Item Response Theory - Final Essay

Marius Keute

September 26, 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.

The full R and Markdown code used for generating this essay is available on Github:

https://github.com/mkeute/IRT-essay

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)):

Q1. My sexual appetite has gotten in the way of my relationships.

Q2. My sexual thoughts and behaviors are causing problems in my life.

Q3. My desires to have sex have disrupted my daily life.

Q4. I sometimes fail to meet my commitments and responsibilities because of my sexual behaviors.

Q5. I sometimes get so horny I could lose control.

Q6. I find myself thinking about sex while at work.

Q7. I feel that sexual thoughts and feelings are stronger than I am.

Q8. I have to struggle to control my sexual thoughts and behavior.

Q9. I think about sex more than I would like to.

Q10. 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), available at http://openpsychometrics.org/_rawdata/SCS.zip) 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:

$$P(M_{i,q} = 1 | X_{i,q'}) = \sigma(X_{i,q'} \hat{\beta}_{q'}),$$

where $M_{i,q}$ is 1 if the i^{th} person has a missing value at item $q \in \{Q1, Q2, ...Q10\}$ and 0 otherwise, $X_{i,q'}$ denotes the item values of all other items except item q, σ is the logistic function $\sigma(x) = \frac{1}{1-e^{-x}}$, and $\hat{\beta}_{q'}$ are the estimated regression weights based on all other items (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 cases with any missing values 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.

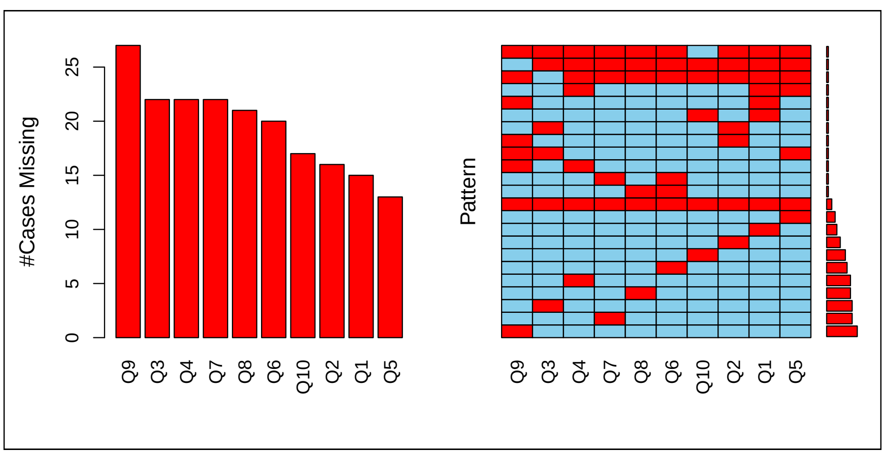


Figure 1: Pattern of missing SCS values.

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 potential ceiling tendency with item Q6 (few cases with response 1, many with response 4).

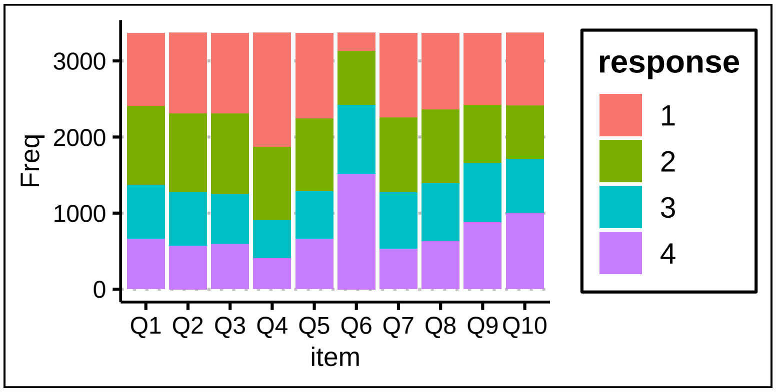


Figure 2: Distribution of non-dichotomized responses per item

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. Likewise, I will refer to the latent person scores that the model estimates as ‘ability’, again for terminological compliance, while they really do not indicate an ability but rather the sexual compulsivity trait that the SCS is supposed to measure.

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).

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)

X	stat	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	max	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
2	mean	2.3	2.2	2.2	1.9	2.2	3.1	2.2	2.3	2.5	2.5
3	median	2.0	2.0	2.0	2.0	2.0	3.0	2.0	2.0	2.0	3.0
4	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

X	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
item easiness (percent in category 1)	40.50	37.80	37.30	27.10	38.20	71.90	37.70	41.20	49.40	50.80
number of cases in category 1	1365.00	1274.00	1255.00	914.00	1285.00	2423.00	1269.00	1389.00	1663.00	1711.00
discrimination	0.45	0.45	0.44	0.34	0.29	0.26	0.42	0.37	0.31	0.36

Tetrachoric intercorrelations of the (dichotomized) items 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 (see below for further scrutiny of factorial structure). 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, in and of itself, a very good representation of the entire scale, since all item discriminations were only moderate in size.

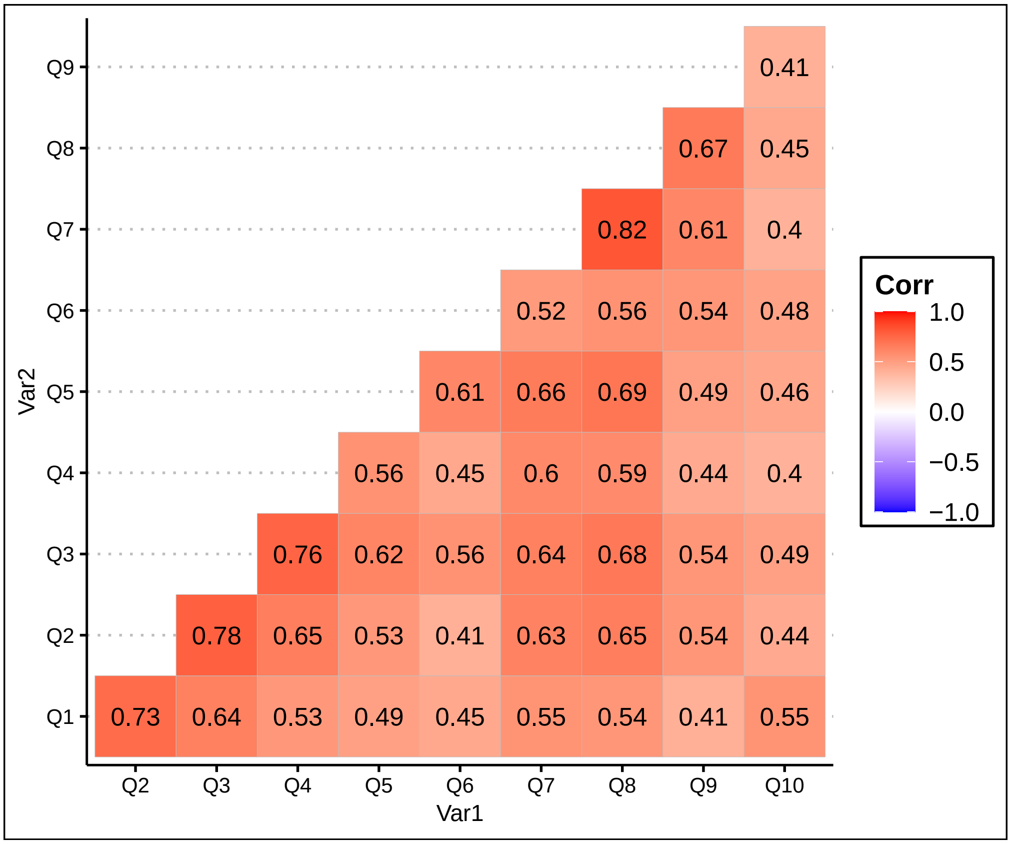


Figure 3: Tetrachoric intercorrelations between items.

4 Rasch models

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, Rasch model assumes that this relationship is identical for all items, i.e., the logistic function can only be shifted in threshold, 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. Of note, due to limitations of the implementation, `lavaan` is not able to estimate Rasch models using maximum likelihood estimation, but only using mean- and variance-adjusted weighted least squares (WLSMV) estimation, which limits model fit comparisons. The `lavaan` model did not give back IRT-compatible difficulty (β) coefficients immediately, but they had to be calculated by dividing the items estimated `tau` coefficients by the respective `lambda` coefficients.

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 in the data). 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.

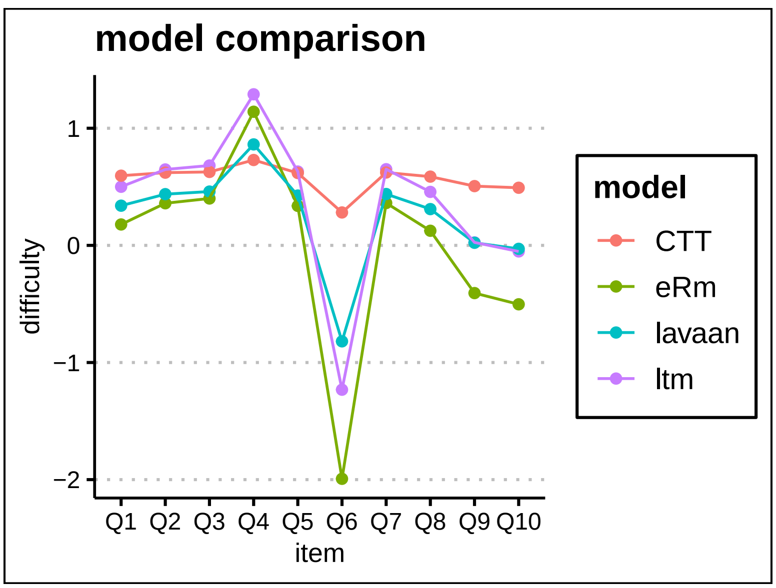


Figure 4: Item difficulties in comparison.

The overall likelihood-based model fit indices are shown in Table3. Of note, log-likelihood and information criteria can only be reported for those models fitted using `ltm` and `eRm`, while the `lavaan` model's fit indices are not comparable, as it was not fitted using maximum likelihood estimation. For brevity, I skip the discussion of the `lavaan` model fit indices. Comparing the fit indices for the `ltm` and `eRm` Rasch models, it can be seen that `eRm` had an overall higher log-likelihood. Since it also had one free parameter less (because of the sum constraint, see above), it was, overall, the preferred model also according to the Akaike and Bayes-Schwarz information criteria.

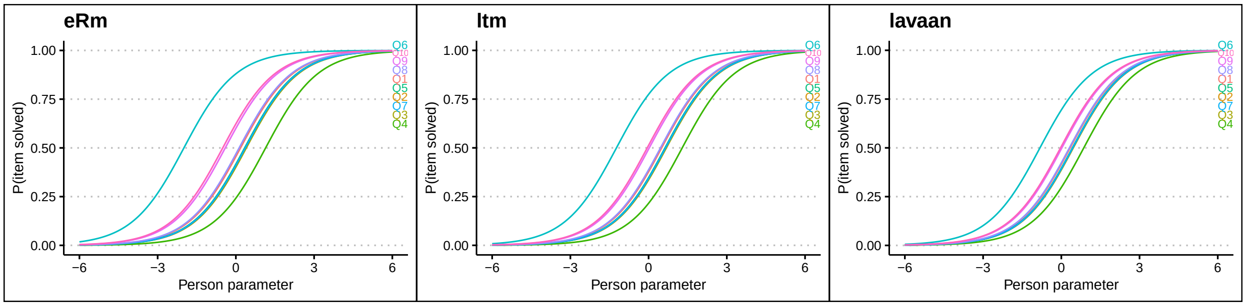


Figure 5: Item-characteristic curves for the three Rasch models.

Table 3: Fit indices for Rasch models

X	loglik	npar	AIC	BIC	cAIC
eRm	-17893.02	9	35804.04	35859.14	35868.14
ltm	-18566.79	10	37153.59	37214.81	NA

Finally, to get an impression of how the three models perform for each item, I calculated the mean 0-1-loss per item, compared to the actual, dichotomized data $D_{i,q}$, as:

$\mathcal{L}_q = \frac{1}{n}\sum_{i=1}^{n} |f(\theta_i, \beta_q) - D_{i,q}|$

that is to say, I used the item difficulty parameters β_q and person ability scores θ_i estimated by the models to predict the expected response for each person and item:

$$f(\theta_i, \beta_q) =: \begin{cases} 1 \text{ if } \sigma(\theta_i - \beta_q) > 0.5, \\ 0 \text{ otherwise} \end{cases}$$

The mean loss is then calculated as the proportion of incorrectly predicted cases for each item. It is shown in Figure 6. It can be seen that the three models, despite differences in parameterization, performed very similarly, with the `eRm` and `lavaan` models performing almost identically, whereas the `ltm` model tended to incur a slightly higher loss, except for items Q6 and Q10. Interestingly, these are the most difficult items, and the fact that `ltm` outperformed `eRm` especially for those items might be related to the differences between conditional vs. marginal maximum likelihood estimation, which have the strongest effect in cases where either all or no responses are correct, i.e., for particularly easy or difficult items.

4.3 Differential Item Functioning

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 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 7. 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).

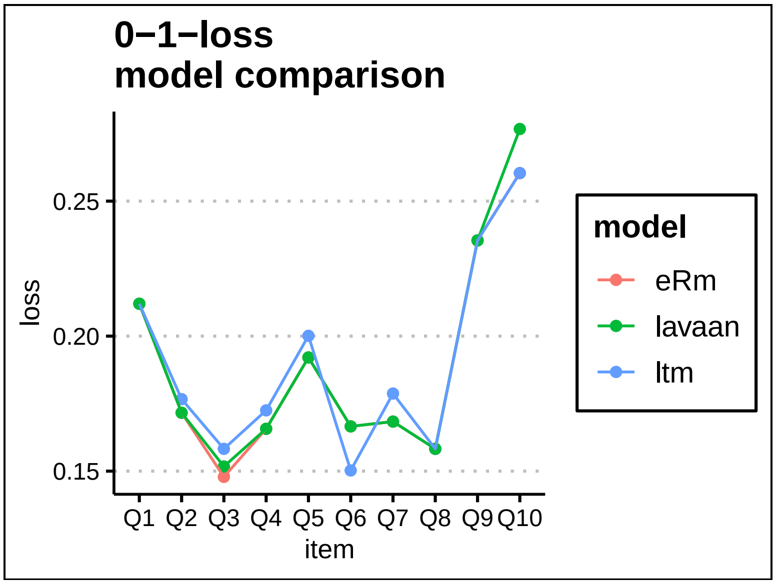


Figure 6: Mean 0-1-loss (proportion of incorrectly predicted responses) per item.

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 ($\Delta_{Lord} = -1.14$). Item Q5 reads 'I sometimes get so horny I could lose control', and it appears plausible that DIF for this item could emerge by gender-specific societal norms, and possibly gender-specific consequences of 'being horny' for everyday functioning. Since I have only been considering Rasch (one-parameter) models so far, this analysis only tested for uniform DIF.

5 Higher-parameterized 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.

For fitting the 2-PL model, I used the `ltm::ltm` function, and the Rasch model fitted using `ltm::rasch` served as a baseline model for comparison. ICCs and estimated item difficulty parameters are shown in Figure 8 and 9, respectively. It can clearly be seen that item discriminativities, and with them the slopes of the ICC curves, vary considerably between items. A side-effect of two-parameter modeling is that the ICCs now cross each other, i.e., there is no clear order of difficulty between items anymore, but whether one item is more difficult than another can now be dependent on the person ability.

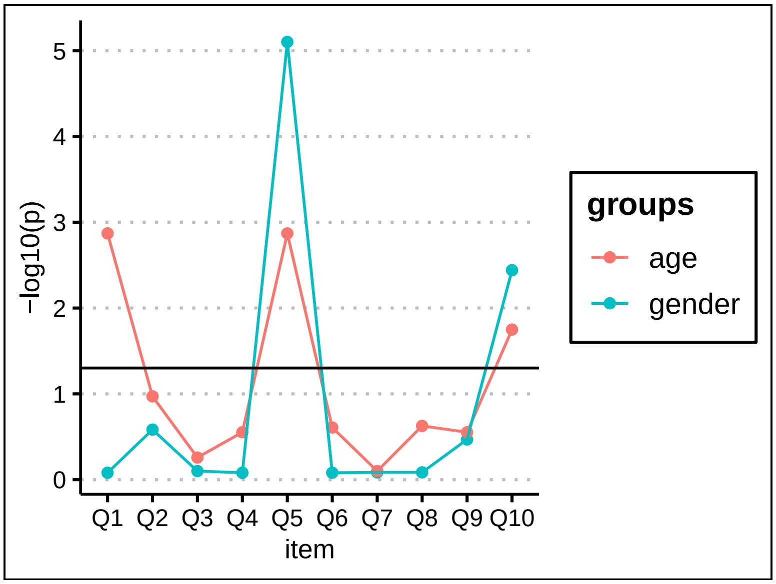


Figure 7: p-values (FDR corrected, negative log-transformed) from DIF analysis. Black line: significance threshold (p $<$ .05)

Comparing the two models using a Likelihood Ratio Test, I found the 2-PL model to fit the data significantly better than the Rasch model (log-LR = 1741.55, p $<$ .001).

For further model comparison, I calculated infit and outfit indices for each item and model. Infit and outfit indices are based on model residuals. While outfit (short for outlier-sensitive fit statistic) is particularly sensitive to unexpected responses in cases where item difficulty and person ability are far apart (e.g., a low-ability person unexpectedly solves several very difficult items), infit (short for information-weighted fit statistic) is particularly sensitive to unexpected responses in cases where item difficulty and person ability match (e.g., a person solves far less or far more than half of the items whose difficulty equals their ability). Both are based on the normalized residuals $Z_{iq} = \frac{D_{iq} - \mathbb{E}(D_{iq})}{\sqrt{var(D_{iq})}}$, where D_{iq} is the actual response of person i to item q, $\mathbb{E}(D_{iq}) = P(D_{iq} = 1|\beta_q, \theta_i)$ is the conditional expectation for this person's response to the item, given the model parameters, calculated using the logistic function as shown above. $var(D_{iq})$ can be calculated as $P(D_{iq} = 1|\beta_q, \theta_i)(1 - P(D_{iq} = 1|\beta_q, \theta_i))$. The infit index for item q is then defined as: $Infit_q = \sum_{i=1}^{n}(\frac{var(D_{iq})}{\sum_{i=1}^{n} var(D_{iq})} Z_{iq}^2)$, the outfit index is defined as: $Outfit_q = \sum_{i=1}^{n} \frac{Z_{iq}^2}{n}$. For both indices, values close to 1 indicate good fit, whereas higher values indicate under- and lower values overfitting.

The infit and outfit values for both models are, for the most part, in the acceptable (>0.7 and <1.3) range (see Figure 10). For infit, the 2-PL model consistently outperforms the Rasch model, whereas the outfit values tend to be lower overall, and both models are much closer to each other.

6 Polytomous IRT model

Next, I went back to the original, non-dichotomized data and fitted a polytomous IRT model to it, i.e., a model that accounts for more than two response categories per item and therefore takes the full richness of the data into account. The first step in polytomous IRT modeling is choosing the appropriate model class, depending on the structure of the response alternatives. In the case of

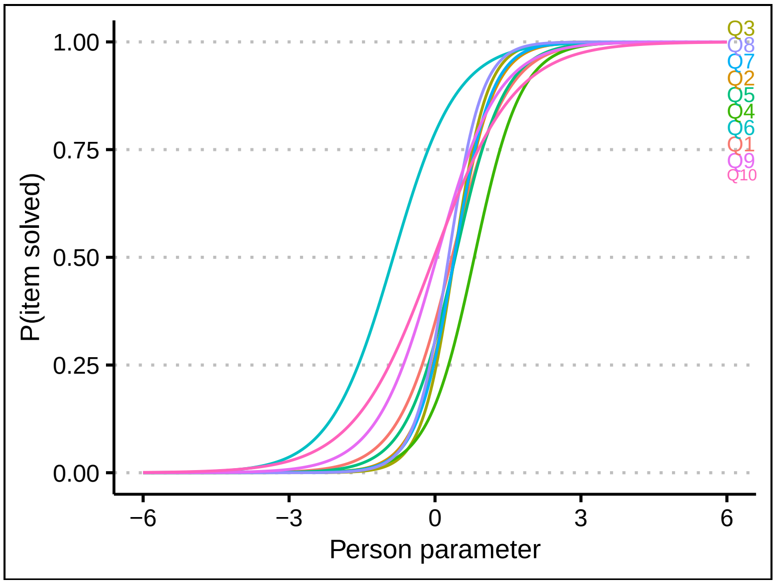


Figure 8: Item-Characteristic Curves from 2-PL model

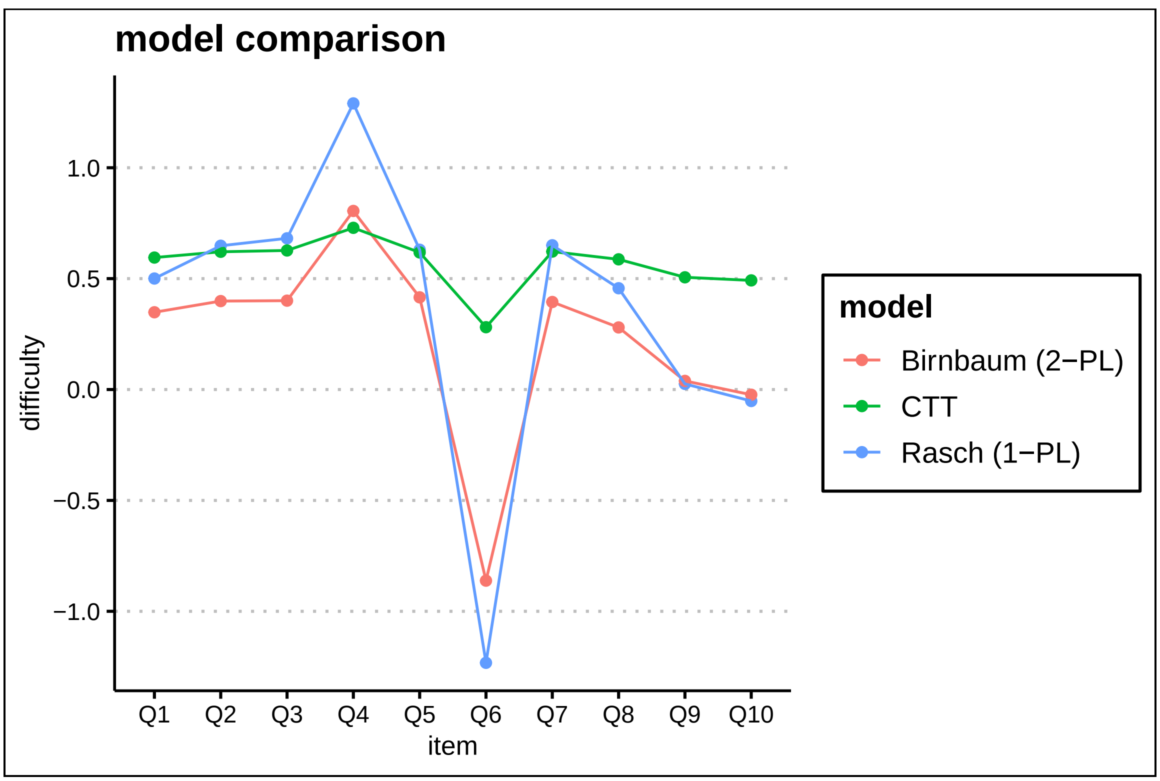


Figure 9: Estimated item difficulties and discriminativities based on CTT, Rasch model and 2-PL model. For both IRT models, error bars indicate standard errors.

Figure 10: Infit and outfit indices for each item in the Rasch and 2-PL models

the SCS, responses are ordered, ranging from 1 to 4, indicating monotonously increasing degrees of agreement with the item (or correctness, in IRT terminology), therefore a graded response model (GRM) seemed appropriate. I used the `ltm::grm` function, following the procedure outlined in Smyth (2022).

The first decision to be made in the GRM was whether to use an unconstrained or constrained model, i.e., whether or not to allow item discriminativity to vary between items. I found that an unconstrained model was a significantly better fit to the data (LRT = 821.42, $p<.001$), analogously to the dichotomized data models, where the 2-PL model with varying item discriminativities was also preferred over the Rasch model.

Exemplary ICCs for item Q2, i.e., the item with the highest discriminativity in the dichotomized case, are shown in Figure 11). There is now one ICC curve for each response category, while the y-axis still gives the probability for a person with a given latent ability level to give a response in a certain category. One can see nicely that the response categories follow the monotonous order that is intended by design.

Additionally, Item Information Curves can be generated, as shown in Figure 12). The figure contains one IIC curve per item, indicating how informative each item is with respect to the overall sum score as a function of the latent person ability. It is, therefore, roughly equivalent to item discriminativity, additionally resolved by person ability levels. It can be seen that the majority of items are most informative at a medium level of person ability, and that items Q8, Q3, Q7, and Q2, are the most informative items in the medium ability range. Item Q6, on the other hand, tends to be relatively informative also in the lower ability range, while item Q4 extends its region of high informativity slightly higher into the high-ability range than the other items.

7 Factor models

Following the IRT analysis of the dichotomized data, I went back to the original, non-dichotomized data, to investigate its factorial structure. It has been suggested that the SCS can best be described by two latent factors, one comprising items Q1, Q2, Q3, Q4, and Q10, being related to consequences of sexual behavior and compulsivity to one's lifestyle, and a second one comprising items Q5, Q6, Q7, Q8, and Q9, being related to the compulsivity of one's sexual thoughts without necessarily affecting

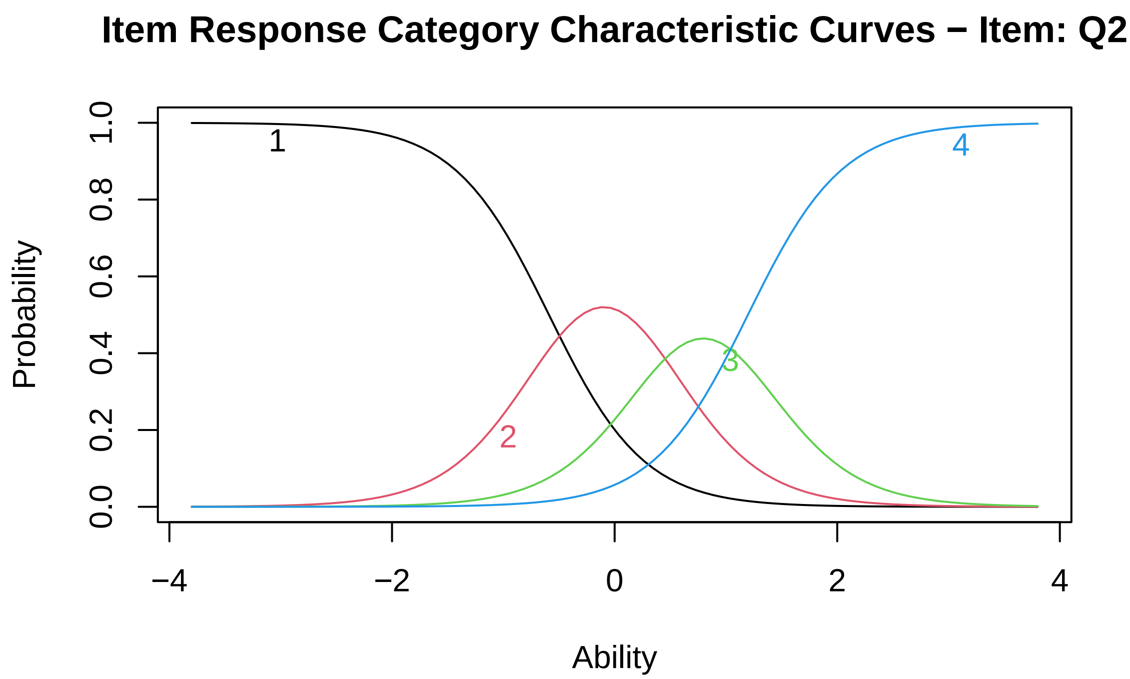


Figure 11: Item-Characteristic curves from polytomous GRM for item Q2

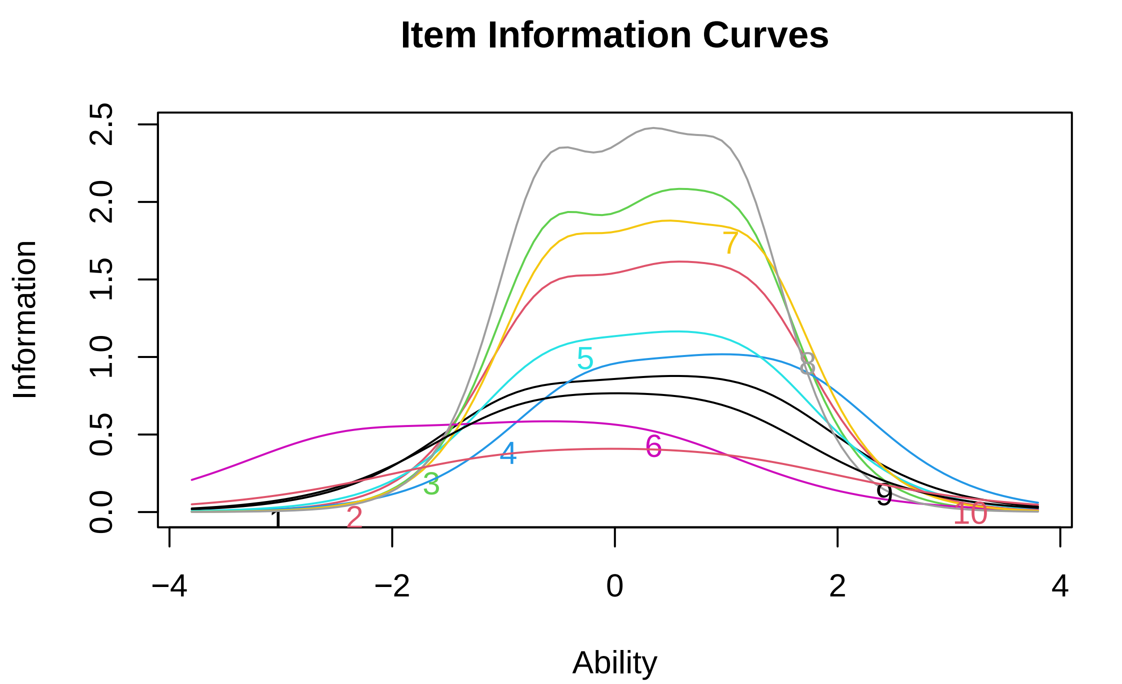


Figure 12: Item Information curves from polytomous GRM for item Q2

actual behavior. Using `lavaan::cfa`, I fitted several confirmatory factor analysis (CFA) models to the data to find the latent structure that describes the data best. I specified four candidate latent structures:

The first candidate structure was a unidimensional model, i.e., the data can be explained by a single underlying latent factor.

The second candidate structure was a two-factor correlated-traits model, i.e., two latent factors were specified with item loadings as described above, and correlations between the two latent factors were allowed.

The third candidate structure was a bifactor model, i.e., two latent factors were specified with item loadings as described above, with an additional general factor on which all items load. The general factor was constrained to be orthogonal to the subfactors.

The final candidate structure was a hierarchical factor model, which specifies the two item-specific factors, and additionally has them load on a shared second-order factor.

Models were fitted with standardized latent variables, i.e., the variance of all latent factors was fixed to unit. The models were specified in `lavaan` syntax as follows:

```
Unidimensional model:
  xi1 =~ Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9+Q10

Correlated-traits model:
  xi1 =~ Q1+Q2+Q3+Q4+Q10
  xi2 =~ Q5+Q6+Q7+Q8+Q9

Bifactor model:
  G =~ Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9+Q10
  xi1 =~ Q1+Q2+Q3+Q4+Q10
  xi2 =~ Q5+Q6+Q7+Q8+Q9
  G ~~ 0*xi1
  G ~~ 0*xi2

Hierarchical model:
  xi1 =~ Q1+Q2+Q3+Q4+Q10
  xi2 =~ Q5+Q6+Q7+Q8+Q9
  G =~ xi1+xi2
```

The comparison of fits between the four factor models is shown in 4. Models were compared with respect to the Akaike (AIC) and Bayes-Schwarz (BIC) information criteria. Among the four candidate models, the bifactor model was clearly the preferred one among the four candidate models according to AIC as well as BIC.

The 'winning' bifactor model is illustrated in Figure 13. Obviously, the fact that the bifactor model is the preferred option among the four candidate models does not mean that it is necessarily a good description of the data in an absolute sense. To understand the absolute goodness-of-fit (not just compared to other models), there is a range of fit indices that we can consider. In particular, the root mean squared error of approximation (RMSEA), standardized root mean squared residual (SRMR), comparative fit index (CFI), and Tucker-Lewis index (TLI) are informative. For the bifactor model, RMSEA was at 0.073 (RMSEA < 0.08 indicating acceptable, RMSEA < 0.05 indicating good fit

Table 4: Model comparison between CFA models

X	model	Df	AIC	BIC
1	hierarchical	33	85601.06	85735.75
2	bifactor	24	85200.96	85390.74
3	correlated traits	34	85599.06	85727.62
4	unidimensional	35	86567.75	86690.19

by convention), SRMR was at 0.028 (SRMR < 0.05 indicating good fit by convention), CFI was at 0.973 (CFI > 0.95 indicating good fit by convention), and TLI was at 0.95 (TLI > 0.95 indicating acceptable, TLI > 0.97 indicating good fit by convention). Overall, the bifactor model was, therefore, an acceptable to good fit for the SCS data.

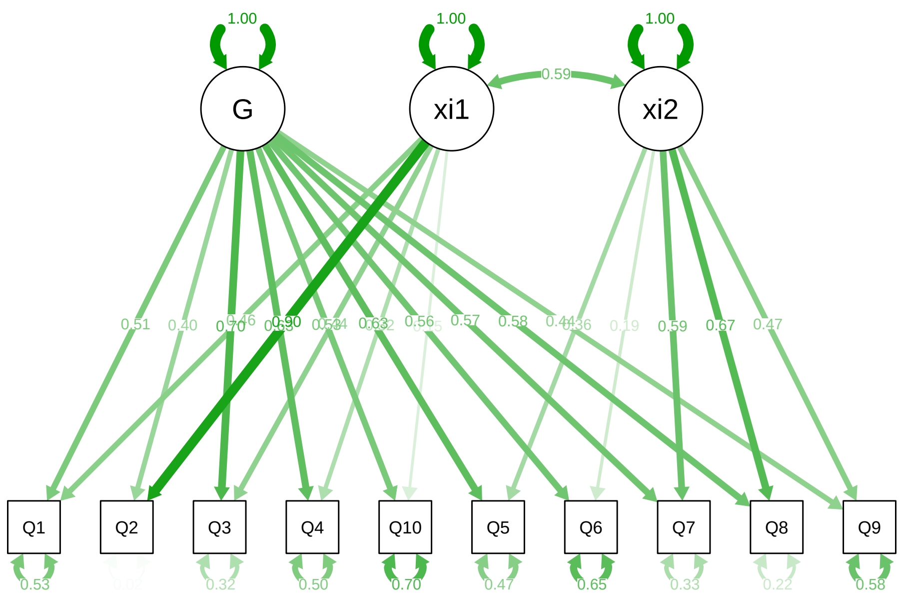


Figure 13: Factor structure and loadings of bifactor model

An open question with respect to the factorial structure is to which subscale item Q10 should belong. While it has been assigned to the first subfactor, its loading on the factor is low (0.18, around half as high as the second-lowest loading item, Q4). To investigate the issue, I fitted two alternative bifactor models, one (Alternative 1) where item Q10 belonged to the second subfactor, together with items Q5 - Q9, and one (Alternative 2) where item Q10 constituted its own, third subfactor. Additionally, I used the `psych::omega` function to fit a bifactor model with automatized identification of the factor structure. This procedure also returned a three-factor structure, where the first four items loaded on one subfactor, items Q5 - Q9 loaded on another subfactor, and items Q1, Q6, and Q10 loaded on a third subfactor (see Figure 14). I refitted this model in `lavaan` to facilitate comparisons.

Considering all fit indices reported above (see Table 5), the original factor structure was clearly preferred over the Alternative 1 bifactor model, but the Alternative 2 bifactor model was a close competitor to the original bifactor model, that is to say, item Q10 arguably constitutes a third subfactor on its own. The automatically identified factor structure, however, was clearly preferred over the original model by all fit indices.

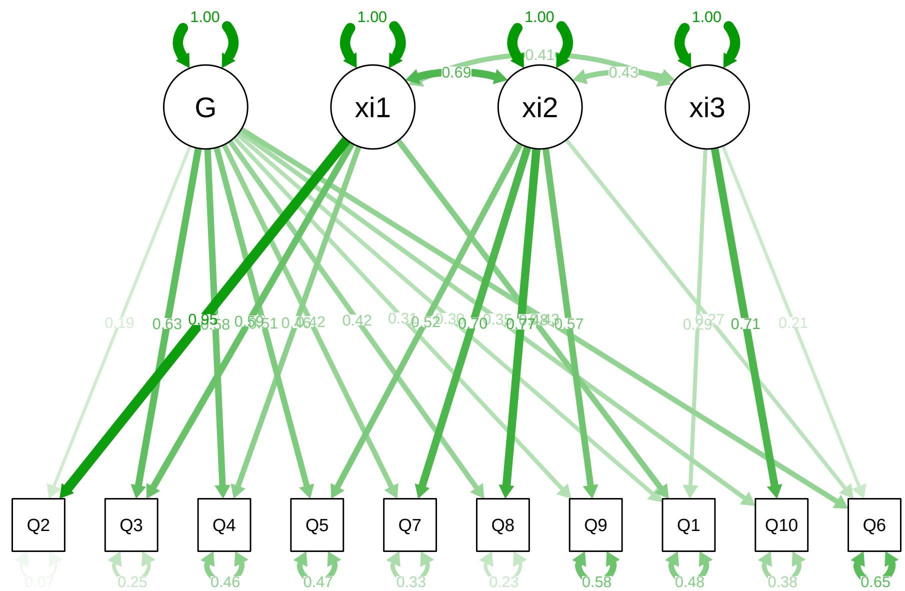


Figure 14: Factor structure and loadings of automatically identified bifactor model

Table 5: Model comparison between different bifactor models

X	npar	aic	bic	cfi	tli	srmr	rmsea
Bifactor (original)	31	85200.96	85390.74	0.9732703	0.9498819	0.0280486	0.0732652
Bifactor (Alternative 1)	31	85214.73	85404.51	0.9724219	0.9482911	0.0308382	0.0744188
Bifactor (Alternative 2)	33	85201.26	85403.29	0.9733748	0.9455394	0.0286056	0.0763733
Bifactor (Automatized)	35	84955.38	85169.65	0.9886453	0.9744519	0.0187802	0.0523093

Looking at the content of the items (see Introduction), we can see that item Q10 is the only item that explicitly involves sex partners and difficulties to find sex partners, while item Q1 mentions relationships, item Q6 mentions the work environment, but all other items do not explicitly refer to relationships with other persons. It could, therefore, be, that responses to those items are not only influenced by sexual compulsivity, but also by a range of social and communicative abilities that might influence whether someone has difficulties finding sex partners and getting along with relationship partners and coworkers or not. Therefore, it appears plausible that those three items apparently form a third subfactor.

8 Reliability and Unidimensionality

As outlined above, the SCS is better described by a bifactor model with two or three subfactors than by a fully unidimensional model. It can, therefore, be concluded, that the scale is not unidimensional. To determine the composite reliability of the scales with respect to its subscales determined by confirmatory factor analysis, I calculated composite reliability using `semTools::reliability`. Composite reliability is a measure of internal consistency and is related to the question of whether item sum scores are a good representation of the underlying, latent quantity.

The function gives back Cronbach's α as well as the ω_1, ω_2, and ω_3 coefficients both for the original

Table 6: Composite reliability scores for the bifactor models with two and three subfactors

X	G	xi_1	xi_2	xi_3	model
alpha	0.90	0.83	0.84	NA	two-factor
omega	0.88	0.70	0.70	NA	two-factor
omega_2	0.60	0.34	0.34	NA	two-factor
omega_3	0.60	0.34	0.34	NA	two-factor
alpha	0.90	0.84	0.84	0.66	three-factor
omega	0.81	0.83	0.78	0.52	three-factor
omega_2	0.33	0.56	0.53	0.29	three-factor
omega_3	0.33	0.56	0.53	0.29	three-factor

bifactor model with two subfactors (see Figure 13) and for the better-fitting, but more complex bifactor structure with three subfactors (see Figure 14). The results are shown in Table 6.

It can be seen that the estimated reliability differs substantially between scores, and between models. Unsurprisingly, the third subfactor ξ_3 from the three-factor model has relatively low reliability, which might be related to the fact that it is based on fewer items. The remaining factors G, ξ_1 and ξ_2 have acceptable to good composite reliability in both models according to Cronbach's α and ω, ranging from 0.71 to 0.90. Differences between scores are related to the way those coefficients are calculated. All three variants of ω are calculated as a fraction, with the squared sum of loadings of a given factor, multiplied with the factor variance (which is 1 in our models) in the numerator. What differs is the denominator. Different variants of ω are calculated with denominators based on either the residual variance, empirical covariance, or model-implied covariance. The fact that ω_3 is consistently smaller than α could point to relatively equal general factor loadings.

9 Measurement Invariance

As a last step in data analysis, I investigated measurement invariance of the SCS data. Measurement invariance means that the factorial structure is independent of other variables that are not part of the structure. In particular, I tested whether or not the three-subfactor bifactorial structure was identical across genders and age groups, similar to the analysis of DIF (see above). Following the procedure outlined in Xu (2012) and Van de Schoot, Lugtig, and Hox (2012), I compared four models with increasingly strict measurement invariance assumptions. The *first* model assumes *configural invariance*, i.e., model parameters can vary freely in each group, and no assumption about invariance between the groups is made. The *second* model assumes *weak invariance* or *metric invariance*, i.e., equal factor loadings across groups. The *third* model assumes *strong invariance* or *scalar invariance*, where factor loadings and item intercepts are assumed to be equal across groups. The final model assumes *strict* invariance, where, additionally, item residual variances are assumed to be equal across groups. Of note, I had to use the two-subfactor bifactorial structure to test for gender measurement invariance, and the three-subfactor bifactorial structure to test for age group measurement invariance, since otherwise, there would have been always one or several models that did not converge.

For gender, the overall preferred model was the one with weak invariance (see Table 7), while for age, eithert the model with weak or configural invariance was preferred, depending on whether AIC, BIC, or the significance test were considered. (see Table 8).

Looking at modification indices of the weak invariance models for age and gender, I found that the

Table 7: Model comparison for measurement invariance across genders

X	DF	AIC	BIC	Chisq	Chisq_diff	DF_diff	p
configural MI	48	85189.15	85691.16	522.6432	NA	NA	NA
weak MI	65	85177.70	85575.64	545.1978	22.554651	17	0.1643182
strong MI	72	85223.77	85578.85	605.2644	60.066515	7	0.0000000
strict MI	82	85209.85	85503.71	611.3490	6.084678	10	0.8080995

Table 8: Model comparison for measurement invariance across age groups

X	DF	AIC	BIC	Chisq	Chisq_diff	DF_diff	p
configural MI	40	84873.17	85424.16	234.9036	NA	NA	NA
weak MI	58	84890.48	85331.26	288.2057	53.30208	18	0.0000236
strong MI	64	84910.36	85314.41	320.0886	31.88295	6	0.0000172
strict MI	74	84925.39	85268.22	355.1176	35.02897	10	0.0001235

suggested modifications for the gender model included allowing factor loadings for items Q1 and Q5 on the latent factors ξ_1 and ξ_2, respectively, to differ between groups, while for the age model, freeing factor loadings for item Q1, Q5, and Q10 between groups was suggested. This is roughly in agreement with what I found with respect to DIF (see above).

10 Theoretical Part: Key differences between IRT and CTT

10.1 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.

10.2 Core Ideas and Terminology

A test, in the sense CTT as well as IRT use the term, is designed to measure a defined *trait* or *state* of a unit (often a person). The trait/state itself is assumed to be unobservable and is captured by combining several *items* that are thought to be reflective of the trait/state. Items in the test-theoretic sense have defined response categories, which can be either dichotomous (e.g., yes/no or solved/unsolved), or ordered (e.g. I... do not agree / agree a little / fully agree), or multinomial (e.g. my favorite color is... red/blue/green). Multinomial items where the responses can not be ordered, nor dichotomized (e.g., into correct and wrong) are more involved from a theoretical point of view and will not be further discussed here. In the end, all items of a test are usually combined by a linear function, i.e., a weighted sum, to form the test *score*. Any theory of psychological tests is concerned with how test scores come about and how they relate to the actual quantity that the test is supposed to measure.

Tests are described by their *Objectivity*, *Reliability*, and *Validity*. Objectivity is concerned with independence of the test result from the person that administers the test. Reliability means consistency or reproducibility of results, and validity refers to the test being valid, i.e., actually measuring the state or trait it is supposed to measure.

Single items within a test, on the other hand, can be described by their *difficulty* and *discrimination*. Item difficulty is related to the proportion of examinees that solve the item, where *solving* can actually mean to give the correct response if a correct response exists, or could otherwise mean responding 'yes' to a question or responding above a defined threshold if there are more than two response options.

The above terms exist, conceptually, in IRT and CTT alike, even though their mathematical formulation might differ.In the following, I will describe how some of them are defined in CTT and IRT, respectively, and how they are linked to underlying theoretical concepts.

10.3 Assumptions and Problems of CTT

At the heart of CTT is the so-called classical true score model. It states that a given person's test score is an additive combination of the person's *true score* and the *test error*: $X_i = \tau_i + \epsilon_i$. The true score is defined (!) as the expected value $\mathbb{E}X_i$. As a corollary, for the error it holds that $\mathbb{E}\epsilon_i = 0$. CTT assumes that true scores and test errors are uncorrelated within the same test and across tests (the true score of any test is uncorrelated with the error of any other test), and that test errors for the same person across several repetitions of the same test or across different tests are uncorrelated, that is to say, the test is assumed to be equally accurate regardless of the specific examinee and regardless of whether the examinee has a high or low true score (Crocker and Algina (2008)).

Taken together, those assumptions amount to the tenet that the variable of interest is represented by the test scores in an unbiased way, and that only unsystematic test errors prevent tests from representing it perfectly.

While this theory obviously has some shortcomings, which I will discuss below, it also has advantages. With the above assumptions in place, calculating test-retest reliability, internal consistency, or interrater reliability are straightforward and essentially only require calculating correlations between test scores. Test scores can easily be corrected for reliability-related attenuation, and also item-level characteristics such as difficulty and discrimination can easily be calculated as arithmetic means of responses and correlations between item responses and test scores, respectively. Overall, CTT provides a simple theoretical framework that can prove useful in practice.

However, as mentioned, CTT comes with several downsides. First, it is primarily a theory of test errors, while it only has a very simple notion of how the 'true score' is related to the examinee's actual state/trait and to the single items that the test is made up of. The assumptions of uncorrelated errors are restrictive, since it does not appear plausible that tests are equally accurate irrespective of whether the examinee has a high or low value on the quantity of interest. Finally, CTT provides no method for considering test and person characteristics independently from each other (Van der Linden and Hambleton (1997)).

10.4 Strengths and Limitations of IRT

IRT has been designed to overcome some of the limitations of CTT. It provides a theory of individual *item responses*, rather than starting with the total test score. Moreover, IRT does not have a notion of test errors. Instead, responses to single items of a test are modeled as a *probabilistic* function of the difficulty of the item and the ability (latent trait/state) of the examinee, the so-called *item response function*. If a person with low ability solves an item with high difficulty, this is not thought of as an *error* within IRT, but rather as a less likely, yet not impossible, realization of a random variable. Both item difficulties and person abilities are unknown a priori and are estimated from data using a probabilistic (often logistic) model, as described in more detail in the practical section.

IRT is, overall, a more flexible framework to characterize tests than CTT. For instance, it does not assume tests to be equally accurate for low- and high-ability persons. Rather, the logistic function comes with a natural saturation property that can model items to have different sensitivity to different ability levels. Graphically, this becomes apparent when looking at ICCs (see above) (Crocker and Algina (2008)).

The key difference between CTT and IRT is that IRT explicitly formulates a testable latent variable model that explains the relationship between person abilities and item properties, while CTT simply assumes a linear link between item responses and person abilities (test scores), without providing a proper means to test this assumption. Whithin IRT, the question of whether items can be summed up, i.e., whether items are independent of each other with respect to person ability, and whether the test is unidimensional, can be explicitly tested. Moreover, the item coefficients of an IRT model are, at least in theory, sample-independent, in contrast to CTT.

The notions of reliability, item difficulty, and item discrimination are also defined within the latent variable model in IRT, rather than being calculated from the sample data. Reliability is usually tested in terms of unidimensionality of a factor model, i.e., whether item responses can well be explained by a single underlying latent factor. Item difficulty and discrimination, on the other hand, can be directly read off from the latent variable model. Item difficulty describes the item threshold, i.e., the level of person ability necessary to have a 50% chance of solving an item. Item discrimination, on the other hand, is fixed to be equal across all items in the simplest case (the Rasch model) and is a free parameter only in higher-parameterized models such as the Birnbaum model. It can be thought of as an item's sensitivity to changes in person ability. Graphically, it appears as the slope of the ICC in its steepest point.

The greater flexibility of IRT models comes, on the other hand, with greater responsibility on the part of the human user. IRT models are undoubtedly more difficult to understand and operate, and require more choices (e.g., should a Rasch model with one free parameter, a Birnbaum model with two free parameters, or an even more complex model be used?). Moreover, IRT models come with a set of assumptions about the item response data in order to describe the test appropriately. Assumptions of IRT are, in particular, local independence of items, i.e., mutual conditional independence of item responses given the person ability score, and unidimensionality of the test, and an item response structure that can be described by the item response function, in particular, a monotonic relationship between person ability and probability to solve the item (Crocker and Algina (2008)).

10.5 Conclusion and Application

I have briefly outlined key tenets, strengths and limitations of CTT, and how IRT attempts to overcome those limitations. Simultaneously a strength and a limitation of CTT is its simplicity, as it does not rely on any latent variable modeling and is rather easy to understand and apply. IRT, on the other hand, is the more complex theory, but it allows more flexibility and is, when properly applied, more informative about the test and the examinees.

Methods from both theories, and combinations between them, have been successfully applied to construct and validate tests. It should be noted that both theories are, in and of themselves, insufficient for test construction and validation, since they primarily focus on reliability, and, to a lesser extent, objectivity issues. The question of test validity, on the other hand, is arguably the most important quality of a test, but there is no purely mathematical way to describe or ascertain validity - rather, it must be derived from, and validated by, theory, observation, and sometimes intuition.

11 Analysis code

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

```
library(ggplot2)
library(ggthemes)
library(reshape2)
library(readxl)
library(VIM)
```

```
## Loading required package: colorspace

## Loading required package: grid

## VIM is ready to use.

## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues

##
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':
##
##     sleep
```

```
library(mice)
```

```
##
## Attaching package: 'mice'

## The following object is masked from 'package:stats':
##
##     filter

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:reshape2':
##
##     smiths
```

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(ggcorrplot)
library(eRm)
```

```
##
## Attaching package: 'eRm'

## The following object is masked from 'package:psych':
##
##     sim.rasch
```

```
library(ltm)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: msm

## Loading required package: polycor

##
## Attaching package: 'polycor'

## The following object is masked from 'package:psych':
##
##     polyserial

##
## Attaching package: 'ltm'

## The following object is masked from 'package:psych':
##
##     factor.scores
```

```
library(patchwork)
```

```
##
## Attaching package: 'patchwork'
```

```
## The following object is masked from 'package:MASS':
##
##     area
```

```
library(difR)
library(semPlot)
library(lavaan)
```

```
## This is lavaan 0.6-11
## lavaan is FREE software! Please report any bugs.

##
## Attaching package: 'lavaan'

## The following object is masked from 'package:psych':
##
##     cor2cov
```

```
library(semTools)
```

```
##

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

## This is semTools 0.5-6

## All users of R (or SEM) are invited to submit functions or ideas for functions.

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

##
## Attaching package: 'semTools'

## The following objects are masked from 'package:psych':
##
##     reliability, skew
```

```
#####
#part 1: data preparation, descriptive analyses
#####
{

df = read_xlsx("SCS_data.xlsx")
#df = read_xlsx("data.csv")
#if data have been re-downloaded from
#openpsychometrics, uncomment the above line

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))
```

```r
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))
```

```r
#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)]
write.csv(descriptives, "descriptives.csv")
#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
```

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

  #tetrachoric correlations
  tetra_cor = tetrachoric(dich[,SCS_vars])
  ggcorrplot(tetra_cor$rho, type = "lower", lab = TRUE)+theme_clean()
  ggsave("tetrachoric_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("item easiness\n(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)
  write.csv(dich.stats,"dich_stats.csv")
}
```

```
#####
#part 3: estimate and analyze Rasch model
#####

#model fitting
{
  #approach 1: eRm
  #prepare data for eRm estimation
  #(just item data in wide format)
  rasch_model_eRm = RM(dich[,SCS_vars])

  #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)

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") +
```

```r
    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

iccplot_lavaan=ICC_plot(difficulties_lavaan)+ggtitle("lavaan")



difficulties = rbind( data.frame(model="eRm",
                          item=factor(SCS_vars),
                          difficulty=as.numeric(difficulties_eRm)),
              data.frame(model="ltm",
                         item=factor(SCS_vars),
                         difficulty=as.numeric(difficulties_ltm)),
              data.frame(model="lavaan",
                         item=factor(SCS_vars),
                         difficulty=as.numeric(difficulties_lavaan)),
              data.frame(model="CTT",
                         item=factor(SCS_vars),
                         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=SCS_vars,limits=SCS_vars)


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

#arrange plots vertically and save
iccplot_eRm|iccplot_ltm|iccplot_lavaan
```

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


#compare fits
  #select second line of output (corresponding to marginal MLE)
eRm_fit = IC(person.parameter(rasch_model_eRm))[[1]][2,]

ltm_fit = c()
ltm_fit['value'] = smr_ltm$logLik
ltm_fit['npar'] = 10
ltm_fit['AIC'] = smr_ltm$AIC
ltm_fit['BIC'] = smr_ltm$BIC
ltm_fit['cAIC'] = NA

rasch_model_fits = rbind(eRm_fit,ltm_fit)
rownames(rasch_model_fits) = c("eRm","ltm")
colnames(rasch_model_fits)[1] = "loglik"
write.csv(rasch_model_fits,"rasch_model_fits.csv")



#calculate loss per item

predict_responses = function(item_dffc,person_params,item_discr=1){
  item_dffc = as.numeric(item_dffc)
  if (length(item_discr)==1) item_discr = rep(item_discr,length(item_dffc))
  person_params = as.numeric(person_params)
  preds = matrix(nrow=length(person_params),ncol=length(item_dffc))
  for (p in 1:length(person_params)){
    for(i in 1:length(item_dffc)){
      preds[p,i] = logistic(x=person_params[p],d = item_dffc[i], a = item_discr[i])
  }}
  return(preds)
}


#extract latent person abilities
person_params_eRm = person.parameter(rasch_model_eRm)$theta.table[,"Person Parameter"]
person_params_ltm=factor.scores(rasch_model_ltm,dich[,SCS_vars])[[1]][,"z1"]
person_params_lavaan = as.numeric(predict(rasch_model_lavaan))

#make predictions for individual persons and items
preds_ltm = predict_responses(difficulties_ltm,person_params_ltm)>.5
preds_eRm = predict_responses(difficulties_eRm,person_params_eRm)>.5
preds_lavaan = predict_responses(difficulties_lavaan,person_params_lavaan)>.5
```

```r
#calculate and plot mean 0-1-loss per item
itemloss_eRm = colMeans(dich[,SCS_vars]!=preds_eRm)
itemloss_ltm = colMeans(dich[,SCS_vars]!=preds_ltm)
itemloss_lavaan = colMeans(dich[,SCS_vars]!=preds_lavaan)
itemloss = rbind(data.frame(model="eRm",item=SCS_vars,loss=itemloss_eRm),
                 data.frame(model="ltm",item=SCS_vars,loss=itemloss_ltm),
                 data.frame(model="lavaan",item=SCS_vars,loss=itemloss_lavaan))

ggplot(itemloss,aes(x=item,y=loss,
                    color=model,group=model)) +
  geom_point() + geom_line() + theme_clean() + ggtitle("0-1-loss\nmodel comparison")+
  scale_x_discrete(breaks=SCS_vars,limits=SCS_vars)

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

}
```

```
## 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                              2426.372    1524.464
##   Degrees of freedom                                45          45
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  1.610
##   Shift parameter                                           17.052
##        simple second-order correction (WLSMV)
##
## Model Test Baseline Model:
##
##   Test statistic                             39114.763   24317.567
##   Degrees of freedom                                45          45
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.610
##
## 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                                           6.642       6.642
##
## Parameter Estimates:
##
##   Standard errors                           Robust.sem
##   Information                                 Expected
##   Information saturated (h1) model        Unstructured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   SCS =~
##     Q1                1.000
##     Q2                1.000
##     Q3                1.000
##     Q4                1.000
##     Q5                1.000
##     Q6                1.000
##     Q7                1.000
##     Q8                1.000
##     Q9                1.000
##     Q10               1.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     SCS               0.000
##    .Q1                0.000
##    .Q2                0.000
##    .Q3                0.000
##    .Q4                0.000
##    .Q5                0.000
```

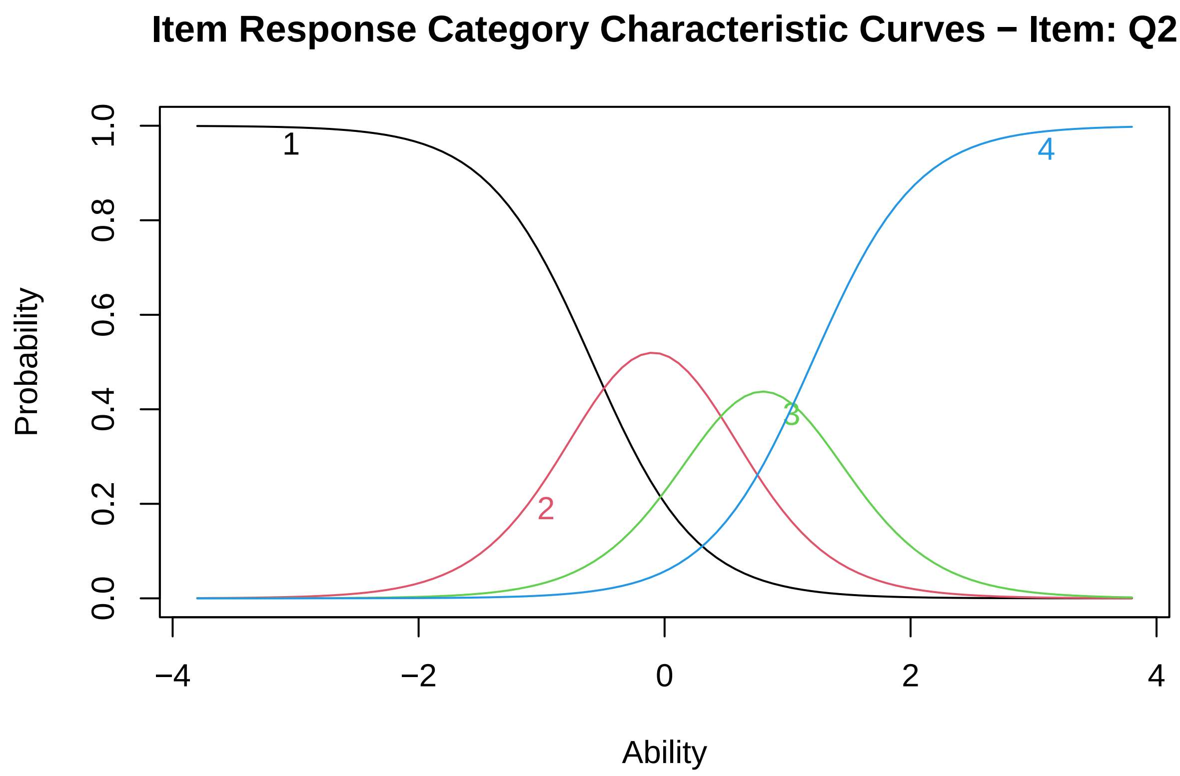
```
##    .Q6                0.000
##    .Q7                0.000
##    .Q8                0.000
##    .Q9                0.000
##    .Q10               0.000
##
## Thresholds:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     Q1|t1             0.339    0.031   10.980    0.000
##     Q2|t1             0.437    0.031   14.069    0.000
##     Q3|t1             0.459    0.031   14.754    0.000
##     Q4|t1             0.862    0.033   26.357    0.000
##     Q5|t1             0.425    0.031   13.692    0.000
##     Q6|t1            -0.819    0.032  -25.219    0.000
##     Q7|t1             0.438    0.031   14.103    0.000
##     Q8|t1             0.310    0.031   10.052    0.000
##     Q9|t1             0.022    0.031    0.724    0.469
##     Q10|t1           -0.029    0.031   -0.965    0.335
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     SCS               1.000
##    .Q1                1.000
##    .Q2                1.000
##    .Q3                1.000
##    .Q4                1.000
##    .Q5                1.000
##    .Q6                1.000
##    .Q7                1.000
##    .Q8                1.000
##    .Q9                1.000
##    .Q10               1.000
##
## Scales y*:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     Q1                0.707
##     Q2                0.707
##     Q3                0.707
##     Q4                0.707
##     Q5                0.707
##     Q6                0.707
##     Q7                0.707
##     Q8                0.707
##     Q9                0.707
##     Q10               0.707
```

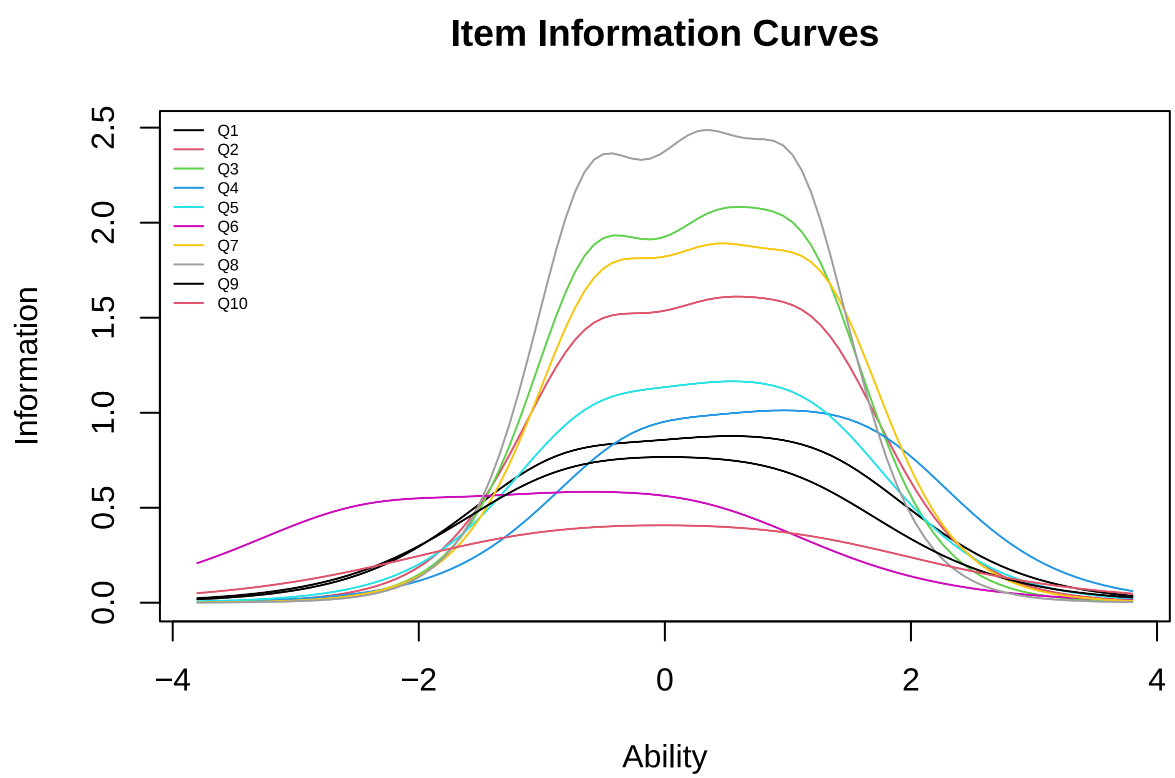
```
#DIF
{
```

```
  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=SCS_vars,
                                                        limits=SCS_vars)

  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(SCS_vars),
                                          difficulty=as.numeric(difficulties_ltm)),
                               data.frame(model="Birnbaum (2-PL)",
                                          item=factor(SCS_vars),
                                          difficulty=as.numeric(difficulties_2PL)),
                               data.frame(model="CTT",
                                          item=factor(SCS_vars),
```

```
                                   difficulty=1-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")+
  scale_x_discrete(breaks=SCS_vars,limits=SCS_vars)
ggsave("difficulties_plot_2PL.pdf",width = 6,height = 4)


#calculate item-wise infit and outfit

get_outfit = function(ltm_model){
  X=ltm_model$X
  personscores = factor.scores(ltm_model,X)[[1]][,"z1"]
  dffc = coef(ltm_model)[,"Dffclt"]
  discr = coef(ltm_model)[,"Dscrmn"]
  expected = predict_responses(dffc, personscores, discr)
  var_X = expected * (1-expected)
  Z_ij = (X-expected)/sqrt(var_X)
  chisq = colSums(Z_ij**2)
  #divide chisq by n
  return(chisq/nrow(ltm_model$X))
    }

get_infit = function(ltm_model){
  X=ltm_model$X
  personscores = factor.scores(ltm_model,X)[[1]][,"z1"]
  dffc = coef(ltm_model)[,"Dffclt"]
  discr = coef(ltm_model)[,"Dscrmn"]
  expected = predict_responses(dffc, personscores, discr)
  var_X = expected * (1-expected)
  Z_ij = (X-expected)/sqrt(var_X)
  infit = c()
  for (i in 1:length(dffc)){
    infit[i] = sum((var_X[,i] * (Z_ij[,i]**2))/sum(var_X[,i]))}
  return(infit)
}

outfit_rasch = get_outfit(rasch_model_ltm)
outfit_2PL = get_outfit(twoPL_model)
infit_rasch = get_infit(rasch_model_ltm)
infit_2PL = get_infit(twoPL_model)

inoutfit = rbind(data.frame(model="Rasch",fit="outfit",
                            item=names(outfit_rasch), value=outfit_rasch),
                 data.frame(model="2-PL",fit="outfit",
                            item=names(outfit_2PL), value=outfit_2PL),
                 data.frame(model="Rasch",fit="infit",
```

```
                                  item=names(outfit_rasch),value=infit_rasch),
                       data.frame(model="2-PL",fit="infit",
                                  item=names(outfit_2PL),value=infit_2PL))

  ggplot(inoutfit,aes(x=item,y=value,
                                     color=model,group=model)) + facet_wrap(~fit)+
    geom_point() + geom_line() + theme_clean() + ggtitle("model comparison")+
    scale_x_discrete(breaks=SCS_vars,limits=SCS_vars)
  ggsave("inoutfit_plot_2PL.pdf",width = 8,height = 3)

}

#polytomous IRT model
{
  grm_constrained = grm(df_clean[,SCS_vars],constrained=T)
  grm_unconstrained = grm(df_clean[,SCS_vars],constrained=F)
  anova(grm_constrained, grm_unconstrained)

  #have to save this plot by hand, automatic saving does
  #not work for some reason
  plot(grm_unconstrained, item=2, ask=F)

  plot(grm_unconstrained, type="IIC", ask=F,legend=T,cx='topleft',cex=.5)

}
```


Item Response Category Characteristic Curves − Item: Q2
Probability
Ability
1
2
3
4
0.0
0.2
0.4
0.6
0.8
1.0
−4
−2
0
2
4


Item Information Curves
Information
Ability
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
Q9
Q10
0.0
0.5
1.0
1.5
2.0
2.5
−4
−2
0
2
4

```r
#####
#part 4: factorial structure of the data
#####

#factor analyses
{

  covdat = cov(df_clean[,SCS_vars])
  N=nrow(df_clean)


  #unidimensional model
  unidimensional_model <- '
  xi1 =~ Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9+Q10

'

  unidimensional_cfa <- cfa(unidimensional_model,
                      sample.cov=covdat,
                      sample.nobs=N,
                      std.lv=T)


  #correlated traits
  correlated_traits_model <- '
  xi1 =~ Q1+Q2+Q3+Q4+Q10
  xi2 =~ Q5+Q6+Q7+Q8+Q9
  xi1 ~~ xi2
'

  correlated_traits_cfa <- cfa(correlated_traits_model,
                               sample.cov=covdat,
                               sample.nobs=N,
                               std.lv=T)


  #bifactor model (general factor and two item-specific factors)
  bifactor_model <- '
  G =~ Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9+Q10
  xi1 =~ Q1+Q2+Q3+Q4+Q10
  xi2 =~ Q5+Q6+Q7+Q8+Q9
  G ~~ 0*xi1
  G ~~ 0*xi2
'

  bifactor_cfa <- cfa(bifactor_model,
              sample.cov=covdat,
```

```
            sample.nobs=N,
            std.lv=T)




#hierarchical model

hierarchical_model <- '
xi1 =~ Q1+Q2+Q3+Q4+Q10
xi2 =~ Q5+Q6+Q7+Q8+Q9
G =~ xi1+xi2
'

hierarchical_cfa <- cfa(hierarchical_model,
                    sample.cov=covdat,
                    sample.nobs=N,
                    std.lv=T)

#
smr_hierarchical = summary(hierarchical_cfa, fit=T)$FIT
smr_bifactor = summary(bifactor_cfa, fit=T)$FIT
smr_correlated_traits = summary(correlated_traits_cfa, fit=T)$FIT
smr_unidimensional = summary(unidimensional_cfa, fit=T)$FIT

cfaaov_df = data.frame(model=c("hierarchical","bifactor","correlated traits",
                               "unidimensional"),
                       Df = c(smr_hierarchical["df"],
                              smr_bifactor["df"],
                              smr_correlated_traits["df"],
                              smr_unidimensional["df"]),
                       AIC = c(smr_hierarchical["aic"],
                               smr_bifactor["aic"],
                               smr_correlated_traits["aic"],
                               smr_unidimensional["aic"]),
                       BIC = c(smr_hierarchical["bic"],
                               smr_bifactor["bic"],
                               smr_correlated_traits["bic"],
                               smr_unidimensional["bic"]))

write.csv(cfaaov_df,"cfaaov_df.csv")



pdf("semplot_bifactor.pdf", width = 8,height = 4)
semPaths(bifactor_cfa, "std")
```

```
dev.off()




#fit alternative bifactor model (item Q10 belongs to subscale 2)
bifactor_model_2 <- '
G =~ Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9+Q10
xi1 =~ Q1+Q2+Q3+Q4
xi2 =~ Q5+Q6+Q7+Q8+Q9+Q10
G ~~ 0*xi1
G ~~ 0*xi2

'

bifactor_cfa_2 <- cfa(bifactor_model_2,
                    sample.cov=covdat,
                    sample.nobs=N,
                    std.lv=T)

#fit alternative bifactor model (item Q10 is its own subscale)
#-> does not converge
bifactor_model_3 <- '
G =~ Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9+Q10
xi1 =~ Q1+Q2+Q3+Q4
xi2 =~ Q5+Q6+Q7+Q8+Q9
xi3 =~ Q10
G ~~ 0*xi1
G ~~ 0*xi2
G ~~ 0*xi3

'

bifactor_cfa_3 <- cfa(bifactor_model_3,
                      sample.cov=covdat,
                      sample.nobs=N,
                      std.lv=T)

bifactor_model_4 <- '
G =~ Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9+Q10
xi1 =~ Q1+Q2+Q3+Q4
xi2 =~ Q5+Q6+Q7+Q8+Q9
xi3 =~ Q10+Q6+Q1
G ~~ 0*xi1
G ~~ 0*xi2
G ~~ 0*xi3
```

```r
'

  bifactor_cfa_4<- cfa(bifactor_model_4,
                       sample.cov=covdat,
                       sample.nobs=N,
                       std.lv=T)


  pdf("semplot_bifactor_automatic.pdf", width = 8,height = 4)
  semPaths(bifactor_cfa_4, "std")
  dev.off()


  smr_bif1=summary(bifactor_cfa,fit=T)$FIT
  smr_bif2=summary(bifactor_cfa_2,fit=T)$FIT
  smr_bif3=summary(bifactor_cfa_3,fit=T)$FIT
  smr_bif4=summary(bifactor_cfa_4,fit=T)$FIT

  bif_comparison=rbind(smr_bif1,smr_bif2,smr_bif3,smr_bif4)[,c("npar","aic",
                                                              "bic","cfi",
                                                              "tli","srmr",
                                                              "rmsea")]
  rownames(bif_comparison) = c("Bifactor (original)",
                               "Bifactor (Alternative 1)",
                               "Bifactor (Alternative 2)",
                               "Bifactor (Automatized)")
  write.csv(bif_comparison, "bifactor_comparison.csv")

  }
```

```
## Warning in lav_model_vcov(lavmodel = lavmodel, lavsamplestats = lavsamplestats, : lavaan WAR
##     Could not compute standard errors! The information matrix could
##     not be inverted. This may be a symptom that the model is not
##     identified.

## lavaan 0.6-11 ended normally after 33 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        22
##
##   Number of observations                          3368
##
## Model Test User Model:
##
##   Test statistic                               882.107
##   Degrees of freedom                                33
##   P-value (Chi-square)                           0.000
```

```
##
## Model Test Baseline Model:
##
##   Test statistic                             16314.385
##   Degrees of freedom                                45
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.948
##   Tucker-Lewis Index (TLI)                       0.929
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -42761.201
##   Loglikelihood unrestricted model (H1)     -42320.148
##
##   Akaike (AIC)                               85566.402
##   Bayesian (BIC)                             85701.088
##   Sample-size adjusted Bayesian (BIC)        85631.184
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.087
##   90 Percent confidence interval - lower         0.082
##   90 Percent confidence interval - upper         0.092
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.041
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   xi1 =~
##     Q1                0.316       NA
##     Q2                0.364       NA
##     Q3                0.380       NA
##     Q4                0.307       NA
##     Q10               0.248       NA
##   xi2 =~
##     Q5                0.335       NA
```

```
##     Q6               0.223       NA
##     Q7               0.366       NA
##     Q8               0.395       NA
##     Q9               0.312       NA
##   G =~
##     xi1              2.156       NA
##     xi2              2.159       NA
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)
##    .Q1               0.613       NA
##    .Q2               0.400       NA
##    .Q3               0.347       NA
##    .Q4               0.542       NA
##    .Q10              1.060       NA
##    .Q5               0.610       NA
##    .Q6               0.651       NA
##    .Q7               0.379       NA
##    .Q8               0.298       NA
##    .Q9               0.783       NA
##    .xi1              1.000
##    .xi2              1.000
##     G                1.000
##
## lavaan 0.6-11 ended normally after 36 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        31
##
##   Number of observations                          3368
##
## Model Test User Model:
##
##   Test statistic                               460.194
##   Degrees of freedom                                24
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                             16314.385
##   Degrees of freedom                                45
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.973
##   Tucker-Lewis Index (TLI)                       0.950
```

```
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -42550.245
##   Loglikelihood unrestricted model (H1)     -42320.148
##
##   Akaike (AIC)                               85162.490
##   Bayesian (BIC)                             85352.274
##   Sample-size adjusted Bayesian (BIC)        85253.773
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.073
##   90 Percent confidence interval - lower         0.068
##   90 Percent confidence interval - upper         0.079
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.028
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   G =~
##     Q1                0.555    0.037   15.043    0.000
##     Q2                0.427    0.044    9.700    0.000
##     Q3                0.755    0.032   23.399    0.000
##     Q4                0.650    0.026   25.352    0.000
##     Q5                0.706    0.024   29.145    0.000
##     Q6                0.542    0.020   27.799    0.000
##     Q7                0.606    0.031   19.389    0.000
##     Q8                0.628    0.035   17.991    0.000
##     Q9                0.512    0.031   16.608    0.000
##     Q10               0.629    0.023   27.218    0.000
##   xi1 =~
##     Q1                0.500    0.043   11.616    0.000
##     Q2                0.969    0.037   26.171    0.000
##     Q3                0.474    0.046   10.233    0.000
##     Q4                0.333    0.038    8.741    0.000
##     Q10               0.174    0.034    5.130    0.000
##   xi2 =~
##     Q5                0.402    0.033   12.229    0.000
```

```
##     Q6                 0.186    0.027    6.805    0.000
##     Q7                 0.625    0.030   21.014    0.000
##     Q8                 0.727    0.030   24.391    0.000
##     Q9                 0.541    0.030   18.240    0.000
##
## Covariances:
##                     Estimate  Std.Err  z-value  P(>|z|)
##   G ~~
##     xi1                0.000
##     xi2                0.000
##   xi1 ~~
##     xi2                0.587    0.040   14.619    0.000
##
## Variances:
##                     Estimate  Std.Err  z-value  P(>|z|)
##    .Q1                 0.618    0.017   35.498    0.000
##    .Q2                 0.025    0.070    0.361    0.718
##    .Q3                 0.368    0.013   28.020    0.000
##    .Q4                 0.540    0.016   33.830    0.000
##    .Q5                 0.585    0.017   34.252    0.000
##    .Q6                 0.604    0.017   35.041    0.000
##    .Q7                 0.380    0.013   29.779    0.000
##    .Q8                 0.257    0.013   19.267    0.000
##    .Q9                 0.780    0.021   37.281    0.000
##    .Q10                0.981    0.027   36.001    0.000
##     G                  1.000
##     xi1                1.000
##     xi2                1.000
##
## lavaan 0.6-11 ended normally after 20 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        21
##
##   Number of observations                          3368
##
## Model Test User Model:
##
##   Test statistic                               882.107
##   Degrees of freedom                                34
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                             16314.385
##   Degrees of freedom                                45
##   P-value                                        0.000
```

```
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.948
##   Tucker-Lewis Index (TLI)                       0.931
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -42761.201
##   Loglikelihood unrestricted model (H1)     -42320.148
##
##   Akaike (AIC)                               85564.402
##   Bayesian (BIC)                             85692.966
##   Sample-size adjusted Bayesian (BIC)        85626.239
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.086
##   90 Percent confidence interval - lower         0.081
##   90 Percent confidence interval - upper         0.091
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.041
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   xi1 =~
##     Q1                0.751    0.017   43.800    0.000
##     Q2                0.864    0.016   54.288    0.000
##     Q3                0.903    0.016   57.356    0.000
##     Q4                0.729    0.016   44.778    0.000
##     Q10               0.589    0.020   29.150    0.000
##   xi2 =~
##     Q5                0.796    0.017   45.830    0.000
##     Q6                0.531    0.016   32.959    0.000
##     Q7                0.871    0.016   55.485    0.000
##     Q8                0.939    0.016   60.481    0.000
##     Q9                0.743    0.019   39.957    0.000
##
## Covariances:
```

```
##                    Estimate  Std.Err  z-value  P(>|z|)
##   xi1 ~~
##     xi2               0.823    0.008   97.055    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .Q1                0.613    0.017   36.174    0.000
##    .Q2                0.400    0.013   30.985    0.000
##    .Q3                0.347    0.012   28.420    0.000
##    .Q4                0.542    0.015   35.844    0.000
##    .Q10               1.060    0.027   39.326    0.000
##    .Q5                0.610    0.017   35.930    0.000
##    .Q6                0.651    0.017   38.936    0.000
##    .Q7                0.379    0.012   30.971    0.000
##    .Q8                0.298    0.011   26.302    0.000
##    .Q9                0.783    0.021   37.597    0.000
##     xi1               1.000
##     xi2               1.000
##
## lavaan 0.6-11 ended normally after 16 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        20
##
##   Number of observations                          3368
##
## Model Test User Model:
##
##   Test statistic                              1863.230
##   Degrees of freedom                                35
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                             16314.385
##   Degrees of freedom                                45
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.888
##   Tucker-Lewis Index (TLI)                       0.856
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -43251.763
##   Loglikelihood unrestricted model (H1)     -42320.148
```

```
##
##   Akaike (AIC)                                86543.526
##   Bayesian (BIC)                              86665.968
##   Sample-size adjusted Bayesian (BIC)         86602.418
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                           0.125
##   90 Percent confidence interval - lower          0.120
##   90 Percent confidence interval - upper          0.129
##   P-value RMSEA <= 0.05                           0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                            0.053
##
## Parameter Estimates:
##
##   Standard errors                              Standard
##   Information                                  Expected
##   Information saturated (h1) model           Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   xi1 =~
##     Q1                0.705    0.017   40.881    0.000
##     Q2                0.807    0.016   49.865    0.000
##     Q3                0.847    0.016   52.974    0.000
##     Q4                0.694    0.016   42.474    0.000
##     Q5                0.779    0.017   44.852    0.000
##     Q6                0.530    0.016   33.110    0.000
##     Q7                0.821    0.016   51.418    0.000
##     Q8                0.877    0.016   55.177    0.000
##     Q9                0.715    0.019   38.366    0.000
##     Q10               0.599    0.020   30.060    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .Q1                0.679    0.018   38.055    0.000
##    .Q2                0.496    0.014   35.669    0.000
##    .Q3                0.445    0.013   34.435    0.000
##    .Q4                0.592    0.016   37.725    0.000
##    .Q5                0.638    0.017   37.168    0.000
##    .Q6                0.652    0.017   39.293    0.000
##    .Q7                0.464    0.013   35.089    0.000
##    .Q8                0.410    0.012   33.361    0.000
##    .Q9                0.824    0.021   38.517    0.000
##    .Q10               1.048    0.026   39.650    0.000
```

```
##     xi1              1.000
## Warning in lav_object_post_check(object): lavaan WARNING: some estimated ov
## variances are negative
## Warning in lav_model_vcov(lavmodel = lavmodel, lavsamplestats = lavsamplestats, : lavaan WAR
##     Could not compute standard errors! The information matrix could
##     not be inverted. This may be a symptom that the model is not
##     identified.
## Warning in lav_object_post_check(object): lavaan WARNING: some estimated ov
## variances are negative
## lavaan 0.6-11 ended normally after 36 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        31
##
##   Number of observations                          3368
##
## Model Test User Model:
##
##   Test statistic                               460.194
##   Degrees of freedom                                24
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                             16314.385
##   Degrees of freedom                                45
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.973
##   Tucker-Lewis Index (TLI)                       0.950
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -42550.245
##   Loglikelihood unrestricted model (H1)     -42320.148
##
##   Akaike (AIC)                               85162.490
##   Bayesian (BIC)                             85352.274
##   Sample-size adjusted Bayesian (BIC)        85253.773
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.073
```

```
##   90 Percent confidence interval - lower         0.068
##   90 Percent confidence interval - upper         0.079
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.028
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   G =~
##     Q1                0.555    0.037   15.043    0.000
##     Q2                0.427    0.044    9.700    0.000
##     Q3                0.755    0.032   23.399    0.000
##     Q4                0.650    0.026   25.352    0.000
##     Q5                0.706    0.024   29.145    0.000
##     Q6                0.542    0.020   27.799    0.000
##     Q7                0.606    0.031   19.389    0.000
##     Q8                0.628    0.035   17.991    0.000
##     Q9                0.512    0.031   16.608    0.000
##     Q10               0.629    0.023   27.218    0.000
##   xi1 =~
##     Q1                0.500    0.043   11.616    0.000
##     Q2                0.969    0.037   26.171    0.000
##     Q3                0.474    0.046   10.233    0.000
##     Q4                0.333    0.038    8.741    0.000
##     Q10               0.174    0.034    5.130    0.000
##   xi2 =~
##     Q5                0.402    0.033   12.229    0.000
##     Q6                0.186    0.027    6.805    0.000
##     Q7                0.625    0.030   21.014    0.000
##     Q8                0.727    0.030   24.391    0.000
##     Q9                0.541    0.030   18.240    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   G ~~
##     xi1               0.000
##     xi2               0.000
##   xi1 ~~
##     xi2               0.587    0.040   14.619    0.000
##
```

```
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .Q1                0.618    0.017   35.498    0.000
##    .Q2                0.025    0.070    0.361    0.718
##    .Q3                0.368    0.013   28.020    0.000
##    .Q4                0.540    0.016   33.830    0.000
##    .Q5                0.585    0.017   34.252    0.000
##    .Q6                0.604    0.017   35.041    0.000
##    .Q7                0.380    0.013   29.779    0.000
##    .Q8                0.257    0.013   19.267    0.000
##    .Q9                0.780    0.021   37.281    0.000
##    .Q10               0.981    0.027   36.001    0.000
##     G                 1.000
##     xi1               1.000
##     xi2               1.000
##
## lavaan 0.6-11 ended normally after 4502 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        31
##
##   Number of observations                          3368
##
## Model Test User Model:
##
##   Test statistic                               473.809
##   Degrees of freedom                                24
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                             16314.385
##   Degrees of freedom                                45
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.972
##   Tucker-Lewis Index (TLI)                       0.948
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -42557.053
##   Loglikelihood unrestricted model (H1)     -42320.148
##
##   Akaike (AIC)                               85176.105
##   Bayesian (BIC)                             85365.889
```

```
##   Sample-size adjusted Bayesian (BIC)       85267.388
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                         0.075
##   90 Percent confidence interval - lower        0.069
##   90 Percent confidence interval - upper        0.081
##   P-value RMSEA <= 0.05                         0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                          0.031
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   G =~
##     Q1                0.709    0.032   22.311    0.000
##     Q2                0.600    0.031   19.140    0.000
##     Q3                0.906    0.023   39.644    0.000
##     Q4                0.748    0.018   42.238    0.000
##     Q5                0.717    0.019   37.769    0.000
##     Q6                0.513    0.017   31.025    0.000
##     Q7                0.702    0.022   31.625    0.000
##     Q8                0.745    0.024   30.419    0.000
##     Q9                0.585    0.024   24.446    0.000
##     Q10               0.598    0.021   28.099    0.000
##   xi1 =~
##     Q1                0.031    0.779    0.040    0.968
##     Q2                9.871  244.525    0.040    0.968
##     Q3                0.024    0.593    0.040    0.968
##     Q4                0.015    0.367    0.040    0.968
##   xi2 =~
##     Q5                0.340    0.023   14.496    0.000
##     Q6                0.169    0.020    8.570    0.000
##     Q7                0.514    0.025   20.707    0.000
##     Q8                0.599    0.027   21.887    0.000
##     Q9                0.469    0.026   17.856    0.000
##     Q10               0.091    0.024    3.753    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   G ~~
```

```
##     xi1               0.000
##     xi2               0.000
##   xi1 ~~
##     xi2               0.038    0.933    0.040    0.968
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .Q1                0.672    0.022   30.629    0.000
##    .Q2              -96.642 4827.250   -0.020    0.984
##    .Q3                0.341    0.014   23.613    0.000
##    .Q4                0.514    0.015   33.219    0.000
##    .Q5                0.614    0.017   36.758    0.000
##    .Q6                0.641    0.016   38.902    0.000
##    .Q7                0.382    0.013   29.795    0.000
##    .Q8                0.266    0.013   20.598    0.000
##    .Q9                0.773    0.021   36.877    0.000
##    .Q10               1.041    0.027   37.852    0.000
##     G                 1.000
##     xi1               1.000
##     xi2               1.000
##
## lavaan 0.6-11 ended normally after 44 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        33
##
##   Number of observations                          3368
##
## Model Test User Model:
##
##   Test statistic                               455.655
##   Degrees of freedom                                22
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                             16314.385
##   Degrees of freedom                                45
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.973
##   Tucker-Lewis Index (TLI)                       0.945
##
## Loglikelihood and Information Criteria:
##
```

```
##   Loglikelihood user model (H0)             -42547.976
##   Loglikelihood unrestricted model (H1)     -42320.148
##
##   Akaike (AIC)                               85161.951
##   Bayesian (BIC)                             85363.979
##   Sample-size adjusted Bayesian (BIC)        85259.123
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.077
##   90 Percent confidence interval - lower         0.070
##   90 Percent confidence interval - upper         0.083
##   P-value RMSEA <= 0.05                          0.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.029
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   G =~
##     Q1                0.565       NA
##     Q2                0.414       NA
##     Q3                0.785       NA
##     Q4                0.675       NA
##     Q5                0.687       NA
##     Q6                0.517       NA
##     Q7                0.605       NA
##     Q8                0.629       NA
##     Q9                0.501       NA
##     Q10               0.599       NA
##   xi1 =~
##     Q1                0.477       NA
##     Q2                1.031       NA
##     Q3                0.443       NA
##     Q4                0.308       NA
##   xi2 =~
##     Q5                0.423       NA
##     Q6                0.216       NA
##     Q7                0.627       NA
##     Q8                0.720       NA
##     Q9                0.556       NA
```

```
##   xi3 =~
##     Q10               0.701       NA
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   G ~~
##     xi1               0.000
##     xi2               0.000
##     xi3               0.000
##   xi1 ~~
##     xi2               0.565       NA
##     xi3               0.259       NA
##   xi2 ~~
##     xi3               0.225       NA
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .Q1                0.630       NA
##    .Q2               -0.088       NA
##    .Q3                0.351       NA
##    .Q4                0.523       NA
##    .Q5                0.593       NA
##    .Q6                0.619       NA
##    .Q7                0.380       NA
##    .Q8                0.265       NA
##    .Q9                0.776       NA
##    .Q10               0.556       NA
##     G                 1.000
##     xi1               1.000
##     xi2               1.000
##     xi3               1.000
##
## lavaan 0.6-11 ended normally after 50 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        35
##
##   Number of observations                          3368
##
## Model Test User Model:
##
##   Test statistic                               206.177
##   Degrees of freedom                                20
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
```

```
##   Test statistic                             16314.385
##   Degrees of freedom                                45
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.989
##   Tucker-Lewis Index (TLI)                       0.974
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -42423.237
##   Loglikelihood unrestricted model (H1)     -42320.148
##
##   Akaike (AIC)                               84916.473
##   Bayesian (BIC)                             85130.746
##   Sample-size adjusted Bayesian (BIC)        85019.535
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.053
##   90 Percent confidence interval - lower         0.046
##   90 Percent confidence interval - upper         0.059
##   P-value RMSEA <= 0.05                          0.246
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.019
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   G =~
##     Q1                0.323    0.078    4.142    0.000
##     Q2                0.200    0.090    2.238    0.025
##     Q3                0.679    0.067   10.188    0.000
##     Q4                0.598    0.051   11.820    0.000
##     Q5                0.570    0.047   12.163    0.000
##     Q6                0.415    0.034   12.331    0.000
##     Q7                0.450    0.061    7.317    0.000
##     Q8                0.455    0.069    6.632    0.000
##     Q9                0.353    0.057    6.224    0.000
##     Q10               0.419    0.043    9.732    0.000
```

```
##   xi1 =~
##     Q1                0.520    0.038   13.799    0.000
##     Q2                1.014    0.034   29.725    0.000
##     Q3                0.637    0.070    9.062    0.000
##     Q4                0.473    0.061    7.802    0.000
##   xi2 =~
##     Q5                0.576    0.048   12.086    0.000
##     Q6                0.261    0.031    8.308    0.000
##     Q7                0.747    0.039   19.312    0.000
##     Q8                0.837    0.038   22.000    0.000
##     Q9                0.658    0.034   19.304    0.000
##   xi3 =~
##     Q10               0.836    0.060   13.952    0.000
##     Q6                0.203    0.028    7.228    0.000
##     Q1                0.314    0.037    8.607    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   G ~~
##     xi1               0.000
##     xi2               0.000
##     xi3               0.000
##   xi1 ~~
##     xi2               0.694    0.036   19.205    0.000
##     xi3               0.409    0.058    7.102    0.000
##   xi2 ~~
##     xi3               0.430    0.052    8.334    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .Q1                0.570    0.023   24.772    0.000
##    .Q2                0.078    0.064    1.211    0.226
##    .Q3                0.296    0.016   19.016    0.000
##    .Q4                0.491    0.017   29.436    0.000
##    .Q5                0.589    0.017   33.958    0.000
##    .Q6                0.606    0.017   36.168    0.000
##    .Q7                0.378    0.012   30.431    0.000
##    .Q8                0.272    0.012   22.220    0.000
##    .Q9                0.776    0.021   37.153    0.000
##    .Q10               0.531    0.096    5.557    0.000
##     G                 1.000
##     xi1               1.000
##     xi2               1.000
##     xi3               1.000
```

```
#reliability, unidimensionality
{
  twofrel = round(semTools::reliability(bifactor_cfa)[-5,],2)
```

```
  twofrel = cbind(twofrel,NA)
  twofrel = cbind(twofrel, "two-factor")
  colnames(twofrel) = c("G", "xi_1","xi_2","xi_3","model")
  rownames(twofrel) = c("alpha","omega","omega_2","omega_3")

  threefrel = round(semTools::reliability(bifactor_cfa_4)[-5,],2)
  threefrel = cbind(threefrel,"three-factor")
  colnames(threefrel) = colnames(twofrel)
  rownames(threefrel) = rownames(twofrel)

  write.csv(rbind(twofrel,threefrel),
          file="composite_reliability.csv")


  }

#measurement invariance
{

  #fit MI models
  gender_configural = cfa(bifactor_model,
                        data=df_clean[,c(SCS_vars,"gender")],
                        group="gender")
  gender_weak = cfa(bifactor_model,
                        data=df_clean[,c(SCS_vars,"gender")],
                        group="gender",
                        group.equal=c("loadings") )
  gender_strong = cfa(bifactor_model,
                  data=df_clean[,c(SCS_vars,"gender")],
                  group="gender",
                  group.equal=c("loadings","intercepts") )
  gender_strict = cfa(bifactor_model,
                    data=df_clean[,c(SCS_vars,"gender")],
                    group="gender",
                    group.equal=c("loadings","intercepts","residuals") )



  mi_gender_modelcomp = anova(gender_configural,
                              gender_weak,
                              gender_strong,
                              gender_strict)

  mi_gender_compout=data.frame(cbind(DF=mi_gender_modelcomp$Df,
                        AIC=mi_gender_modelcomp$AIC,
                        BIC=mi_gender_modelcomp$BIC,
      Chisq=mi_gender_modelcomp$Chisq,
```

```
      Chisq_diff=mi_gender_modelcomp$`Chisq diff`,
      DF_diff=mi_gender_modelcomp$`Df diff`,
      p=mi_gender_modelcomp$`Pr(>Chisq)`))
rownames(mi_gender_compout) = c("configural MI",
                                "weak MI",
                                "strong MI",
                                "strict MI")
write.csv(mi_gender_compout,"mi_gender_compout.csv")



df_agegroups = df_clean[,c(SCS_vars)]
df_agegroups$age = df_clean$age > median(df_clean$age)
age_configural = cfa(bifactor_model_4,
                        data=df_agegroups[,c(SCS_vars,"age")],
                        group="age")
age_weak = cfa(bifactor_model_4,
                  data=df_agegroups[,c(SCS_vars,"age")],
                  group="age",
                  group.equal=c("loadings") )
age_strong = cfa(bifactor_model_4,
                    data=df_agegroups[,c(SCS_vars,"age")],
                    group="age",
                    group.equal=c("loadings","intercepts") )
age_strict = cfa(bifactor_model_4,
                    data=df_agegroups[,c(SCS_vars,"age")],
                    group="age",
                    group.equal=c("loadings","intercepts","residuals") )

mi_age_modelcomp = anova(age_configural,
                            age_weak,
                            age_strong,
                            age_strict)

mi_age_compout=data.frame(cbind(DF=mi_age_modelcomp$Df,
                                   AIC=mi_age_modelcomp$AIC,
                                   BIC=mi_age_modelcomp$BIC,
                                   Chisq=mi_age_modelcomp$Chisq,
                                   Chisq_diff=mi_age_modelcomp$`Chisq diff`,
                                   DF_diff=mi_age_modelcomp$`Df diff`,
                                   p=mi_age_modelcomp$`Pr(>Chisq)`))
rownames(mi_age_compout) = c("configural MI",
                                "weak MI",
                                "strong MI",
                                "strict MI")
write.csv(mi_age_compout,"mi_age_compout.csv")
```

```
  #factor structure (just a reminder for
  #modification index interpretation):
  # G =~ Q1+Q2+Q3+Q4+Q5+Q6+Q7+Q8+Q9+Q10
  # xi1 =~ Q1+Q2+Q3+Q4
  # xi2 =~ Q5+Q6+Q7+Q8+Q9
  # xi3 =~ Q10+Q6+Q1

  #look at mod. indices of strong invariance model
  modindices(gender_weak)
  modindices(age_weak)


}
```

```
## Warning in lav_object_post_check(object): lavaan WARNING: some estimated ov
## variances are negative
```

```
##     lhs op rhs block group level     mi    epc sepc.lv sepc.all sepc.nox
## 1     G =~  Q1     1     1     1  2.748  0.194   0.061    0.057    0.057
## 11  xi1 =~  Q1     1     1     1  4.307  0.131   0.071    0.066    0.066
## 15  xi2 =~  Q5     1     1     1  0.460  0.038   0.022    0.020    0.020
## 20  xi3 =~ Q10     1     1     1  3.230 -0.340  -0.282   -0.239   -0.239
## 23    G ~~ xi1     1     1     1  0.120  0.003   0.016    0.016    0.016
## 24    G ~~ xi2     1     1     1  1.242  0.009   0.049    0.049    0.049
## 25    G ~~ xi3     1     1     1  0.864 -0.015  -0.056   -0.056   -0.056
## 57    G =~  Q1     2     2     1  2.748 -0.194  -0.059   -0.054   -0.054
## 67  xi1 =~  Q1     2     2     1  4.307 -0.131  -0.069   -0.063   -0.063
## 71  xi2 =~  Q5     2     2     1  0.460 -0.038  -0.022   -0.020   -0.020
## 76  xi3 =~ Q10     2     2     1  3.229  0.340   0.298    0.251    0.251
## 79    G ~~ xi1     2     2     1  0.120 -0.003  -0.016   -0.016   -0.016
## 80    G ~~ xi2     2     2     1  1.242 -0.008  -0.047   -0.047   -0.047
## 81    G ~~ xi3     2     2     1  0.864  0.014   0.052    0.052    0.052
## 131 xi1 =~  Q5     1     1     1  1.130  0.056   0.030    0.027    0.027
## 132 xi1 =~  Q6     1     1     1  0.271 -0.025  -0.014   -0.014   -0.014
## 133 xi1 =~  Q7     1     1     1  1.579 -0.053  -0.029   -0.027   -0.027
## 134 xi1 =~  Q8     1     1     1  1.358 -0.052  -0.028   -0.026   -0.026
## 135 xi1 =~  Q9     1     1     1  7.909  0.148   0.080    0.071    0.071
## 136 xi1 =~ Q10     1     1     1  1.539 -0.170  -0.092   -0.078   -0.078
## 137 xi2 =~  Q1     1     1     1  0.044  0.010   0.006    0.006    0.006
## 138 xi2 =~  Q2     1     1     1  0.865 -0.068  -0.040   -0.037   -0.037
## 139 xi2 =~  Q3     1     1     1  0.076  0.014   0.008    0.008    0.008
## 140 xi2 =~  Q4     1     1     1  0.495  0.036   0.021    0.020    0.020
## 141 xi2 =~ Q10     1     1     1  0.384 -0.075  -0.044   -0.038   -0.038
## 142 xi3 =~  Q2     1     1     1  8.945  0.180   0.149    0.139    0.139
## 143 xi3 =~  Q3     1     1     1  0.000  0.001   0.001    0.001    0.001
## 144 xi3 =~  Q4     1     1     1  9.187 -0.111  -0.092   -0.088   -0.088
## 145 xi3 =~  Q5     1     1     1 18.720  0.150   0.124    0.112    0.112
## 146 xi3 =~  Q7     1     1     1  5.908 -0.072  -0.060   -0.056   -0.056
```

```
## 147 xi3 =~  Q8      1      1      1  9.139 -0.091  -0.075   -0.070   -0.070
## 148 xi3 =~  Q9      1      1      1  9.461  0.115   0.095    0.085    0.085
## 149  Q1 ~~  Q2      1      1      1  1.440  0.030   0.030    0.131    0.131
## 150  Q1 ~~  Q3      1      1      1 10.279  0.048   0.048    0.124    0.124
## 151  Q1 ~~  Q4      1      1      1  5.222 -0.036  -0.036   -0.069   -0.069
## 152  Q1 ~~  Q5      1      1      1  0.293  0.009   0.009    0.016    0.016
## 153  Q1 ~~  Q6      1      1      1  4.728  0.042   0.042    0.074    0.074
## 154  Q1 ~~  Q7      1      1      1  0.273  0.007   0.007    0.015    0.015
## 155  Q1 ~~  Q8      1      1      1  1.039 -0.014  -0.014   -0.037   -0.037
## 156  Q1 ~~  Q9      1      1      1  3.674 -0.033  -0.033   -0.053   -0.053
## 157  Q1 ~~ Q10      1      1      1 13.314 -0.178  -0.178   -0.331   -0.331
## 158  Q2 ~~  Q3      1      1      1  8.149 -0.058  -0.058   -0.362   -0.362
## 159  Q2 ~~  Q4      1      1      1  0.254 -0.009  -0.009   -0.042   -0.042
## 160  Q2 ~~  Q5      1      1      1 15.274  0.070   0.070    0.303    0.303
## 161  Q2 ~~  Q6      1      1      1  2.739 -0.028  -0.028   -0.117   -0.117
## 162  Q2 ~~  Q7      1      1      1  2.091 -0.020  -0.020   -0.111   -0.111
## 163  Q2 ~~  Q8      1      1      1  3.110 -0.026  -0.026   -0.174   -0.174
## 164  Q2 ~~  Q9      1      1      1  4.084  0.035   0.035    0.138    0.138
## 165  Q2 ~~ Q10      1      1      1 13.299  0.118   0.118    0.529    0.529
## 166  Q3 ~~  Q4      1      1      1  9.789  0.072   0.072    0.199    0.199
## 167  Q3 ~~  Q5      1      1      1 24.268 -0.078  -0.078   -0.201   -0.201
## 168  Q3 ~~  Q6      1      1      1  0.146  0.006   0.006    0.015    0.015
## 169  Q3 ~~  Q7      1      1      1  0.032 -0.002  -0.002   -0.007   -0.007
## 170  Q3 ~~  Q8      1      1      1  0.710  0.010   0.010    0.038    0.038
## 171  Q3 ~~  Q9      1      1      1  6.528  0.040   0.040    0.093    0.093
## 172  Q3 ~~ Q10      1      1      1  3.128 -0.035  -0.035   -0.092   -0.092
## 173  Q4 ~~  Q5      1      1      1  0.316 -0.010  -0.010   -0.018   -0.018
## 174  Q4 ~~  Q6      1      1      1  5.176 -0.037  -0.037   -0.069   -0.069
## 175  Q4 ~~  Q7      1      1      1  3.327  0.024   0.024    0.058    0.058
## 176  Q4 ~~  Q8      1      1      1  3.371  0.023   0.023    0.066    0.066
## 177  Q4 ~~  Q9      1      1      1  4.877 -0.038  -0.038   -0.065   -0.065
## 178  Q4 ~~ Q10      1      1      1  2.008 -0.029  -0.029   -0.058   -0.058
## 179  Q5 ~~  Q6      1      1      1  5.784  0.040   0.040    0.069    0.069
## 180  Q5 ~~  Q7      1      1      1  0.190  0.006   0.006    0.014    0.014
## 181  Q5 ~~  Q8      1      1      1  0.066 -0.004  -0.004   -0.010   -0.010
## 182  Q5 ~~  Q9      1      1      1  7.356 -0.049  -0.049   -0.078   -0.078
## 183  Q5 ~~ Q10      1      1      1 10.479  0.067   0.067    0.124    0.124
## 184  Q6 ~~  Q7      1      1      1  1.686 -0.018  -0.018   -0.039   -0.039
## 185  Q6 ~~  Q8      1      1      1  3.121 -0.023  -0.023   -0.062   -0.062
## 186  Q6 ~~  Q9      1      1      1 16.389  0.071   0.071    0.110    0.110
## 187  Q6 ~~ Q10      1      1      1  5.676  0.077   0.077    0.139    0.139
## 188  Q7 ~~  Q8      1      1      1  1.748  0.022   0.022    0.075    0.075
## 189  Q7 ~~  Q9      1      1      1  1.794 -0.021  -0.021   -0.043   -0.043
## 190  Q7 ~~ Q10      1      1      1  3.234 -0.032  -0.032   -0.073   -0.073
## 191  Q8 ~~  Q9      1      1      1  0.863  0.015   0.015    0.037    0.037
## 192  Q8 ~~ Q10      1      1      1  2.850 -0.030  -0.030   -0.083   -0.083
## 193  Q9 ~~ Q10      1      1      1  3.367  0.041   0.041    0.068    0.068
## 194 xi1 =~  Q5      2      2      1  0.768  0.047   0.025    0.022    0.022
```

```
## 195 xi1 =~  Q6      2      2      1  3.329 -0.094  -0.050   -0.051   -0.051
## 196 xi1 =~  Q7      2      2      1  0.825  0.040   0.021    0.020    0.020
## 197 xi1 =~  Q8      2      2      1  0.262 -0.024  -0.012   -0.011   -0.011
## 198 xi1 =~  Q9      2      2      1  0.278 -0.029  -0.016   -0.013   -0.013
## 199 xi1 =~ Q10      2      2      1  8.121  0.410   0.217    0.183    0.183
## 200 xi2 =~  Q1      2      2      1  1.224 -0.061  -0.036   -0.033   -0.033
## 201 xi2 =~  Q2      2      2      1  2.428  0.116   0.068    0.064    0.064
## 202 xi2 =~  Q3      2      2      1  0.496 -0.035  -0.021   -0.019   -0.019
## 203 xi2 =~  Q4      2      2      1  0.009 -0.005  -0.003   -0.003   -0.003
## 204 xi2 =~ Q10      2      2      1  2.161  0.204   0.120    0.101    0.101
## 205 xi3 =~  Q2      2      2      1  0.104  0.018   0.016    0.015    0.015
## 206 xi3 =~  Q3      2      2      1  0.018 -0.004  -0.004   -0.004   -0.004
## 207 xi3 =~  Q4      2      2      1  0.696 -0.027  -0.024   -0.023   -0.023
## 208 xi3 =~  Q5      2      2      1 13.751  0.119   0.104    0.093    0.093
## 209 xi3 =~  Q7      2      2      1  2.387 -0.044  -0.038   -0.036   -0.036
## 210 xi3 =~  Q8      2      2      1  4.128 -0.059  -0.052   -0.047   -0.047
## 211 xi3 =~  Q9      2      2      1  7.636  0.101   0.088    0.074    0.074
## 212  Q1 ~~  Q2      2      2      1  0.351  0.014   0.014    0.056    0.056
## 213  Q1 ~~  Q3      2      2      1 18.487 -0.065  -0.065   -0.152   -0.152
## 214  Q1 ~~  Q4      2      2      1  4.800  0.034   0.034    0.064    0.064
## 215  Q1 ~~  Q5      2      2      1  0.369  0.010   0.010    0.017    0.017
## 216  Q1 ~~  Q6      2      2      1  1.440  0.023   0.023    0.039    0.039
## 217  Q1 ~~  Q7      2      2      1  1.566  0.017   0.017    0.036    0.036
## 218  Q1 ~~  Q8      2      2      1  0.409  0.009   0.009    0.022    0.022
## 219  Q1 ~~  Q9      2      2      1 15.629 -0.073  -0.073   -0.104   -0.104
## 220  Q1 ~~ Q10      2      2      1  7.020  0.131   0.131    0.249    0.249
## 221  Q2 ~~  Q3      2      2      1  5.068  0.044   0.044    0.227    0.227
## 222  Q2 ~~  Q4      2      2      1  1.518 -0.021  -0.021   -0.089   -0.089
## 223  Q2 ~~  Q5      2      2      1  0.018 -0.002  -0.002   -0.009   -0.009
## 224  Q2 ~~  Q6      2      2      1  0.316 -0.009  -0.009   -0.034   -0.034
## 225  Q2 ~~  Q7      2      2      1  0.023  0.002   0.002    0.010    0.010
## 226  Q2 ~~  Q8      2      2      1  1.740 -0.020  -0.020   -0.105   -0.105
## 227  Q2 ~~  Q9      2      2      1  5.698  0.043   0.043    0.137    0.137
## 228  Q2 ~~ Q10      2      2      1  0.485 -0.020  -0.020   -0.086   -0.086
## 229  Q3 ~~  Q4      2      2      1  0.280  0.011   0.011    0.029    0.029
## 230  Q3 ~~  Q5      2      2      1  0.111  0.005   0.005    0.012    0.012
## 231  Q3 ~~  Q6      2      2      1  0.631  0.012   0.012    0.028    0.028
## 232  Q3 ~~  Q7      2      2      1  1.296 -0.015  -0.015   -0.041   -0.041
## 233  Q3 ~~  Q8      2      2      1  0.231 -0.006  -0.006   -0.019   -0.019
## 234  Q3 ~~  Q9      2      2      1  2.369  0.026   0.026    0.050    0.050
## 235  Q3 ~~ Q10      2      2      1  3.706  0.037   0.037    0.095    0.095
## 236  Q4 ~~  Q5      2      2      1  0.846  0.016   0.016    0.029    0.029
## 237  Q4 ~~  Q6      2      2      1 13.713 -0.060  -0.060   -0.108   -0.108
## 238  Q4 ~~  Q7      2      2      1  0.879  0.013   0.013    0.029    0.029
## 239  Q4 ~~  Q8      2      2      1  0.986  0.013   0.013    0.034    0.034
## 240  Q4 ~~  Q9      2      2      1  5.437 -0.042  -0.042   -0.066   -0.066
## 241  Q4 ~~ Q10      2      2      1  0.607 -0.015  -0.015   -0.031   -0.031
## 242  Q5 ~~  Q6      2      2      1 13.738  0.063   0.063    0.102    0.102
```

```
## 243  Q5 ~~  Q7      2      2      1  0.815  0.014   0.014    0.029    0.029
## 244  Q5 ~~  Q8      2      2      1  2.360 -0.024  -0.024   -0.056   -0.056
## 245  Q5 ~~  Q9      2      2      1 21.424 -0.090  -0.090   -0.127   -0.127
## 246  Q5 ~~ Q10      2      2      1  7.436  0.055   0.055    0.103    0.103
## 247  Q6 ~~  Q7      2      2      1  0.039 -0.003  -0.003   -0.006   -0.006
## 248  Q6 ~~  Q8      2      2      1 10.693 -0.046  -0.046   -0.106   -0.106
## 249  Q6 ~~  Q9      2      2      1 16.172  0.076   0.076    0.104    0.104
## 250  Q6 ~~ Q10      2      2      1 10.511 -0.102  -0.102   -0.189   -0.189
## 251  Q7 ~~  Q8      2      2      1 14.108  0.066   0.066    0.192    0.192
## 252  Q7 ~~  Q9      2      2      1 11.827 -0.060  -0.060   -0.103   -0.103
## 253  Q7 ~~ Q10      2      2      1  7.883 -0.049  -0.049   -0.113   -0.113
## 254  Q8 ~~  Q9      2      2      1  4.036  0.036   0.036    0.072    0.072
## 255  Q8 ~~ Q10      2      2      1  2.408 -0.028  -0.028   -0.073   -0.073
## 256  Q9 ~~ Q10      2      2      1 19.178  0.101   0.101    0.159    0.159
```

12 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.Polytomo

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

Van de Schoot, Rens, Peter Lugtig, and Joop Hox. 2012. "A Checklist for Testing Measurement Invariance." *European Journal of Developmental Psychology* 9 (4): 486–92.

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

Xu, Kate. 2012. "Multiple Group Measurement: Invariance Analysis in Lavaan." *Cambridge Mass* 37.