   2   2  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   2   3  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   2   4  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   2   5  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   3   1  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   3   2  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   3   3  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   3   4  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   3   5  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   4   1  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   4   2  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   4   3  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   4   4  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   4   5  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   5   1  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   5   2  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   5   3  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   5   4  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
##   5   5  Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  gender  age
```

```r
#####
#part 2: CTT-style item analysis
#####
{

  #biserial correlations
  biserial_cor = biserial(dich[,SCS_vars],dich[,SCS_vars])
  ggcorrplot(biserial_cor, type = "lower", lab = TRUE)+theme_clean()
  ggsave("biserial_cor_mat.pdf",width = 6, height = 6)
  #dichotomous item statistics (percent and N correct, discriminativity)
  dich.distro = rbind(as.character(round(100*unlist(lapply(dich[,SCS_vars],
                                                           mean)),1)),
                      as.character(as.integer(unlist(lapply(dich[,SCS_vars],
                                                            sum)))))
  rownames(dich.distro) = c("percent in category 1",
                            "number of cases in category 1")
  discrimination = c()
  for (item in 1:10){
    itemname = SCS_vars[item]
    discrimination[itemname] = as.character(round(biserial(
      rowSums(dich[,-item]),dich[,item]),2))
  }

  dich.stats = rbind(dich.distro, discrimination)
}
```

```
## Warning in biserialc(x[, j], y[, i], j, i): For x = 1 y = 1 x seems to be
## dichotomous, not continuous

## Warning in biserialc(x[, j], y[, i], j, i): For x = 2 y = 2 x seems to be
## dichotomous, not continuous

## Warning in biserialc(x[, j], y[, i], j, i): For x = 3 y = 3 x seems to be
## dichotomous, not continuous

## Warning in biserialc(x[, j], y[, i], j, i): For x = 4 y = 4 x seems to be
## dichotomous, not continuous

## Warning in biserialc(x[, j], y[, i], j, i): For x = 5 y = 5 x seems to be
## dichotomous, not continuous

## Warning in biserialc(x[, j], y[, i], j, i): For x = 6 y = 6 x seems to be
## dichotomous, not continuous

## Warning in biserialc(x[, j], y[, i], j, i): For x = 7 y = 7 x seems to be
## dichotomous, not continuous

## Warning in biserialc(x[, j], y[, i], j, i): For x = 8 y = 8 x seems to be
## dichotomous, not continuous

## Warning in biserialc(x[, j], y[, i], j, i): For x = 9 y = 9 x seems to be
## dichotomous, not continuous

## Warning in biserialc(x[, j], y[, i], j, i): For x = 10 y = 10 x seems to be
## dichotomous, not continuous
```

```
#####
#part 2: estimate Rasch model
#####
{
  #approach 1: eRm
  #prepare data for eRm estimation
  #(just item data in wide format)
  rasch_model_eRm = RM(dich[,SCS_vars])
  smr_eRm = summary(rasch_model_eRm)
```

```
#approach 2: ltm
#constraint fixes item discriminativity to 1
rasch_model_ltm = rasch(dich[,SCS_vars],
                        constraint = cbind(length(SCS_vars) + 1, 1))
smr_ltm = summary(rasch_model_ltm)



#TODO check syntax
#aproach 3: lavaan
#modified copy from https://jonathantemplin.com/wp-content/uploads/2022/02/
                    #EPSY906_Example05_Binary_IFA-IRT_Models.nb.html
lavaansyntax = "

  # loadings/discrimination parameters:
  SCS =~ 1*Q1 + 1*Q2 + 1*Q3 + 1*Q4 + 1*Q5 + 1*Q6 + 1*Q7 + 1*Q8 + 1*Q9 + 1*Q10

  # threshholds use the | operator and start at value 1 after t:
  Q1 | t1; Q2 | t1; Q3 | t1; Q4 | t1; Q5 | t1; Q6 | t1; Q7 | t1;
  Q8 | t1; Q9 | t1;Q10 | t1;

  # factor mean:
  SCS ~ 0;

    # factor variance:
  SCS ~~ 1*SCS

  "

rasch_model_lavaan = sem(model = lavaansyntax, data =  dich[,SCS_vars],
                         ordered = SCS_vars, mimic = "Mplus",
                         estimator = "WLSMV", std.lv = TRUE,
                         parameterization = "theta")
smr_lavaan = summary(rasch_model_lavaan, fit.measures = TRUE,
                     rsquare = TRUE, standardized = TRUE)

convertTheta2IRT = function(lavObject){
  #modified copy from
  #https://jonathantemplin.com/wp-content/uploads/2022/02/
      #EPSY906_Example05_Binary_IFA-IRT_Models.nb.html

  if (!lavObject@Options$parameterization == "theta") {
    stop("your model is not estimated with parameterization='theta'")
    }
  output = inspect(object = lavObject, what = "est")
  if (ncol(output$lambda)>1) { stop("IRT conversion is only valid
           for one dimensional factor models.
           Your model has more than one dimension.")
    }
  a = output$lambda
  b = output$tau/output$lambda
  return(list(a = a, b=b))
}

#make ICC plot function
ICC_plot = function(difficulty, discriminativity = 1){
  if (length(discriminativity)==1){
      discriminativity = rep(discriminativity, length(difficulty))
    }
  df = data.frame(x=seq(-6,6,.01))
  for (i in 1:length(difficulty)){
    df[[SCS_vars[i]]] = logistic(x=df$x, d=difficulty[i],
                                 a=discriminativity[i])
  }

  df = melt(df, id.vars = "x")
  colnames(df)[2] = "item"
  plt=ggplot(df, aes(x = x, y = value, color = item, label = item)) +
    geom_line() + theme_clean() + xlab("Person parameter") +
    ylab("P(item solved)")
  return(directlabels::direct.label(plt, "last.qp"))
}



#make ICC plots
difficulties_eRm = -rasch_model_eRm$betapar
iccplot_eRm=ICC_plot(difficulties_eRm)+ggtitle("eRm")



#lme4 difficulties are shifted by .42 from eRm difficulties, why?

difficulties_ltm = smr_ltm$coefficients[1:10,"value"]
iccplot_ltm = ICC_plot(difficulties_ltm)+ggtitle("ltm")


difficulties_lavaan =   convertTheta2IRT(lavObject = rasch_model_lavaan)$b
```

```
#TODO: check ICC plotting fct
iccplot_lavaan=ICC_plot(difficulties_lavaan)+ggtitle("lavaan")



difficulties = rbind( data.frame(model="eRm",
                          item=factor(paste0("Q",as.character(1:10))),
                          difficulty=as.numeric(difficulties_eRm)),
              data.frame(model="ltm",
                         item=factor(paste0("Q",as.character(1:10))),
                         difficulty=as.numeric(difficulties_ltm)),
              data.frame(model="lavaan",
                         item=factor(paste0("Q",as.character(1:10))),
                         difficulty=as.numeric(difficulties_lavaan)),
              data.frame(model="CTT",
                         item=factor(paste0("Q",as.character(1:10))),
                         difficulty=1-as.numeric(dich.distro[1,])/100))
difficulties_plot = ggplot(difficulties,aes(x=item,y=difficulty,
                                            color=model,group=model)) +
  geom_point() + geom_line() + theme_clean() + ggtitle("model comparison")+
  scale_x_discrete(breaks=paste0("Q",1:10),limits=paste0("Q",1:10))


difficulties_plot
ggsave("diffcfig.pdf",width = 4,height = 3)

#arrange plots vertically and save
iccplot_eRm|iccplot_ltm|iccplot_lavaan

ggsave("iccfig.pdf",width = 12,height = 3)


#compare fits
summary(rasch_model_eRm)
smr_lavaan$FIT
smr_ltm$AIC

}
```

```
##
## Results of RM estimation:
##
## Call:  RM(X = dich[, SCS_vars])
##
```

```
## Conditional log-likelihood: -9887.962
## Number of iterations: 8
## Number of parameters: 9
##
## Item (Category) Difficulty Parameters (eta): with 0.95 CI:
##     Estimate Std. Error lower CI upper CI
## Q2     0.363      0.043    0.278    0.447
## Q3     0.403      0.043    0.319    0.488
## Q4     1.146      0.046    1.056    1.237
## Q5     0.336      0.043    0.252    0.420
## Q6    -2.000      0.049   -2.097   -1.903
## Q7     0.358      0.043    0.274    0.443
## Q8     0.124      0.042    0.040    0.207
## Q9    -0.406      0.042   -0.489   -0.323
## Q10   -0.502      0.042   -0.585   -0.419
##
## Item Easiness Parameters (beta) with 0.95 CI:
##          Estimate Std. Error lower CI upper CI
## beta Q1    -0.177      0.043   -0.261   -0.094
## beta Q2    -0.363      0.043   -0.447   -0.278
## beta Q3    -0.403      0.043   -0.488   -0.319
## beta Q4    -1.146      0.046   -1.237   -1.056
## beta Q5    -0.336      0.043   -0.420   -0.252
## beta Q6     2.000      0.049    1.903    2.097
## beta Q7    -0.358      0.043   -0.443   -0.274
## beta Q8    -0.124      0.042   -0.207   -0.040
## beta Q9     0.406      0.042    0.323    0.489
## beta Q10    0.502      0.042    0.419    0.585
##
## lavaan 0.6-11 ended normally after 7 iterations
##
##   Estimator                                       DWLS
##   Optimization method                           NLMINB
##   Number of model parameters                        10
##
##   Number of observations                          3368
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                              2422.527    1522.217
##   Degrees of freedom                                45          45
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  1.609
##   Shift parameter                                           17.049
##        simple second-order correction (WLSMV)
```

```
##
## Model Test Baseline Model:
##
##   Test statistic                             39069.832   24291.970
##   Degrees of freedom                                45          45
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.609
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.939       0.939
##   Tucker-Lewis Index (TLI)                       0.939       0.939
##
##   Robust Comparative Fit Index (CFI)                            NA
##   Robust Tucker-Lewis Index (TLI)                               NA
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.125       0.099
##   90 Percent confidence interval - lower         0.121       0.095
##   90 Percent confidence interval - upper         0.130       0.103
##   P-value RMSEA <= 0.05                          0.000       0.000
##
##   Robust RMSEA                                                  NA
##   90 Percent confidence interval - lower                        NA
##   90 Percent confidence interval - upper                        NA
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.109       0.109
##
## Weighted Root Mean Square Residual:
##
##   WRMR                                           6.637       6.637
##
## Parameter Estimates:
##
##   Standard errors                           Robust.sem
##   Information                                 Expected
##   Information saturated (h1) model        Unstructured
##
## Latent Variables:
##                     Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   SCS =~
##     Q1                1.000                               1.000    0.707
##     Q2                 1.000                               1.000    0.707
##     Q3                 1.000                               1.000    0.707
##     Q4                 1.000                               1.000    0.707
##     Q5                 1.000                               1.000    0.707
##     Q6                 1.000                               1.000    0.707
##     Q7                 1.000                               1.000    0.707
##     Q8                 1.000                               1.000    0.707
##     Q9                 1.000                               1.000    0.707
##     Q10                1.000                               1.000    0.707
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     SCS                0.000                               0.000    0.000
##    .Q1                 0.000                               0.000    0.000
##    .Q2                 0.000                               0.000    0.000
##    .Q3                 0.000                               0.000    0.000
##    .Q4                 0.000                               0.000    0.000
##    .Q5                 0.000                               0.000    0.000
##    .Q6                 0.000                               0.000    0.000
##    .Q7                 0.000                               0.000    0.000
##    .Q8                 0.000                               0.000    0.000
##    .Q9                 0.000                               0.000    0.000
##    .Q10                0.000                               0.000    0.000
##
## Thresholds:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     Q1|t1              0.338    0.031   10.946    0.000    0.338    0.239
##     Q2|t1              0.438    0.031   14.103    0.000    0.438    0.310
##     Q3|t1              0.461    0.031   14.788    0.000    0.461    0.326
##     Q4|t1              0.865    0.033   26.424    0.000    0.865    0.611
##     Q5|t1              0.424    0.031   13.657    0.000    0.424    0.300
##     Q6|t1             -0.823    0.033  -25.320    0.000   -0.823   -0.582
##     Q7|t1              0.436    0.031   14.034    0.000    0.436    0.308
##     Q8|t1              0.309    0.031   10.018    0.000    0.309    0.218
##     Q9|t1              0.022    0.031    0.724    0.469    0.022    0.016
##     Q10|t1            -0.029    0.031   -0.965    0.335   -0.029   -0.021
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     SCS                1.000                               1.000    1.000
##    .Q1                 1.000                               1.000    0.500
##    .Q2                 1.000                               1.000    0.500
##    .Q3                 1.000                               1.000    0.500
##    .Q4                 1.000                               1.000    0.500
##    .Q5                 1.000                               1.000    0.500
##    .Q6                1.000                                1.000    0.500
##    .Q7                1.000                                1.000    0.500
##    .Q8                1.000                                1.000    0.500
##    .Q9                1.000                                1.000    0.500
##    .Q10               1.000                                1.000    0.500
##
## Scales y*:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     Q1                0.707                                0.707    1.000
##     Q2                0.707                                0.707    1.000
##     Q3                0.707                                0.707    1.000
##     Q4                0.707                                0.707    1.000
##     Q5                0.707                                0.707    1.000
##     Q6                0.707                                0.707    1.000
##     Q7                0.707                                0.707    1.000
##     Q8                0.707                                0.707    1.000
##     Q9                0.707                                0.707    1.000
##     Q10               0.707                                0.707    1.000
##
## R-Square:
##                    Estimate
##     Q1                0.500
##     Q2                0.500
##     Q3                0.500
##     Q4                0.500
##     Q5                0.500
##     Q6                0.500
##     Q7                0.500
##     Q8                0.500
##     Q9                0.500
##     Q10               0.500
```

```
##
## Results of RM estimation:
##
## Call:  RM(X = dich[, SCS_vars])
##
## Conditional log-likelihood: -9887.962
## Number of iterations: 8
## Number of parameters: 9
##
## Item (Category) Difficulty Parameters (eta): with 0.95 CI:
##     Estimate Std. Error lower CI upper CI
## Q2     0.363      0.043    0.278    0.447
## Q3     0.403      0.043    0.319    0.488
## Q4     1.146      0.046    1.056    1.237
## Q5     0.336      0.043    0.252    0.420
## Q6    -2.000      0.049   -2.097   -1.903
## Q7     0.358      0.043    0.274    0.443
## Q8     0.124      0.042    0.040    0.207
## Q9    -0.406      0.042   -0.489   -0.323
## Q10   -0.502      0.042   -0.585   -0.419
##
## Item Easiness Parameters (beta) with 0.95 CI:
##          Estimate Std. Error lower CI upper CI
## beta Q1    -0.177      0.043   -0.261   -0.094
## beta Q2    -0.363      0.043   -0.447   -0.278
## beta Q3    -0.403      0.043   -0.488   -0.319
## beta Q4    -1.146      0.046   -1.237   -1.056
## beta Q5    -0.336      0.043   -0.420   -0.252
## beta Q6     2.000      0.049    1.903    2.097
## beta Q7    -0.358      0.043   -0.443   -0.274
## beta Q8    -0.124      0.042   -0.207   -0.040
## beta Q9     0.406      0.042    0.323    0.489
## beta Q10    0.502      0.042    0.419    0.585
```

```
## [1] 37143.44
```

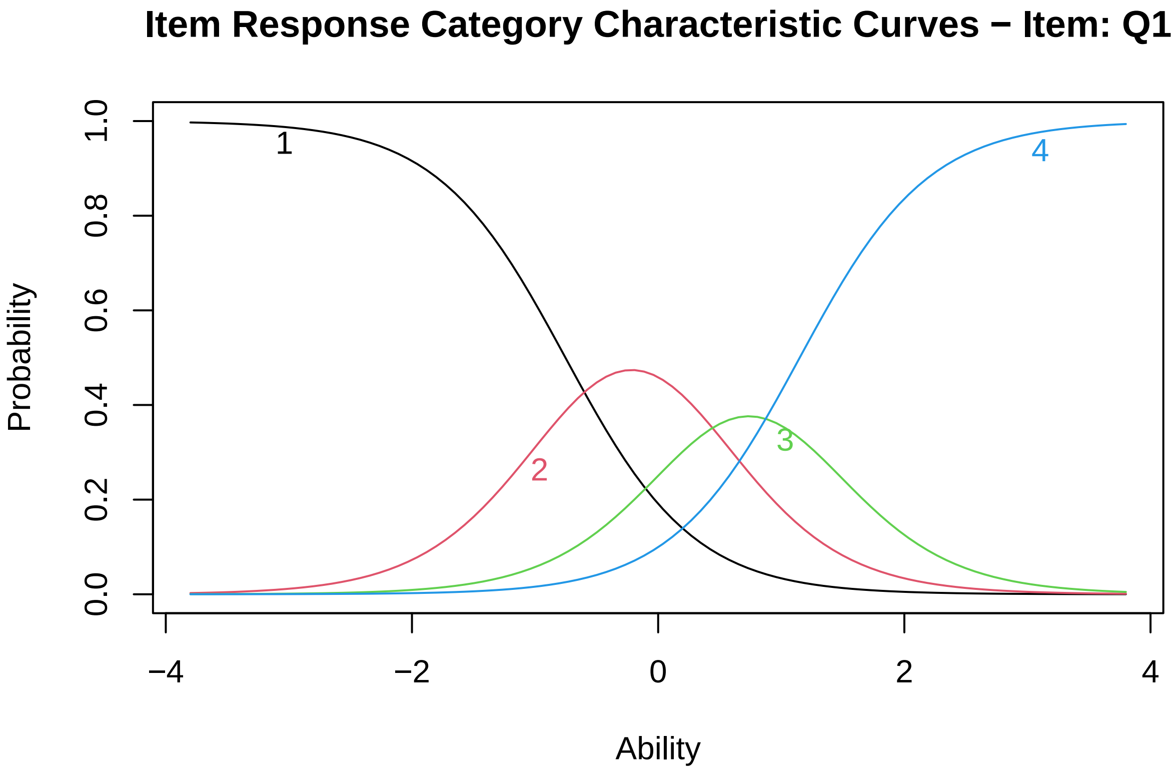
```
#DIF
{
  data_dif_age = dich[,SCS_vars]
  data_dif_age$age = dich$age > median(dich$age)
  dif_ageL = difLord(data_dif_age,"age",FALSE,"1PL")
  dif_ageR = difRaju(data_dif_age,"age",FALSE, "1PL")


  data_dif_gender= dich[,c(SCS_vars,"gender")]
  dif_genderL = difLord(data_dif_gender,"gender",1, "1PL")
  dif_genderR = difRaju(data_dif_gender,"gender",1, "1PL")

  difstats=data.frame(
    p=-log10(p.adjust(
      c(dif_genderL$p.value,dif_ageL$p.value),method="fdr")),
    item = c(dif_genderL$names,dif_ageL$name),
    groups = c(rep("gender",10),rep("age",10))
    )


  ggplot(difstats, aes(x=item, y=p, group=groups,col=groups)) +
    geom_point() + geom_line() + theme_clean() + ylab("-log10(p)")+
    geom_hline(yintercept=-log10(.05))+scale_x_discrete(breaks=paste0("Q",1:10),
                                                        limits=paste0("Q",1:10))

  ggsave("DIF_pvals.pdf",width = 4,height = 3)
  }

#alternative model: 2PL
{

  #fit 1PL and 2PL, compare fit
  twoPL_model = ltm(dich[,SCS_vars] ~ z1, IRT.param = TRUE)
  difficulties_2PL = coef(twoPL_model)[,"Dffclt"]
  discriminativities_2PL = coef(twoPL_model)[,"Dscrmn"]
  ICC_2PL = ICC_plot(difficulty = difficulties_2PL,
                     discriminativity = discriminativities_2PL)


  Rasch_vs_twoPL_comparison = anova(rasch_model_ltm, twoPL_model)
  difficulties_1vs2PL = rbind( data.frame(model="Rasch (1-PL)",
                                   item=factor(paste0("Q",as.character(1:10))),
                                   difficulty=as.numeric(difficulties_ltm)),
                        data.frame(model="Birnbaum (2-PL)",
                                   item=factor(paste0("Q",as.character(1:10))),
                                   difficulty=as.numeric(difficulties_2PL)),
                        data.frame(model="CTT",
                                   item=factor(paste0("Q",as.character(1:10))),
                                   difficulty=as.numeric(dich.distro[1,])/100))
  ggplot(difficulties_1vs2PL,aes(x=item,y=difficulty,
                                                     color=model,group=model)) +
    geom_point() + geom_line() + theme_clean() + ggtitle("model comparison")
  ggsave("difficulties_plot_2PL.pdf",width = 6,height = 8)

  }

#alternative models: bifactor, ...
{

}
```

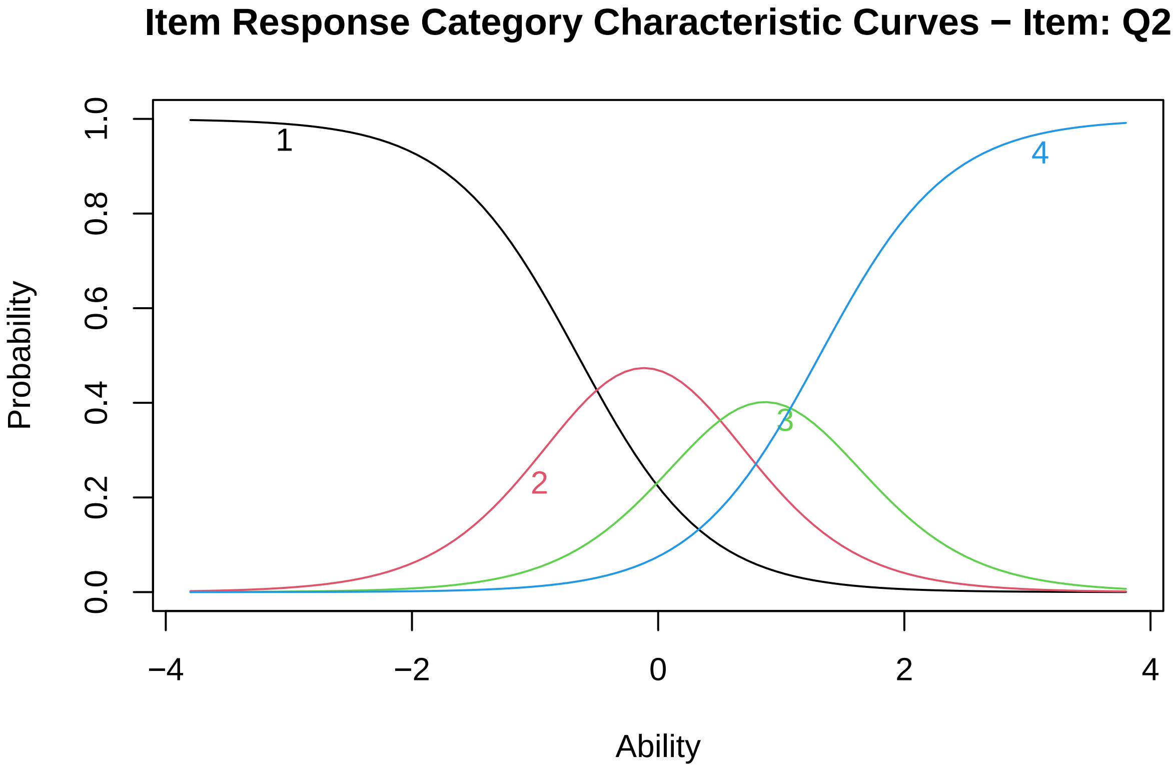
```
## NULL
```

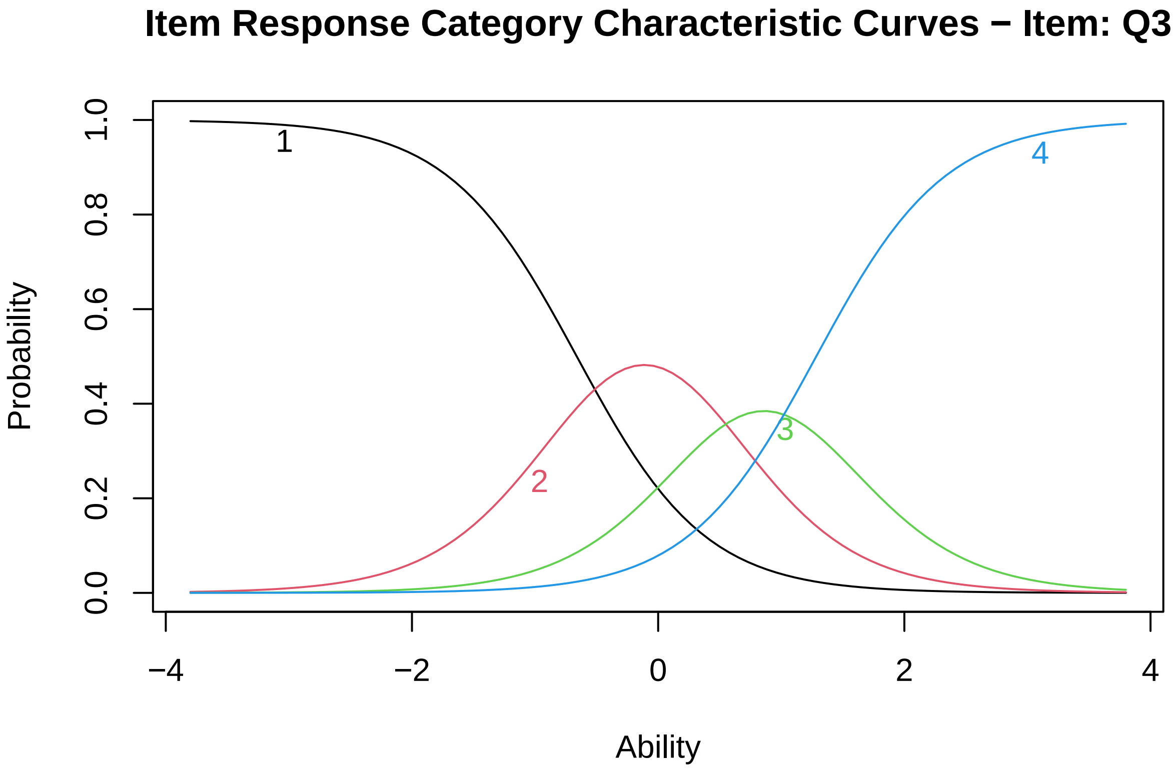
```
#reliability, unidimensionality
{


}
```

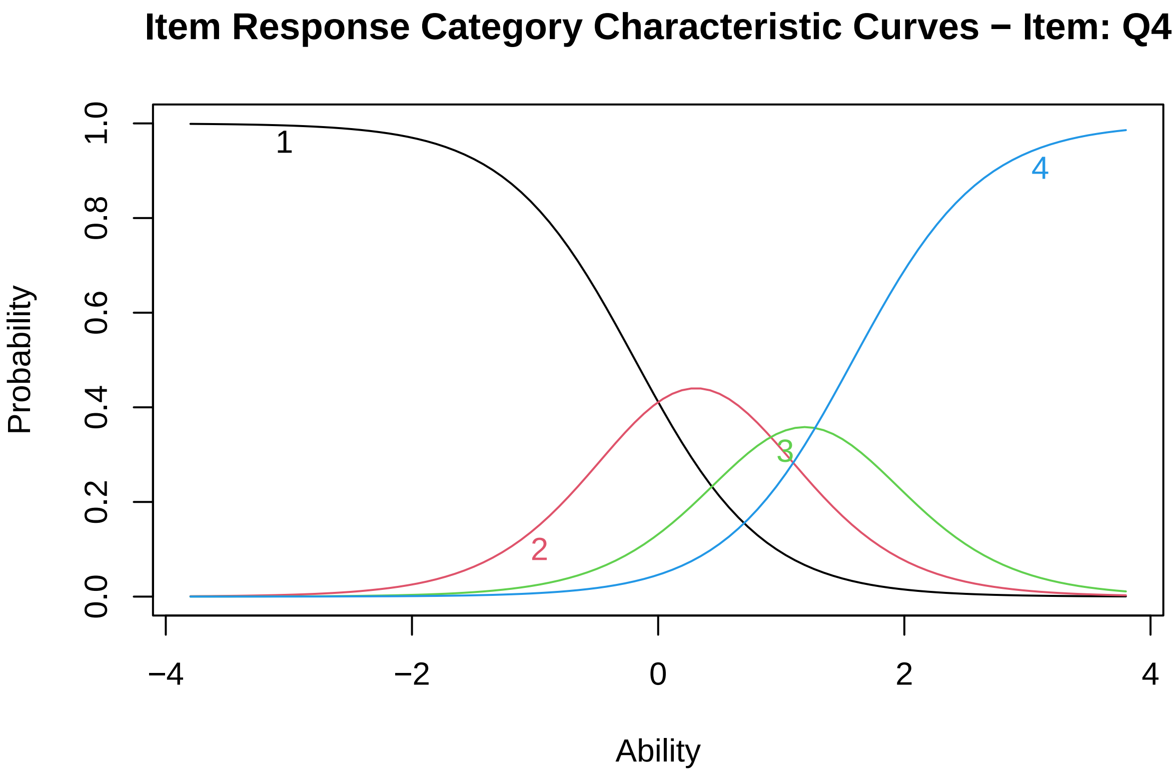
```
## NULL
```

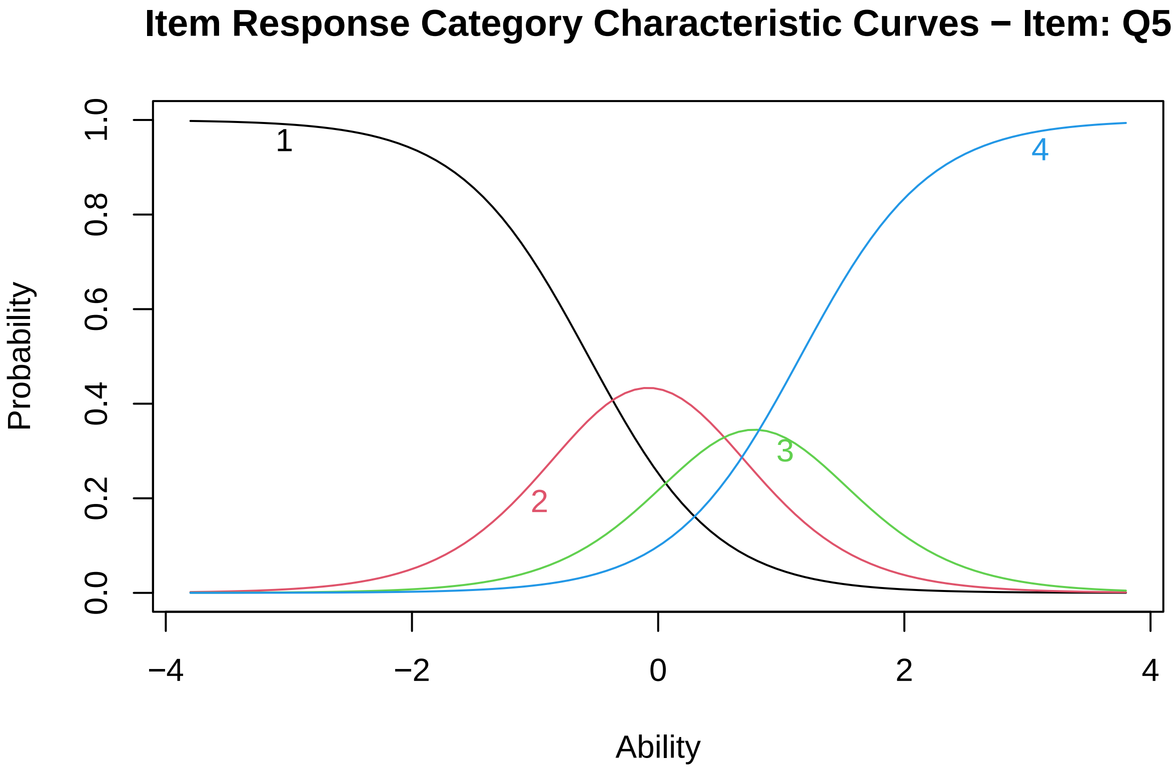
```
#polytomous IRT model
{
  grm_model = grm(df_clean[,SCS_vars],constrained=T)
  plot(grm_model)
}
```

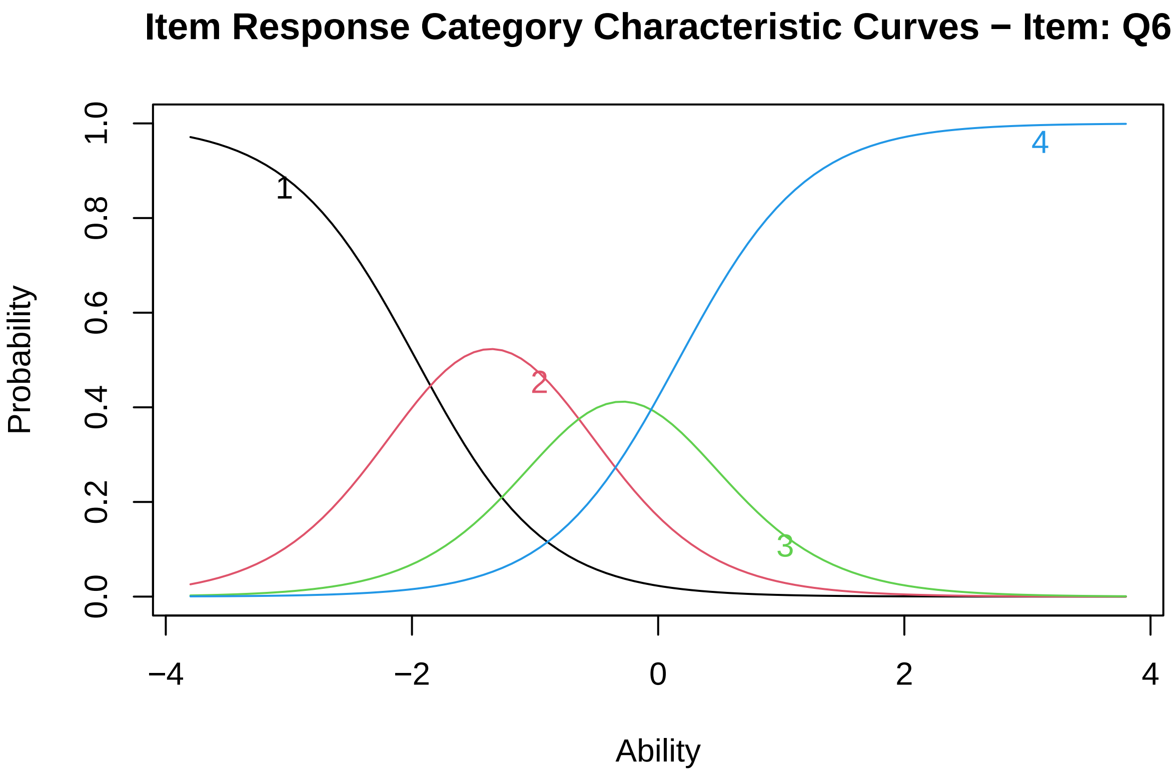


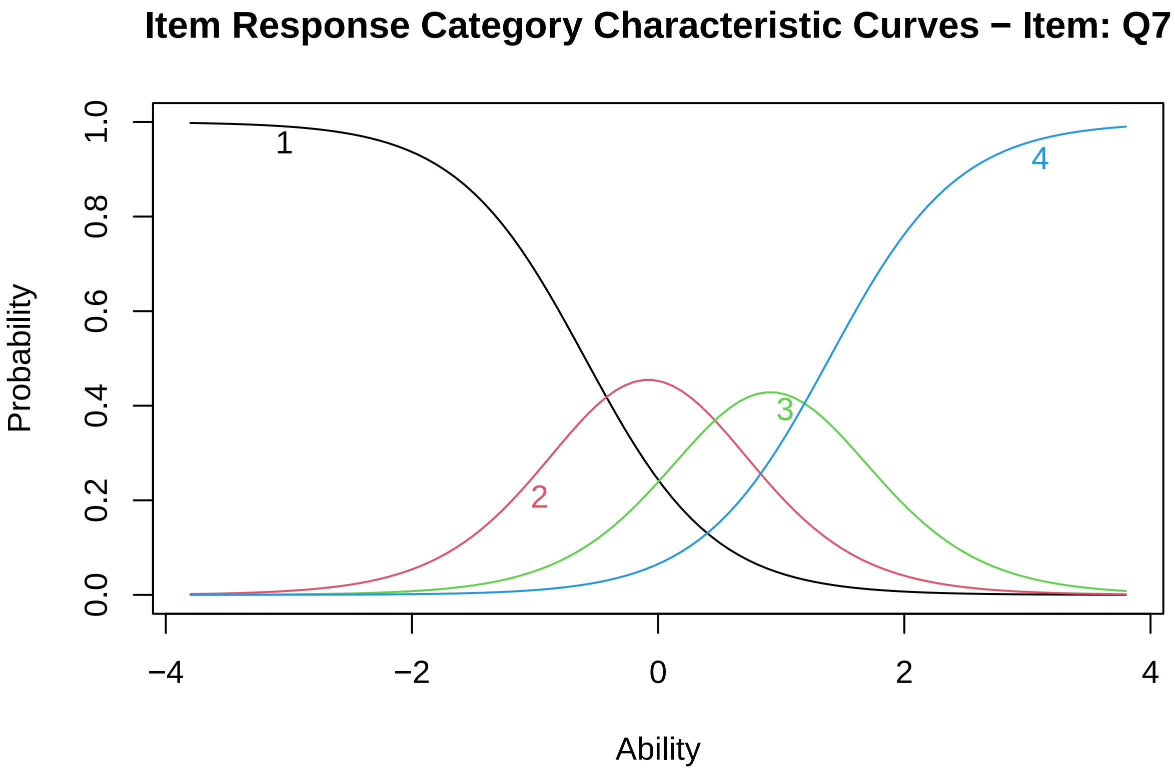

Item Response Category Characteristic Curves − Item: Q2

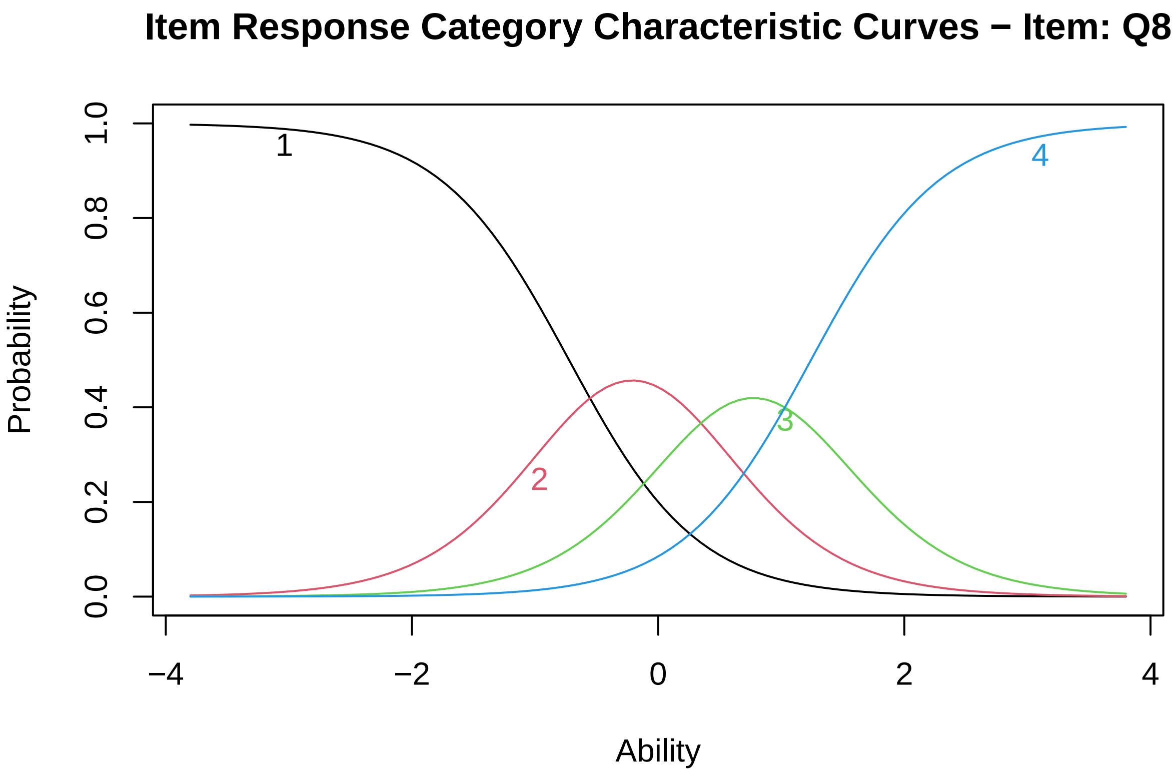

Item Response Category Characteristic Curves − Item: Q3


Item Response Category Characteristic Curves − Item: Q4

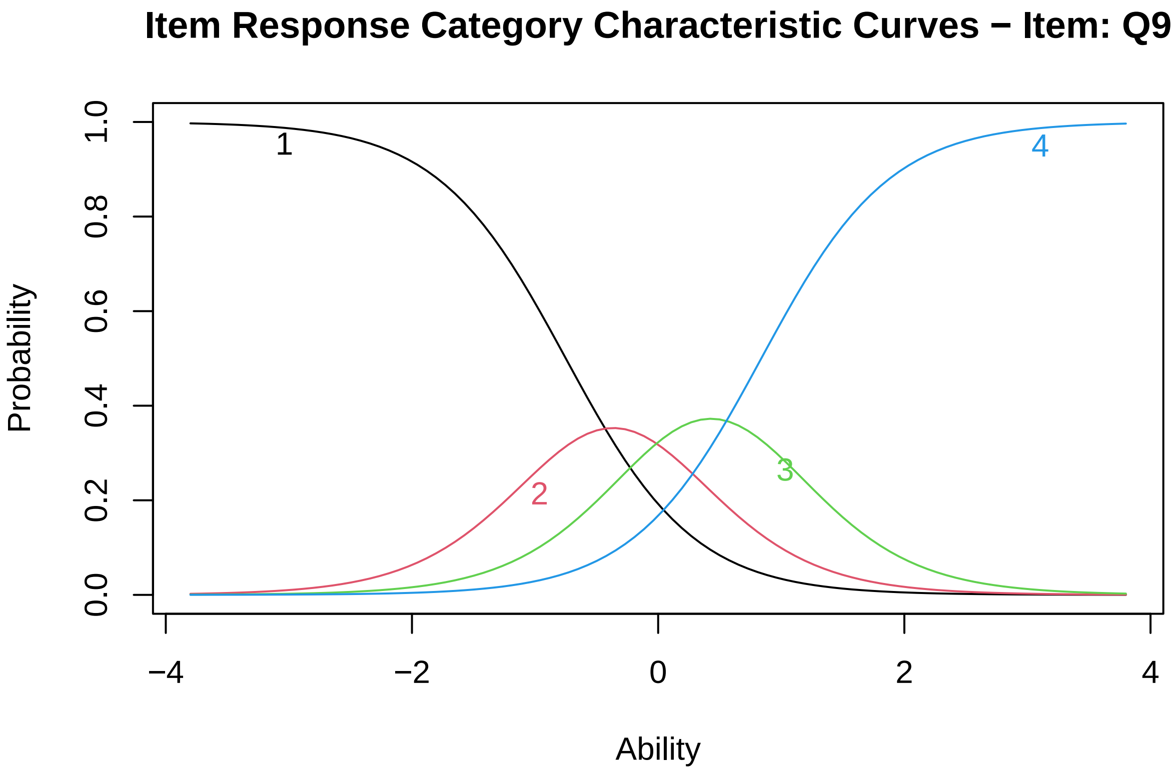

Item Response Category Characteristic Curves − Item: Q5


Item Response Category Characteristic Curves − Item: Q6
Probability
0.0
0.2
0.4
0.6
0.8
1.0
−4
−2
0
2
4
Ability
1
2
3
4

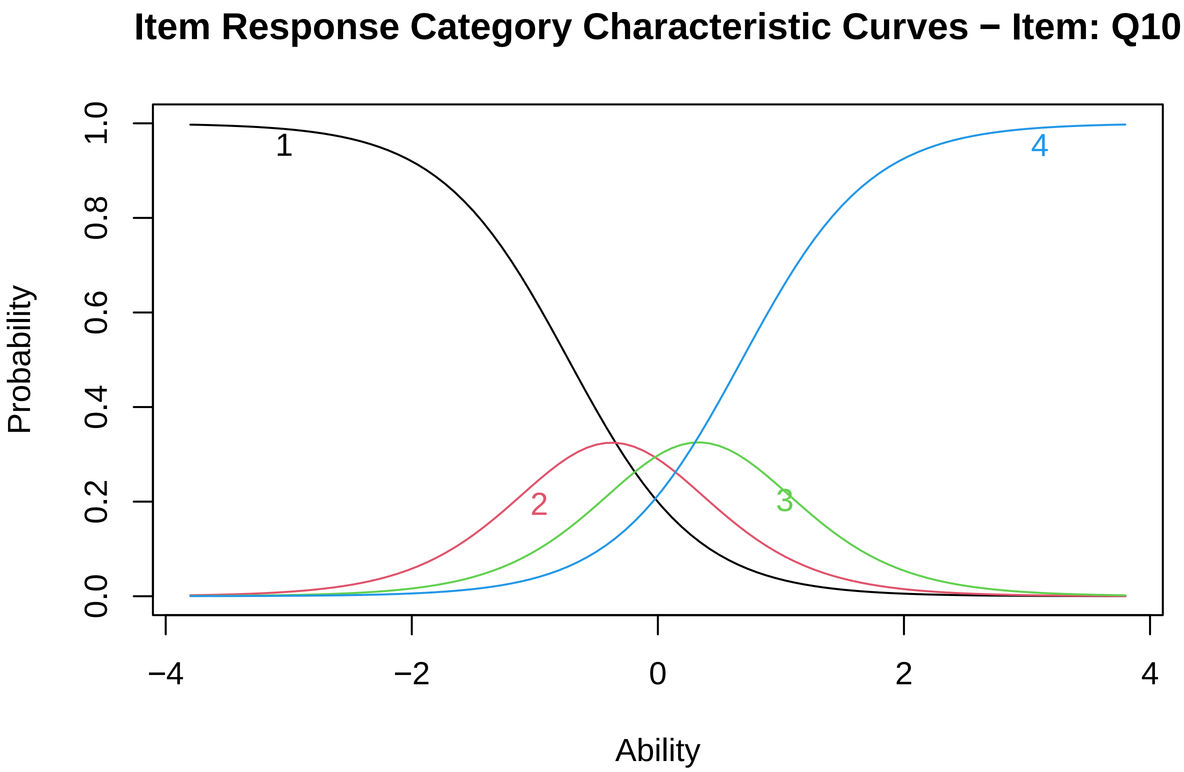

Item Response Category Characteristic Curves − Item: Q7
Probability
0.0
0.2
0.4
0.6
0.8
1.0
−4
−2
0
2
4
Ability
1
2
3
4



Item Response Category Characteristic Curves − Item: Q8



Item Response Category Characteristic Curves − Item: Q9


Item Response Category Characteristic Curves − Item: Q10
Probability
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
4
−4
−2
0
2
4
Ability

14 References

Crocker, Linda, and James Algina. 2008. *Introduction to Classical and Modern Test Theory.* Cengage Learning.

Guan, Ng Chong, and Muhamad Saiful Bahri Yusoff. 2011. “Missing Values in Data Analysis: Ignore or Impute?” *Education in Medicine Journal* 3 (1).

Kalichman, Seth C, and David Rompa. 1995. “Sexual Sensation Seeking and Sexual Compulsivity Scales: Validity, and Predicting HIV Risk Behavior.” *Journal of Personality Assessment* 65 (3): 586–601.

———. 2001. “The Sexual Compulsivity Scale: Further Development and Use with HIV-Positive Persons.” *Journal of Personality Assessment* 76 (3): 379–95.

Magis, David, Sébastien Béland, Francis Tuerlinckx, and Paul De Boeck. 2010. “A General Framework and an r Package for the Detection of Dichotomous Differential Item Functioning.” *Behavior Research Methods* 42 (3): 847–62.

Mair, Patrick, and Reinhold Hatzinger. 2007. “Extended Rasch Modeling: The eRm Package for the Application of IRT Models in r.”

Novick, Melvin R. 1965. “The Axioms and Principal Results of Classical Test Theory.” *ETS Research Bulletin Series* 1965 (1): i–31.

Rasch, Georg. 1960. “Studies in Mathematical Psychology: I. Probabilistic Models for Some Intelligence and Attainment Tests.”

Reynolds, Cecil R, and RA Livingston. 2021. *Mastering Modern Psychological Testing.* Springer.

Rizopoulos, Dimitris. 2006. “Ltm: An r Package for Latent Variable Modelling and Item Response Theory Analyses.” *Journal of Statistical Software* 17 (5): 1–25. https://doi.org/10.18637/jss.v017.i05.

Rosseel, Yves. 2012. “Lavaan: An r Package for Structural Equation Modeling.” *Journal of Statistical Software* 48: 1–36.

Smyth, Rachael. 2022. “Item Response Theory for Polytomous Items.” https://www.uwo.ca/fhs/tc/labs/12.P

Templin, Jonathan. 2022. “IRT Estimation with r Packages Mirt and Lavaan.” https://jonathantemplin.com/irt-estimation-packages-mirt-lavaan/.

Van der Linden, Wim J, and RK Hambleton. 1997. “Handbook of Item Response Theory.” *Taylor & Francis Group. Citado Na pág* 1 (7): 8.