Appendix 1: Used R script

```
#Open R and RStudio.

#Set your preferred working directory (folder) clicking on “Session” on the top menu of RStudio, and then on “Set a Working Directory” and then “Choose Directory…”

#First, let’s install the library mirt on RStudio. On the bottom right window of RStudio, click on the “Packages” tab and then click on “Install”. Type mirt and the name of the package will appear for you to select. Make sure the box “Install dependencies” is marked. Now let’s load the package:
library(mirt)

#Let’s increase the size of the allowed output, so we don’t miss anything:
getOption("max.print")
options(max.print = 999999)

# Replace desired file with the file location
#And then we must load this dataset and give it a name (“data_test”):
library(readr)
data_test <- read_csv("Desired file")
View(scored)

#We are going to run the Rasch model first:
modelRasch <- mirt(data_test, 1, "Rasch")

#And we will ask plots of the item characteristic curves
plot(modelRasch, type="trace")
plot(modelRasch, type = 'trace', facet_items = FALSE)

#And the information curve:
plot(modelRasch, type="info")

#Now let’s run other models. Does the item parameter calibration process get faster or slower? Does it need more or less iterations with the addition of more parameters? Pay attention to that with the next commands:
model2PL <- mirt(data_test, 1, "2PL")

#Now let’s do the item characteristic curves generated in the other models model and compare with the curves from the Rasch model generated earlier. What changed?
plot(model2PL, type="trace")

#And let’s compare the information curves (indicators of reliability and inversely proportional to the standard error of measurement curve):
plot(model2PL, type="info")

#Interesting differences, right? Now let’s see how it looks like when all item characteristic curves are placed all together on the same graph on each model:
plot(modelRasch, type = 'trace', facet_items = FALSE)
plot(model2PL, type = 'trace', facet_items = FALSE)

plot(model2PL, type = 'info', facet_items = FALSE)
#Now we need to check dimensionality. Let’s do a principal component analysis of the residuals. If we find a dimension on the residuals, we might need to think of using a multidimensional model for this test. Let’s obtain the residual matrix.
Rasch_residuals <- residuals(modelRasch, type = "Q3")

#Now let’s compare the fit of the models
```

```
model_comparison_Rasch_2PL <- anova(modelRasch, model2PL)
View(model_comparison_Rasch_2PL)

#What can we interpret in terms of which model had the best fit? Now let’s see the result of my
item parameter calibration:
coef_Rasch <- coef(modelRasch)
coef_2PL <- coef(model2PL)
View(coef_Rasch)
View(coef_2PL)

#Now let’s also see the scores calculated by each model:
scoresRasch <- fscores(modelRasch, full.scores = TRUE, full.scores.SE = TRUE)
scores2PL <- fscores(model2PL, full.scores = TRUE, full.scores.SE = TRUE)
View(scoresRasch)
View(scores2PL)

coef(modelRasch, IRTpars = TRUE)
coef(model2PL, IRTpars = TRUE)

#Transport the values of the scores AND standard errors (SEs) to different folders in Excel. We
will need them to calculate individual reliability estimates later. But let’s calculate the reliability for
the group level:
empirical_rxx(scoresRasch)

#It is also possible to ask mirt to draw a conditional reliability curve (conditional because the
reliability depends on the position of the test taker on the theta scale):
plot(scoresRasch, type = "rxx")

#Now let’s evaluate the adequacy of the item fit indices to the Rasch model (infit, for “inliers” and
outfit, for “outliers”). Values above 1,2 cause concern but do not necessarily degrade the
measure. Values above 1.5 certainly degrade the measure and are strong candidates for
withdrawal. Standardized (z-transformations) values for the fit indices are also produced.
itemfit_Rasch <- itemfit(modelRasch, fit_stats = "infit")
View(itemfit_Rasch)

#Now let’s see the person fit measures:
personfit_Rasch <- personfit(modelRasch)
View(personfit_Rasch)

#Now let’s quickly calculate the descriptive statistics of the item and person measures using the
command describe from the package psych. You can do it with other packages too (such as
sapply, Hmisc or pastecs) and attribute an object to it, so that you can export all your fit results to
Excel.
library(psych)
item_fit_descriptives <- describe(itemfit_Rasch)
person_fit_descriptives <- describe(personfit_Rasch)
library(rio)
export(itemfit_Rasch, "Item_Fit_Full_Data.xlsx")
export(personfit_Rasch, "Person_Fit_Full_Data.xlsx")
export(item_fit_descriptives, "Item_Fit_Descriptives.xlsx")
export(person_fit_descriptives, "Person_Fit_Descriptives.xlsx")

plot(modelRasch, type = "rxx")
plot(model2PL, type = "rxx")

#Now let’s evaluate the adequacy of the item fit indices to the 2PL model (infit, for “inliers” and
outfit, for “outliers”). Values above 1,2 cause concern but do not necessarily degrade the
measure. Values above 1.5 certainly degrade the measure and are strong candidates for
withdrawal. Standardized (z-transformations) values for the fit indices are also produced.
itemfit_2PL <- itemfit(model2PL, fit_stats = "infit")
```

```
View(itemfit_2PL)

#Now let’s see the person fit measures:
personfit_2PL <- personfit(model2PL)
View(personfit_2PL)
export(itemfit_2PL,"itemfit_2PL.xlsx")

# Reliability coefficient:

empirical_rxx(scoresRasch)
empirical_rxx(scores2PL)
plot(modelRasch, type = "rxx")
```