Item Response Theory - Final Essay

Marius Keute

September 14, 2022

Contents

submitted to:

Dr. Stefano Noventa

University of Tübingen

submitted by:

Marius Keute (QDS, 5991873)

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

1 Introduction

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

The 10 items are (Kalichman and Rompa (2001)): 1. My sexual appetite has gotten in the way of my relationships. 2. My sexual thoughts and behaviors are causing problems in my life. 3. My desires to have sex have disrupted my daily life. 4. I sometimes fail to meet my commitments and responsibilities because of my sexual behaviors. 5. I sometimes get so horny I could lose control. 6. I find myself thinking about sex while at work. 7. I feel that sexual thoughts and feelings are stronger than I am. 8. I have to struggle to control my sexual thoughts and behavior. 9. I think about sex more than I would like to. 10. It has been difficult for me to find sex partners who desire having sex as much as I want to.

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

2 Data preparation

The dataset (Kalichman and Rompa (1995)) consisted 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 were 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" were set to missing values. Of the remaining cases, 2295 (68.5%) reported gender "1" and 1053 (31.4%) reported gender "2". The SCS data contained at least one missing value for 133 cases.

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

```
knitr::include_graphics("missingplot.pdf")
```

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

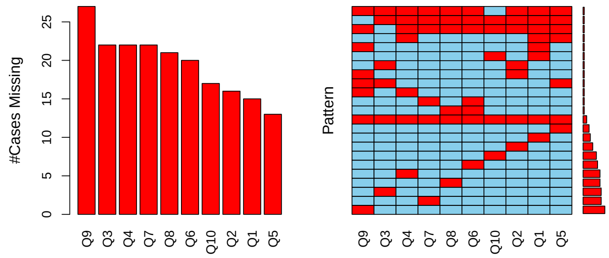

Figure 1: Pattern of missing SCS values.

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

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

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

The distribution of the dichotomized items is shown in 2 Since most variables' median was 2, this was not much different from an item-wise median threshold (see 1). Subsequently, I calculated biserial correlations between all pairs of dichotomized items. Moreover, I calculated item discrimination, i.e., each items ability to discriminate between high- and low-scoring individuals, using the adjusted item-total correlation method (Reynolds and Livingston (2021)), i.e., by calculating biserial correlation coefficients between each (dichotomized) item's scores and the sum of all other (dichotomized) items.

#TODO make tables smaller (too wide) #TODO make captions and table referencing work

Descriptive characteristics of the 10 SCS items are shown in 1, the proportions of 'correct' responses, i.e., responses greater than 2, are shown in 2.

Item intercorrelations are shown in Figure 2 #TODO discuss

3 IRT modeling

3.1 Rasch model estimation

Next, I estimated a Rasch model for the SCS data using three different software implementations. The Rasch model is also known as the one-parameter logistic model. It models a given person's chances of solving a given item as a logistic function of the difference between the i^{th} item's difficulty β_i and the s^{th} person's ability θ_s,

Table 2: Distribution and discrimination of dichotomized items

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
percent in category 1	40.6	37.8	37.3	27.1	38.2	71.9	37.8	41.3	49.4	50.8
number of cases in category 1	1366	1274	1256	912	1285	2421	1272	1392	1663	1711
discrimination	0.45	0.45	0.44	0.34	0.29	0.26	0.42	0.37	0.31	0.36

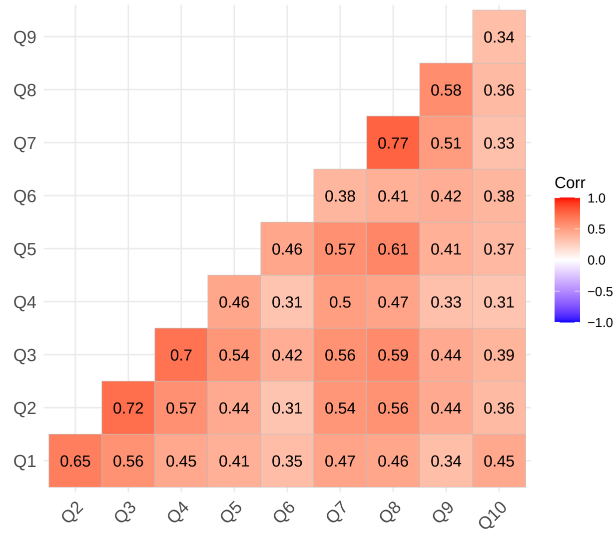

Figure 2: Pattern of missing SCS values.

where β_i and θ_s are latent (unobserved) quantities that are estimated from the dichotomous (solved vs. not solved) item data.

The first method was the one implemented in the R package `eRm` (Mair and Hatzinger (2007))

The second method was a generalized linear mixed regression model with a logistic link function as implemented in `lme4` (Bates et al. (2015)). `lme4` is a popular `R` package for fitting linear and generalized linear mixed-effects models, and as such it can be used to estimate the Rasch model, which is in essence a multilevel logistic regression model (Doran et al. (2007)). I modeled items as fixed effects and subjects as random effects, without an int, resulting in the model formula `solved~`.

The third method was a structural equation model as implemented in `lavaan` (Rosseel (2012), Templin (2022)).

3.2 Model analysis

ICC curves Fit indices

3.3 Alternative 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).

4 Data Dimensionality

5 Reliability and Measurement Invariance

6 Theoretical Exercise: Assumptions and Limits of IRT models

6.1 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. The traditional view of psychological tests (Classical Test Theory, CTT) conceived a given person's total score across all items of a test as an additive combination of the person's true score and a testing error: $X_i = \tau_i + \epsilon_i$ ((**linden1997item?**)).

6.2 Assumptions of IRT models

6.3 Limitations of IRT models

7 Analysis code

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

```
require(ggplot2)
require(ggthemes)
require(reshape2)
require(readxl)
require(VIM)
require(mice)
require(dplyr)
require(tidyr)
require(psych)
require(ggcorrplot)
require(eRm)
require(lme4)
require(lavaan)
require(patchwork)

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

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

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


#missing data motifs
# and missing proportion per item
pdf("missingplot.pdf",width = 8, height = 4)
aggr(df[!complete.cases(df[,SCS_vars]),SCS_vars],
     numbers=TRUE, sortVars=TRUE,prop=FALSE,
     labels=SCS_vars,
     ylab=c("#Cases Missing","Pattern"))
dev.off()

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

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


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

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

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


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

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

#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
#####
{

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

  discrimination = c()
  for (item in 1:10){
```

```
    itemname = SCS_vars[item]
    discrimination[itemname] = as.character(round(biserial(rowSums(dich[,-item]),dich[,item]),2))
  }

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

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

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

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

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

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

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

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

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

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

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

```
#####
#part 2: estimate Rasch model
#####
{
  #approach 1: eRm
  #prepare data for eRm estimation
  #(just item data in wide format)
  data_for_eRm = dich[,1:10]
  rasch_model_eRm = RM(data_for_eRm)
  smr_eRm = summary(rasch_model_eRm)

  #approach 2: lme4
  #prepare data for lme4 estimation
  #(item and subject data in long format)
  data_for_lme4 = dich[,1:10]
  data_for_lme4$id = 1:nrow(data_for_lme4)
  data_for_lme4 = melt(data_for_lme4, id.vars = "id")
  rasch_model_lme4 = glmer(value~0+variable+(1|id), data = data_for_lme4,
                      family = binomial(link="logit"))
  smr_lme4 = summary(rasch_model_lme4)

  #TODO check: why are estimates different?
```

```r
#aproach 3: lavaan
#modified copy from https://jonathantemplin.com/wp-content/uploads/2022/02/EPSY906_Example05_Binary_I
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

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

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

  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(betas){
  df = data.frame(x=seq(-6,6,.01))
  for (i in 1:length(betas)){
    df[[SCS_vars[i]]] = logistic(df$x, -betas[i])
  }

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

```r
#make ICC plots
betas_eRm = rasch_model_eRm$betapar
iccplot_eRm=ICC_plot(betas_eRm)+ggtitle("eRm")



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

betas_lme4 = smr_lme4$coefficients[,"Estimate"]
iccplot_lme4 = ICC_plot(betas_lme4)+ggtitle("lme4")


#lavaan betas have perfect negative correlation with other estimates, why?
#(item easiness vs. item difficulty parms?)
betas_lavaan =   convertTheta2IRT(lavObject = rasch_model_lavaan)$b
iccplot_lavaan=ICC_plot(betas_lavaan)+ggtitle("lavaan")



betas = rbind( data.frame(model="eRm",
                          item=factor(paste0("Q",as.character(1:10))),
                          beta=as.numeric(betas_eRm)),
              data.frame(model="lme4",
                         item=factor(paste0("Q",as.character(1:10))),
                         beta=as.numeric(betas_lme4)),
              data.frame(model="lavaan",
                         item=factor(paste0("Q",as.character(1:10))),
                         beta=as.numeric(betas_lavaan)),
              data.frame(model="CTT",
                         item=factor(paste0("Q",as.character(1:10))),
                         beta=as.numeric(dich.distro[1,])/100))
betas_plot = ggplot(betas,aes(x=item,y=beta,color=model,group=model)) +
  geom_point() + geom_line() + theme_clean() + ggtitle("model comparison")

#arrange plots vertically and save
iccplot_eRm/iccplot_lme4/iccplot_lavaan/betas_plot
ggsave("iccfig.pdf",width = 6,height = 8)
#compare fits
summary(rasch_model_eRm)
smr_lavaan$FIT
smr_lme4$AICtab

}
```

```
##
## Results of RM estimation:
##
## Call:  RM(X = data_for_eRm)
##
## Conditional log-likelihood: -9888.878
## Number of iterations: 8
```

```
## Number of parameters: 9
##
## Item (Category) Difficulty Parameters (eta): with 0.95 CI:
##     Estimate Std. Error lower CI upper CI
## Q2     0.359      0.043    0.275    0.444
## Q3     0.404      0.043    0.320    0.489
## Q4     1.143      0.046    1.053    1.234
## Q5     0.337      0.043    0.253    0.421
## Q6    -1.999      0.049   -2.096   -1.902
## Q7     0.361      0.043    0.277    0.446
## Q8     0.127      0.042    0.043    0.210
## Q9    -0.409      0.042   -0.492   -0.326
## Q10   -0.501      0.042   -0.584   -0.418
##
## Item Easiness Parameters (beta) with 0.95 CI:
##          Estimate Std. Error lower CI upper CI
## beta Q1    -0.176      0.043   -0.260   -0.093
## beta Q2    -0.359      0.043   -0.444   -0.275
## beta Q3    -0.404      0.043   -0.489   -0.320
## beta Q4    -1.143      0.046   -1.234   -1.053
## beta Q5    -0.337      0.043   -0.421   -0.253
## beta Q6     1.999      0.049    1.902    2.096
## beta Q7    -0.361      0.043   -0.446   -0.277
## beta Q8    -0.127      0.042   -0.210   -0.043
## beta Q9     0.409      0.042    0.326    0.492
## beta Q10    0.501      0.042    0.418    0.584

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.44261 (tol = 0.002, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unide
##  - Rescale variables?

## lavaan 0.6-9 ended normally after 15 iterations
##
##   Estimator                                         DWLS
##   Optimization method                             NLMINB
##   Number of model parameters                          20
##   Number of equality constraints                       9
##
##   Number of observations                            3368
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                              1421.723    1342.285
##   Degrees of freedom                                44          44
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  1.066
##   Shift parameter                                            9.144
##        simple second-order correction (WLSMV)
##
## Model Test Baseline Model:
##
##   Test statistic                             38946.468   24217.331
##   Degrees of freedom                                45          45
```

```
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.609
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.965       0.946
##   Tucker-Lewis Index (TLI)                       0.964       0.945
##
##   Robust Comparative Fit Index (CFI)                            NA
##   Robust Tucker-Lewis Index (TLI)                               NA
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.096       0.094
##   90 Percent confidence interval - lower         0.092       0.089
##   90 Percent confidence interval - upper         0.101       0.098
##   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                                           0.100       0.100
##
## Weighted Root Mean Square Residual:
##
##   WRMR                                           5.084       5.084
##
## Parameter Estimates:
##
##   Standard errors                           Robust.sem
##   Information                                 Expected
##   Information saturated (h1) model        Unstructured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   SCS =~
##     Q1         (l)    1.216    0.022   55.917    0.000    1.216    0.772
##     Q2         (l)    1.216    0.022   55.917    0.000    1.216    0.772
##     Q3         (l)    1.216    0.022   55.917    0.000    1.216    0.772
##     Q4         (l)    1.216    0.022   55.917    0.000    1.216    0.772
##     Q5         (l)    1.216    0.022   55.917    0.000    1.216    0.772
##     Q6         (l)    1.216    0.022   55.917    0.000    1.216    0.772
##     Q7         (l)    1.216    0.022   55.917    0.000    1.216    0.772
##     Q8         (l)    1.216    0.022   55.917    0.000    1.216    0.772
##     Q9         (l)    1.216    0.022   55.917    0.000    1.216    0.772
##     Q10        (l)    1.216    0.022   55.917    0.000    1.216    0.772
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     SCS               0.000                               0.000    0.000
##    .Q1                0.000                               0.000    0.000
```

```
##    .Q2               0.000                               0.000    0.000
##    .Q3               0.000                               0.000    0.000
##    .Q4               0.000                               0.000    0.000
##    .Q5               0.000                               0.000    0.000
##    .Q6               0.000                               0.000    0.000
##    .Q7               0.000                               0.000    0.000
##    .Q8               0.000                               0.000    0.000
##    .Q9               0.000                               0.000    0.000
##    .Q10              0.000                               0.000    0.000
##
## Thresholds:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     Q1|t1            0.377    0.035   10.888    0.000    0.377    0.240
##     Q2|t1            0.488    0.035   14.001    0.000    0.488    0.310
##     Q3|t1            0.515    0.035   14.767    0.000    0.515    0.327
##     Q4|t1            0.962    0.037   26.095    0.000    0.962    0.611
##     Q5|t1            0.475    0.035   13.656    0.000    0.475    0.301
##     Q6|t1           -0.915    0.036  -25.539    0.000   -0.915   -0.581
##     Q7|t1            0.489    0.035   14.041    0.000    0.489    0.311
##     Q8|t1            0.347    0.035   10.052    0.000    0.347    0.221
##     Q9|t1            0.025    0.034    0.723    0.470    0.025    0.016
##     Q10|t1          -0.030    0.034   -0.896    0.370   -0.030   -0.019
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     SCS              1.000                               1.000    1.000
##    .Q1               1.000                               1.000    0.404
##    .Q2               1.000                               1.000    0.404
##    .Q3               1.000                               1.000    0.404
##    .Q4               1.000                               1.000    0.404
##    .Q5               1.000                               1.000    0.404
##    .Q6               1.000                               1.000    0.404
##    .Q7               1.000                               1.000    0.404
##    .Q8               1.000                               1.000    0.404
##    .Q9               1.000                               1.000    0.404
##    .Q10              1.000                               1.000    0.404
##
## Scales y*:
##                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##     Q1               0.635                               0.635    1.000
##     Q2               0.635                               0.635    1.000
##     Q3               0.635                               0.635    1.000
##     Q4               0.635                               0.635    1.000
##     Q5               0.635                               0.635    1.000
##     Q6               0.635                               0.635    1.000
##     Q7               0.635                               0.635    1.000
##     Q8               0.635                               0.635    1.000
##     Q9               0.635                               0.635    1.000
##     Q10              0.635                               0.635    1.000
##
## R-Square:
##                   Estimate
##     Q1               0.596
##     Q2               0.596
```

```
##     Q3                  0.596
##     Q4                  0.596
##     Q5                  0.596
##     Q6                  0.596
##     Q7                  0.596
##     Q8                  0.596
##     Q9                  0.596
##     Q10                 0.596
```

```
##
## Results of RM estimation:
##
## Call:  RM(X = data_for_eRm)
##
## Conditional log-likelihood: -9888.878
## Number of iterations: 8
## Number of parameters: 9
##
## Item (Category) Difficulty Parameters (eta): with 0.95 CI:
##     Estimate Std. Error lower CI upper CI
## Q2     0.359      0.043    0.275    0.444
## Q3     0.404      0.043    0.320    0.489
## Q4     1.143      0.046    1.053    1.234
## Q5     0.337      0.043    0.253    0.421
## Q6    -1.999      0.049   -2.096   -1.902
## Q7     0.361      0.043    0.277    0.446
## Q8     0.127      0.042    0.043    0.210
## Q9    -0.409      0.042   -0.492   -0.326
## Q10   -0.501      0.042   -0.584   -0.418
##
## Item Easiness Parameters (beta) with 0.95 CI:
##          Estimate Std. Error lower CI upper CI
## beta Q1    -0.176      0.043   -0.260   -0.093
## beta Q2    -0.359      0.043   -0.444   -0.275
## beta Q3    -0.404      0.043   -0.489   -0.320
## beta Q4    -1.143      0.046   -1.234   -1.053
## beta Q5    -0.337      0.043   -0.421   -0.253
## beta Q6     1.999      0.049    1.902    2.096
## beta Q7    -0.361      0.043   -0.446   -0.277
## beta Q8    -0.127      0.042   -0.210   -0.043
## beta Q9     0.409      0.042    0.326    0.492
## beta Q10    0.501      0.042    0.418    0.584
```

```
##       AIC       BIC    logLik  deviance  df.resid
##  35929.51  36022.18 -17953.75  35907.51  33669.00
```

```
#alternative model: 2PL
{}
```

```
## NULL
```

```
#DIF
{}
```

```
## NULL
```

```r
#reliability, unidimensionality
{}
```

```
## NULL
```

```r
#measurement invariance
#polytomous IRT model
```

8 References

Bates, Douglas, Martin Mächler, Ben Bolker, and Steve Walker. 2015. "Fitting Linear Mixed-Effects Models Using lme4." *Journal of Statistical Software* 67 (1): 1–48. https://doi.org/10.18637/jss.v067.i01.

Doran, Harold, Douglas Bates, Paul Bliese, and Maritza Dowling. 2007. "Estimating the Multilevel Rasch Model: With the Lme4 Package." *Journal of Statistical Software* 20: 1–18.

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.

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

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

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

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