Sustainable Finance Assessment Tool

1. Introduction

The aim of work package 4 is to build an online tool to assess sustainable finance literacy. We decided to design a computerized adaptive test (CAT) build with 24 items coming from the survey questions from work package 1 that are calibrated via classical item response theory (3PL model). The first step was to estimate the item parameters (discrimination, difficulty, and guessing) for the selected questions of the survey by means of IRT modeling tools in Python. We then performed some simulations of a CAT in R, using the R package catR in order to detect potential problems. Building on the simulation, we implemented an interactive assessment tool that can be run directly in R Studio. The tool presents items from the existing excel item bank, collects the answers from the console, selects subsequent items by using the Maximum Fisher Information criterion, and re-estimates the ability score $\hat{\theta}$ after each item. Since there are currently only relatively few items in the item bank of the tool, each test is constituted by exactly 10 questions, but this can easily be changed in a future extension of this work and alternative stopping rules can be implemented if desired. Each session produces a CSV file (assessment_log.csv) containing all useful information about the test.

The remainder of this text is organized as follows. In section 2 we present the main features of item response theory and explain how it can be used in adaptive testing. In section 3, we describe how the financial literacy assessment text was created and in section 4 we conclude. Section 5 contains the references and section 6 shows some screen plots explaining how the assessment tool looks like on computer.

2. Item Response Theory

Item Response Theory (IRT) is the standard measurement model commonly used to evaluate standardized tests (see Fischer (1974) or Weiss & Yoes (1991) for a more detailed presentation of this model). As with any mathematical model, there are certain assumptions that must be respected.

2.1 IRT assumptions

IRT is based on the following two assumptions:

- We assume that the probability of a correct answer is attributable to the subject's position on a specific number *k* of latent traits of competence required to answer the type of item in question. These latent traits can be conceptualized as a *k*-dimensional space. In our application, we assume that this space is one-dimensional, meaning that we are interested in measuring sustainable finance literacy as a function of just one parameter θ. We assume that the probability of a correct answer evolves as a logistic function whose inflection point is projected onto the latent trait of competence, thus allowing us to determine the difficulty of the item.
- We assume that there is local independence, meaning that the probability of a subject answering a given item correctly does not depend on the answers to the other items in the test. Thus, an item administered at the beginning of the test should not influence an item administered later, and the correlations between items should be due solely to the influence of the latent trait, which implies that the items are uncorrelated for subjects with the same level of competence.

2.2 The item response function

The central concept of item response theory is that of item characteristic curves (ICCs), also called item response functions (IRFs), which describe the probability of success on a given item as a function of the subjects' competence. For dichotomous models, the logistic function is most often used.

A subject's response to a dichotomous item *j* is coded by

$$x_j = \begin{cases} 1, \text{if the answer is correct} \\ 0, \text{if the answer is wrong} \end{cases}.$$

The probability of a correct answer for item *j* for a subject of competence θ is given by the item response function $P_j(\theta)$. We write

$$P_j(\theta) = \frac{1}{1+e^{-a_j(\theta-b_j)}},$$

where a_j is the item's discrimination (*item slope*) and b_j its difficulty (*item threshold*).

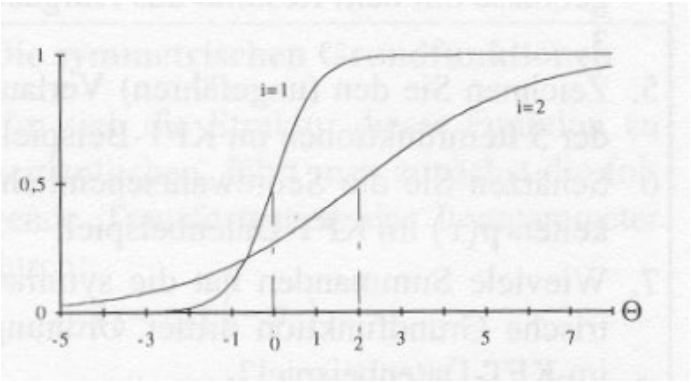

Figure 1: 2 IRFs with parameters a_1=0, b_1=2, and a_2=2, b_2=0.5 respectively (Rost, 1996).

$z_j = a_j(\theta - b_j)$ is called logit. It can also be written as $z_j = a_j\theta + c_j$, where $c_j = -a_j b_j$ is the item intercept.

If there are multiple-choice items in the test, a subject who does not know the correct answer can nevertheless answer correctly by choosing randomly from the proposed alternatives. The probability that a subject with a skill level of θ does not know the correct answer but answers correctly is $g_j\left[1 - \Psi_j(\theta)\right]$, where g_j denotes the probability of guessing correctly (for multiple-choice items g_j=1/A, if A is the number of possible answers to choose from).

The probability of a correct answer for item *j* for a subject with comptence level θ, is hence given by

$$\begin{aligned} P_j(\theta) &= g_j\left[1 - \Psi_j(\theta)\right] + \Psi_j(\theta) \\ &= g_j + (1 - g_j)\Psi_j(\theta). \end{aligned}$$

The interpretation of the parameters for two and three parameter item response models is illustrated in figures 2 and 3.

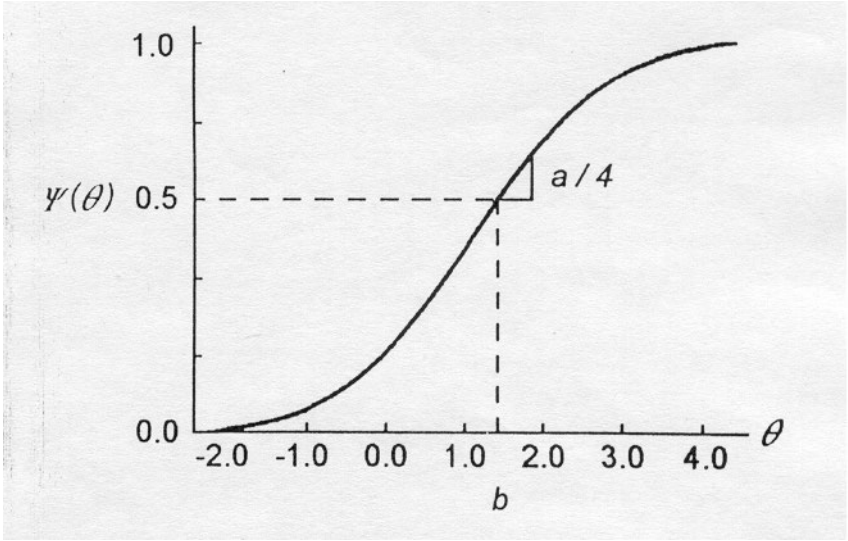

Figure 2: Interpretation of the parameters in a 2-parameter IRT model (Zimokski et ali, 1996)

• The item's difficulty *b* determines the position of the curve relative to the skill axis. It is equal to the x-coordinate of the inflection point of the item's response function. This is also the point where the slope of the response function is greatest. Therefore, the more difficult an item is, the more its curve is shifted to the right.

• The item's discrimination *a* is proportional to the slope of the response function. The steeper the slope, the better the item discriminates between two subjects whose skill levels are to the item's difficulty. When the slope is relatively shallow, the item shows low discrimination, but this discrimination occurs over a fairly wide range of skills.

• The "guessing" parameter *c*, also called the pseudo-chance score level, indicates the probability of giving the correct answer by chance. It is equal to the y-coordinate of the lower asymptote of the response function for the item and corresponds to the probability that a subject with very low competence (tending towards negative infinity) will give the correct answer.

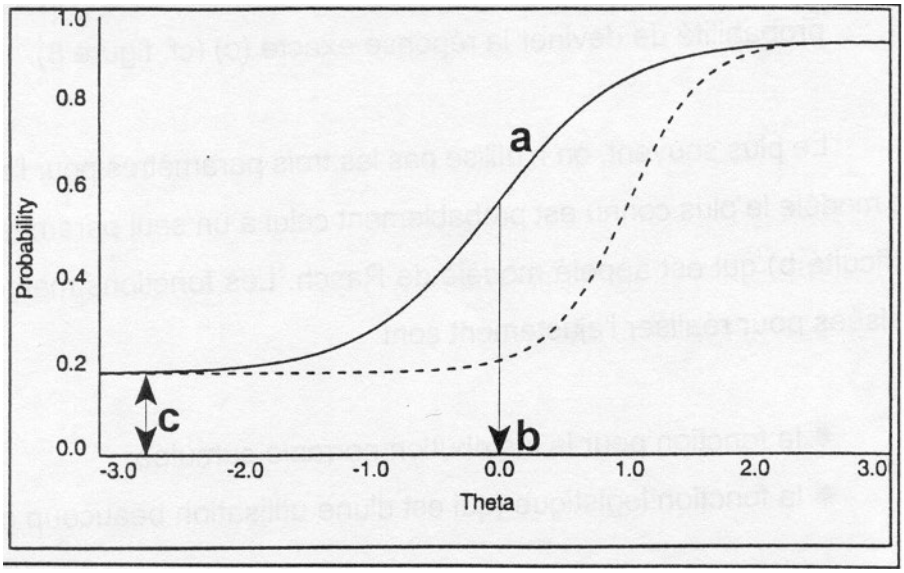

Figure 3: Interpretation of the parameters in a 3-parameter IRT model (Assessment Systems Corporation, 1995).

One of the major advantages of the IRT is that parameter estimation is independent of the population from which the estimation is derived. In other words, the distribution of skills in the sample used to calibrate the items has no effect on the parameter estimates. These parameters remain constant when estimated in groups with different skill levels.

This means that a similar measurement scale can be used in different populations and that subjects can be tested with different items, appropriate to their skill level, while maintaining the comparability of their scores (Anastasi, 1982).

Therefore, we obtain approximately the same item response functions, regardless of the skill distribution in the sample used to calculate the item parameters. Consequently, an IRT provides the probability that an individual will respond correctly to an item at a given skill level, without this probability depending on the number of subjects at that skill level. This invariance property of IRFs is an important characteristic of item response theory models.

2.3 The test response function

The local independence assumption implies that the IRFs are additive, meaning that they can be added together at each proficiency level. By summing the probability of a correct answer for each item at each proficiency level and dividing this sum by the number of items, we obtain an average of the IRFs. The resulting curve is called the test response function (TRF).

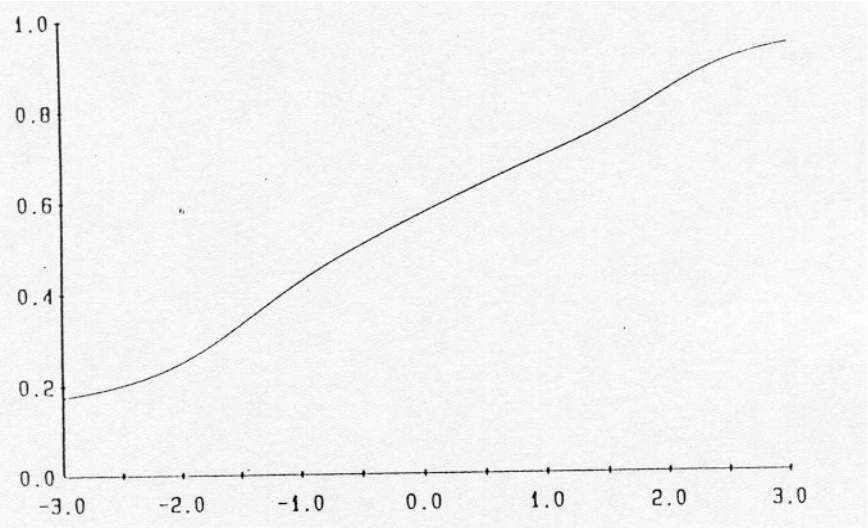

Figure 4: The TRF for a test with 6 items calibrated by a 3-parameter IRT model (Weiss & Yoes, 1991).

The TRF establishes a link between the type of score used in classical test theory, i.e. the number of correct answers, and the type of score used in IRT, i.e. the estimated level of competence for a given subject.

2.4 The item information function

The concept of information is as important as the concept of an item's response function. Information refers to the precision of what is being measured; it is inversely proportional to the measurement error, so the smaller the measurement error, the greater the information.

In IRT, it is possible to determine how much information each item provides at each point on the competence continuum. The more information available at a given competence level, the more precise the assessment of parameters at that level will be. One of the advantages of IRT is therefore

the ability to select items that provide maximum information at specific competence levels. In fact, it is the slope of the IRF that indicates the discriminatory power. The steeper this slope, the greater the probability of correctly answering an item, which varies according to competence, provided one is in the vicinity of the item's maximum slope point. Indeed, the information of an item at a competence level θ is given by

$$I(\theta) = a_j^2 P_j(\theta) \left[1 - \mathrm{P_j}(\theta)\right].$$

The item information function (IIF) thus obtained reaches its maximum at the point where the slope of the IRF is highest.

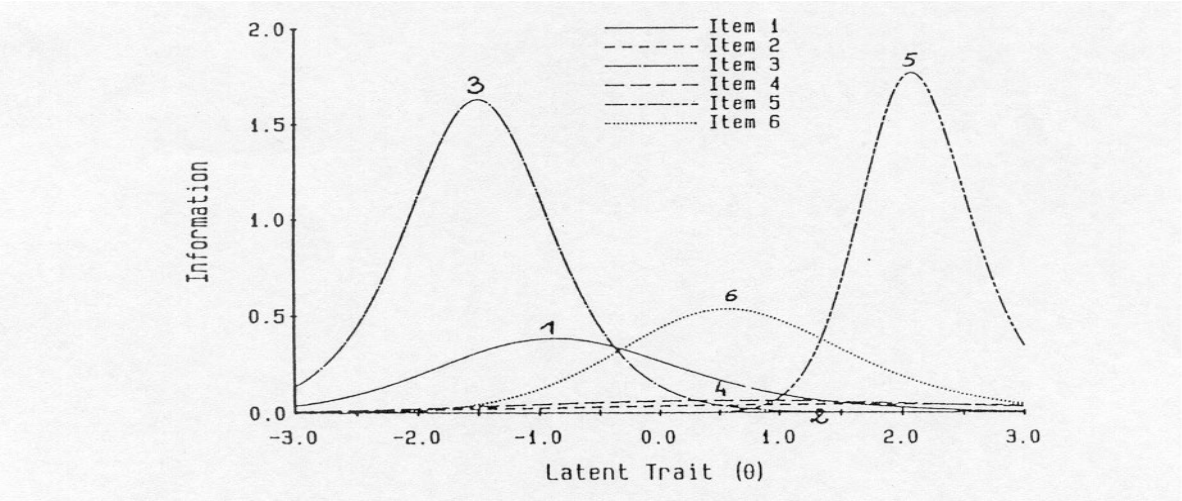

Figure 5: The IIFs of a test with 6 items (Weiss & Yoes, 1991).

It is not surprising that the maximum of the IIF depends on the item's difficulty. Indeed, if an item differentiates well between individuals with low skill levels, it will be easier than an item that differentiates well between individuals with high skill levels. The discrimination parameter influences the shape of the IIF, which implies a certain bandwidth dilemma of reliability: items with a peaked information function discriminate well, but over a narrow part of the latent continuum. If the IIF is flatter the discrimination of the item is lower, but with a higher bandwidth. This effect is known as the bandwidth paradox of reliability. It implies that, when constructing a test, a compromise must be made between the total information provided by an item and the range over which this information is available.

2.5 The test information function

As for the item response functions, the local independence assumption allows the summation of the information from different items for a given proficiency level. This yields the test information function (TIF), which indicates how much information the test provides at each proficiency level and thus represents a tool for determining the quality of the test.

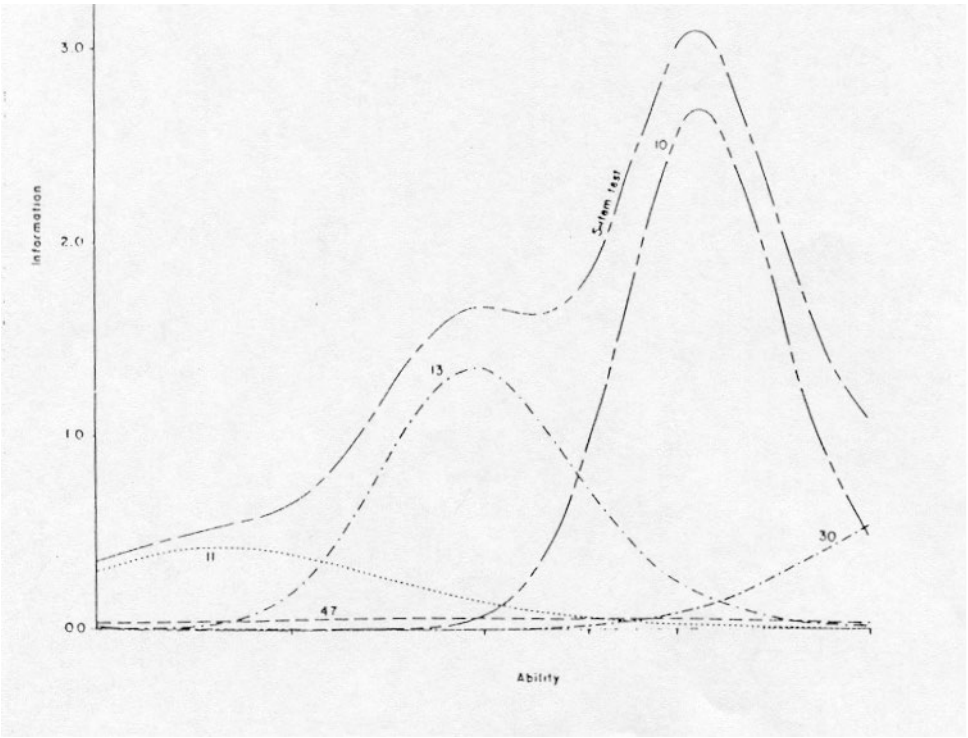

Figure 6: The IIFs and the TIF of a test with 5 items (dans Lord, 1980).

Whereas in classical test theory, the contribution of each item to the reliability or validity of the test depends on all the other items that make up the test, in IRT theory, the contribution of an item to the effectiveness of a test depends only on its intrinsic properties.

The TIF provides a way to identify the levels of the latent trait for which the test measures most accurately and those for which additional items need to be added. One of the most important uses of the concept of information in IRT is that the IIF and the TIF can be developed before the test is constructed.

2.6 Estimation of the competence levels

The subjects' proficiency levels can be estimated either by maximum likelihood estimation or Bayesian modal estimation.

Both methods consider all the information contained in a subject's response sequence to the test items. The response vector for each individual consists of a sequence of 1s and 0s, indicating whether they answered the different items correctly or not. Local independence means that the probability for a given vector is obtained by multiplication. It varies according to the proficiency level θ. This probability is called the likelihood function.

In the case of a single item, the likelihood function simply corresponds to 1 if the subject answered the item correctly, and to 1 otherwise. With two items, the likelihood function corresponds to the product of the two probabilities that reflect the subject's responses. In the general case, we simply multiply the probabilities corresponding to the answers given to the different items. Once we have obtained the likelihood function for a certain sequence of answers, we just need to compute which value θ maximizes the likelihood function for the given answer sequence.

2.6.1 Maximum likelihood estimation

The maximum likelihood estimation of the answers of subject *i* is the number θ maximizing the likelihood function L_i defined by

$$\log L_i(\theta) = \sum_j \{x_{ij} \log P_j(\theta) + (1 - x_{ij}) \log[1 - P_j(\theta)]\}.$$

Hence

$$\frac{\partial \log L_i(\theta)}{\partial \theta} = \sum_j \frac{x_{ij} - P_j(\theta)}{P_j(\theta)[1 - P_j(\theta)]} \frac{\partial P_j(\theta)}{\partial \theta} = 0,$$

which allows to determine the competence level θ.

In geometric terms, this amounts to determining the projection onto the x-axis of the maximum of the likelihood function.

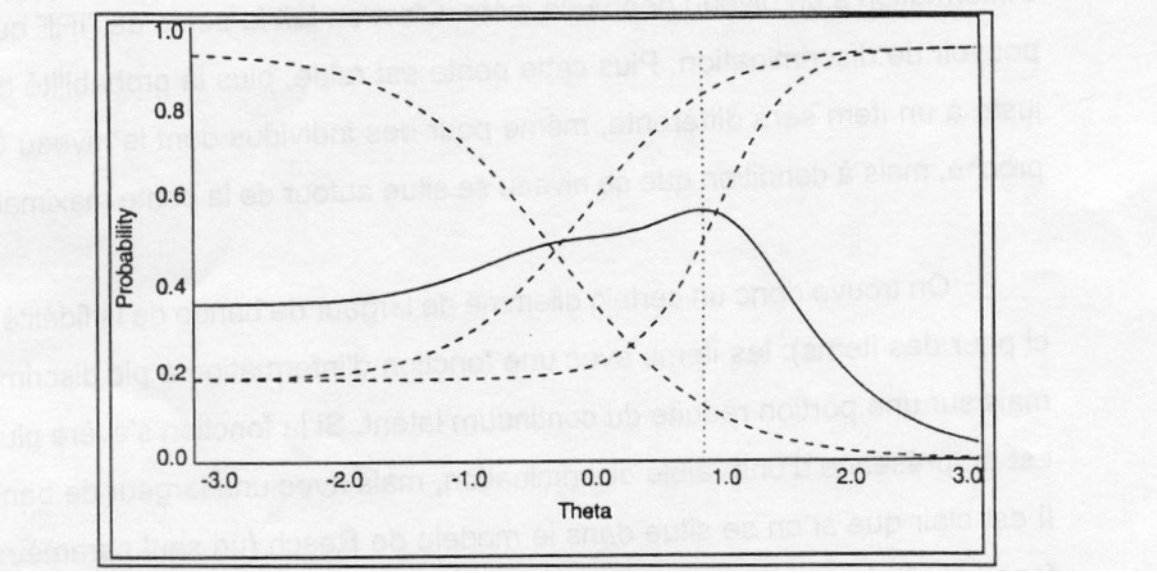

Figure 7: Representation of 3 IRFs (dashed lines) for two correctly answered and one incorrectly answered items and the corresponding likelihood function (full line) with a competence level determined by projection (Assessement Systems Corporation, 1995)

One problem with maximum likelihood estimation is that if a subject answers all items correctly (perfect vector) or all items incorrectly (zero vector), then the likelihood function does not admit a single, uniquely identifiable maximum. This problem is partially solved by Bayesian modal estimation.

2.6.2 Bayesian modal estimation

The basic idea of Bayesian modal estimation method is that if the distribution of θ is known or can be estimated for the population, then this information can be used to make more accurate estimates of θ.

A normal prior distribution is usually assumed for the latent trait. The likelihood function is then weighted (i.e. multiplied) by this prior distribution function. This results in a new distribution called the posterior distribution. The estimate of θ corresponds to the highest value of this posterior likelihood function, which is given by

$$\log L_i(\theta) = \sum_j \{x_{ij} \log P_j(\theta) + (1 - x_{ij}) \log[1 - P_j(\theta)]\} + \log g(\theta),$$

where $g(\theta)$ denotes the prior distribution of θ. So, the likelihood equation to resolve is

$$\sum_j \frac{x_{ij} - P_j(\theta)}{P_j(\theta)\left[1 - P_j(\theta)\right]} \frac{\partial P_j(\theta)}{\partial \theta} + \frac{\partial \log g(\theta)}{\partial \theta} = 0.$$

The problem with this estimation method is that if the prior distribution does not correspond to reality, the estimates are biased. The advantage is that estimates can be made for perfect or zero vectors.

The standard measurement error made when estimating θ using one of the two methods above is given by

$$\text{S.E.}(\theta) = \sqrt{\frac{1}{I(\theta)}},$$

where I(θ) denotes the test information function.

2.7 Estimation of the IRF parameters

Since all applications of IRT depend on item parameters, it is crucial that these parameters are estimated as accurately as possible. This item estimation process is generally called calibration. It can be performed using specialized software (such as LOGIST, BILOG, ASCAL, BICAL or as we choose to do it by one of the R packages designed for this). The selection of software for estimating item parameters depends on the form of the IRT chosen (1-, 2-, or 3-parameter model) and the desired estimation procedure (maximum likelihood or Bayesian).

A common characteristic of all software is that they must arbitrarily define the zero point of the continuum.

In two- or three-parameter models, the zero point is defined as the average skill level of the sample from which the item parameters are estimated. In the one-parameter model, the zero point is sometimes defined as the average difficulty level of the items. This actually violates the parameter invariance property.

2.8 Application of l'IRT to the creation of tests

The theoretical framework previously defined constitutes a very powerful tool for the construction and development of tests. The items whose parameters have been estimated form the item bank from which the test will be created. Unlike classical test theory, item response theory allows us to determine the statistical properties of the test and estimate its performance before it is administered for the first time. We can, in fact, calculate the test's information function and thus obtain the precision with which the test measures across the latent continuum. This makes it possible to determine the standard error of measurement of the test and to fill any gaps by very precisely introducing items with high discriminatory power where a significant measurement error has been observed.

In this methodological approach to testing, the items added are therefore those that contribute to providing more information in the vicinity of the latent continuum where this information is needed. Items that do not contribute to the desired characteristics can be removed from the test. The information function of a test can be examined after adding or removing specific items to measure the resulting effect.

Thus, tests can be constructed that provide maximum information at a precise point on the latent continuum, for example, when distinguishing between subjects who pass or fail an exam. Such a test is called a peaked test. To create such a test, items are selected whose difficulty indices are close to the cutoff score. Their discrimination index should be as high as possible, since information is proportional to the discrimination index, and in the case of a three-parameter model, the guessing parameter is chosen to be as small as possible; indeed, large values of this parameter tend to reduce the information.

Item response theory also makes it possible to concatenate two different tests, a process known as equationalization. Equationing is necessary when dealing with two test formats that were not explicitly designed to be parallel. It is then necessary to be able to place the corresponding scores on the same scale to compare subjects who have taken different tests.

There are essentially two methods of equation construction: Horizontal equation construction allows scores from tests of approximately the same level of difficulty to be placed on the same scale, for example, tests designed for the same grade level. Vertical equation construction is used for scores from tests with different levels of difficulty, for example, school tests taken by different age groups.

The problem arises from the fact that the true parameters of the items are unknown and that, in the parameter estimation process, the origin (the zero point) of the item difficulty scale is arbitrarily fixed (either as the average of the estimates of the subjects' abilities used to calibrate the items, or as the average difficulty of the sample of items used). This implies that scales obtained through separate estimations in two different populations are invariant only modulo an affine transformation. To link two scales, it is therefore necessary to be able to determine this affine transformation, which requires that the two estimation methods have something in common: either a certain number of items, a certain number of subjects who participated in the development of both scales, or a certain number of subjects who took a third test measuring the same latent trait and thus serving as a reference.

3. Creation of the financial literacy assessment tool

In this paragraph, we describe the main steps that led to the creation of our financial literacy assessment tool, which is a computer adaptive testing (CAT) tool, using the catR R package. The test items are taken from our financial literacy survey and the test is based on item response theory.

3.1 Creation of the item bank and calibration of the IRT model

We took the full data set with the answers to the financial literacy survey from work package 2 that was available in August 2025 and selected the 24 questions that required knowledge in sustainable finance to use as item bank for our assessment tool.

We based the calibration of the model on the answers of the 171 first participants of the financial literacy survey (ie all available data in late August 2025). These answers are coded into the Survey_ItemBank_for_R.xlsx file, which contains the following columns:

- Question_ID
- Section
- Item_Text
- Answer_Options
- Correct_Answer
- Notes

IRT parameter estimation

We computed the IRT parameters *a* (discrimination), *b* (difficulty) and *c* (guessing) of the items in the test by means of an IRT Python package and added the parameters into the file Survey_ItemBank_for_R.xlsx.

For the guessing parameter ***c***, we ensured that its value was aligned with the number of answer choices for each item. Specifically, we calculated ***c*** as the inverse of the number of possible options (e.g., for an item with 3 options, c=1/3=0.333). This reflects the probability of a respondent answering correctly by pure guessing.

3.2 CAT Simulation

First, we used R version 4.4.1 w with the package catR in order to simulate the answers of potential subjects with the aim of detecting potential problems.

The excel file Survey_Item_Bank.xlsx was imported into Rstudio and we added the IRT parameters.

Since the catR package works with a 4-parameter IRT model, we had to add a fourth parameter which in our case is the constant 1.

The catR package indeed expects the item bank to include four columns, even when using a 3-parameter logistic model. While our item parameters included only the discrimination (a), difficulty (b), and guessing (c) values, we needed to supply a fourth column to satisfy the package's input structure. The fourth column typically represents either: 1- the upper asymptote d in a 4-paratmeter logistic mode; (4PL) model, or 2- the scaling constant D used in the logistic function. In this simulation's case, we were not modeling a 4PL test, and we did not apply a custom logistic scale. Therefore, we added a column filled with the default value 1, which corresponds to the standard logistic scaling and implies no modification item characteristic curve.

We then successfully ran the simulation using the corrected structure as shown in *figure 1* in the appendix. The final code used is shown below:

```
> library(catR)
> library(readxl)
> Survey_ItemBank_for_R <- read_excel("Desktop/All Files/Courses/Student Job/Assignment 1
/Survey_ItemBank_for_R.xlsx")
> View(Survey_ItemBank_for_R)
> a <- Survey_ItemBank_for_R$a
> b <- Survey_ItemBank_for_R$b
> c <- Survey_ItemBank_for_R$c
> itemBank <- cbind(a, b, c, rep(1, length(a)))
> theta_value <- 0.5
> responsePattern <- randomCAT(

+     itemBank = itemBank,
+     trueTheta = theta_value,
+     start = list(theta = theta_value),
+     test = list(method = "ML"),
+     stop = list(rule = "length", thr = 10),
+     final = list(method = "ML")
+ )
```

This successfully generated a CAT simulation result using 10 question items, where:

- *library(catR) & library(readxl):* loads necessary packages.
- *Survey_ItemBank_for_R <- read_excel("Desktop/All Files/Courses/Student Job/Assignment 1 /Survey_ItemBank_for_R.xlsx"):* loads item bank from excel.
- *View(Survey_ItemBank_for_R):* views the dataset.
- *a <- Survey_ItemBank_for_R$a & b <- Survey_ItemBank_for_R$b & c <- Survey_ItemBank_for_R$c:* extracts item parameters
- *itemBank <- cbind(a, b, c, rep(1, length(a))):* creates the item bank with a, b, c, and D (=1).
- *theta_value <- 0.5:* sets initial theta.

- *itemBank = itemBank*: uses the matrix we created with a, b, c, and D values.
- *trueTheta = theta_value:* sets the user's true ability which is used to simulate their responses.
- *start = list(theta = theta_value):* sets the initial guess for ability when the test begins.
- *test = list(method = "ML"):* uses Maximum likelihood (ML) as the method to estimate ability after each response.
- *stop = list(rule = "length", thr = 10):* stops the test after 10 items, therefore uses a fixed-length stopping rule of 10 items. A 10-question test is sufficient to demonstrate the core functionality of the adaptive testing algorithm, especially within the scope of a 27 question-survey, without making the simulation overly long or complex.
- *final = list(method = "ML"):* after the test ends, it re-estimates the final ability using ML based on all responses.

3.3 Finalization of the interactive assessment tool

Methodological Enhancements and Rationale

In the initial phase we validated the pipeline by generating a simulated response pattern with randomCAT(). Building on that we then created the interactive assessment tool that presents questions to a real test-taker, records the answers, gives a final score for the literacy of the candidate and saves all answers plus the score into a data file.

Explanation of the code

1. Package checks and imports

```
if (!requireNamespace("catR", quietly = TRUE)) install.packages("catR")
if (!requireNamespace("readxl", quietly = TRUE)) install.packages("readxl")
library(catR)
library(readxl)
```

- *requireNamespace()* checks whether a package is installed (base R). if missing, *install.packages()* installs it.
- *library(catR)* exposes *nextItem()* and *thetaEst()* we used conceptually; *library(readxl)* exposes *read_excel()* for the workbook. These are the same packages used in the earlier simulation.

2. Load item bank and build the 4-column catR matrix

```
excel_path <- "Survey_ItemBank_for_R.xlsx"
ib <- read_excel(excel_path)

# Expected columns in Excel:
# Question_ID | Item_Text | Answe3r_Options | Correct_Answer | a | b | c

# Build catR item parameter matrix a, b, c, D
a <- as.numeric(ib$a)
b <- as.numeric(ib$b)
c <- as.numeric(ib$c)
itemBank <- cbind(a, b, c, rep(1, length(a)))   # D = 1
```

- *Read_excel()* loads the item bank prepared earlier.
- *as.numeric()* ensures parameters are numeric vectors.
- *cbind(a, b, c, rep(1, length(a)))* constructs the 4-column matrix catR expects; the fourth column *D=1* is the fix documented previously in section 1 of the report (standard logistic scaling, not 4PL).
- *%in%* is a base R operator (set membership) used later to check whether a column exists.

3. Session configuration and state tracking

```
ax_items <- 10L          # fixed-length assessment (can be changed)
theta_current <- 0       # initial theta guess
administered <- integer(0) # indices of items asked
```

```
responses   <- integer(0)  # 1 correct, 0 incorrect
# Keep track of only scored (objective) items for theta estimation
scored_admin <- integer(0)  # indices of scored items
scored_resp  <- integer(0)  # responses (1/0) for scored items
log_rows <- list()
```

- *10L* is an integer literal; fixed length echoes the earlier demonstration and keeps runs short and consistent.
- We maintain two histories: all items administered versus scored items only. This ensures theta $\hat{\theta}$ is not updated by opinion/likert items (those have *NA* for *Correct_Answer*)

4. Opening console banner

```
cat("\n===== Adaptive Assessment (3PL; ML; length =", max_items, ") =====\n")
cat("Answer by typing the OPTION NUMBER (1, 2, 3, ...) and press Enter.\n")
```

- *cat()* writes plain text to the console so the test-taker knows exactly what to do.

5. UI for a single item (input validation and diplay)

```
ask_item <- function(i_row) {
  qid   <- if ("Question_ID" %in% names(ib)) ib$Question_ID[i_row] else i_row
  qtext <- ib$Item_Text[i_row]
  opts  <- ib$Answer_Options[i_row]

  cat("\n------------------------------------------------------------\n")
  cat(sprintf("Question %s\n", as.character(qid)))
  cat(paste0(qtext, "\n"))
  cat("Options:\n")
  cat(paste0(opts, "\n"))  # e.g., "1 = ..., 2 = ..., 3 = ..."

  repeat {
    ans <- readline(prompt = "Your answer (option number): ")
```

```
  if (nzchar(ans) && grepl("^[0-9]+$", ans)) return(as.integer(ans))
  cat("Please enter a number like 1, 2, 3 ...\n")
 }
}
```

- *%in%* (baseR) checks if *"QuestionID"* is a column *ib*; if yes we print it, otherwise we fall back to the row index.
- *sprintf()* formats the question header; paste0 links strings without spaces.
- *repeat {...}* implements an input loop that only exists when the user provides a numeric option; readline() captures typed input; *nzchar()* ensures it is not empty, *grepl("^[0-9]+$", ans)* confirms it is digits only, finally, *as.integer()* converts to an integer option.
- No backslashes are needed in code here, the only *\n* sequences are standard newline characters inside strings.

6. Adaptive loop; selection, scoring, estimation, and logging

```
while (length(administered) < max_items) {
 # select next item (MFI)
 sel <- nextItem(itemBank, theta = theta_current, out = administered, criterion = "MFI")
 i <- sel$item

 # ask user
 choice <- ask_item(i)

 # correct option (must be the option number in Excel)
 correct_option <- suppressWarnings(as.integer(ib$Correct_Answer[i]))

 # If NA (opinion/Likert item), do not score it
 if (is.na(correct_option)) {
  is_correct <- NA_integer_
 } else {
  is_correct <- as.integer(identical(choice, correct_option))
```

```
  # add to scored lists for theta estimation
  scored_admin <- c(scored_admin, i)
  scored_resp  <- c(scored_resp,  is_correct)
 }

 # record in main log (keep NA for opinion items)
 administered <- c(administered, i)
 responses    <- c(responses, ifelse(is.na(is_correct), NA_integer_, is_correct))

 # re-estimate theta ONLY from scored items
 if (length(scored_resp) > 0) {
  theta_current <- thetaEst(
   it     = itemBank[scored_admin, , drop = FALSE],
   x      = scored_resp,
   method = "ML",
   D      = 1
  )
 }

 # log
 log_rows[[length(administered)]] <- data.frame(
  step = length(administered),
  question_index = i,
  question_id = if ("Question_ID" %in% names(ib)) as.character(ib$Question_ID[i])
else as.character(i),
  user_choice = choice,
  correct_option = correct_option,
  correct = is_correct,
  scored = !is.na(is_correct),
  theta_hat = as.numeric(theta_current)
 )
```

```
 cat(sprintf(
   "→ Saved. Correct option = %s | Your answer = %s | Θ̂ (ML) = %.3f\n",
   ifelse(is.na(correct_option), "NA", as.character(correct_option)),
   as.character(choice),
   as.numeric(theta_current)
 ))

 # If opinion-based (no correct answer), show a clear note
 if (is.na(correct_option)) {
   cat("→ Opinion-based item: excluded from scoring.\n")
 }

}
```

- Item selection: *nextItem(..., criterion = "MFI)* chooses the most informative item at the current theta θ̂ while excluding *out = administered* .
- Scoring: If *Correct_Answer* is missing (*NA*), the item is opinion-based which is not scored. Otherwise, we compute correctness. We use *identical(choice, correct_option)* to avoid accidental type pressure; this returns a strict TRUE/FALSE which we turn into 1/0 with *as.integer()*.
- Book-keeping: we maintain both *administered* (all items) and *responses* (1/0/NA when opinion-based).
- Estimation*: thetaEst()* runs ML on scored items only. The subsetting *uses drop = FALSE* to ensure the result remains a matric even if there is only one scored item. *D=1* keeps the same scale as our item bank.
- Logging: we push a *data.frame* row to *log_rows* at every step, including *theta_hat*.
- Console feedback: a concise line confirms the correct option, the user's choicem and the updated θ̂.
- Symbols used*: ifelse(), as.character(), as.numeric(), is.na()* are all base R.

7. Test end and export of results

```
results <- do.call(rbind, log_rows)
write.csv(results, "assessment_log.csv", row.names = FALSE)


cat("\n===== Assessment finished =====\n")
cat(sprintf("Items administered: %d\n", nrow(results)))
cat(sprintf("Final ability estimate (Θ̂, ML, D=1): %.3f\n",
results$theta_hat[nrow(results)]))
cat(sprintf("Total correct (scored): %d / %d\n",
        sum(results$correct, na.rm = TRUE),
        sum(!is.na(results$correct))))
cat('Log saved as "assessment_log.csv" in this folder.\n')
```

- *do.call(rbind, ...)* stacks all logged rows into data frame.
- *write.csv()* creates an audit trail you can inspect or plot later (e.g., $\hat{\theta}$ trajectory).
- We also summarize the run in the console: number of items, final $\hat{\theta}$, and total correct on scored items.

4. Conclusion

The aim of this workpage was the creation of a financial literacy assessment tool. The tool is available on the webpage of our project and can be freely used by everybody having R and R Studio installed on the computer.

This is however just a first step. Indeed, having just 27 test items limits the precision of the tool in most areas of the literacy ability parameters.

But the choice of an IRT model-based CAT allows for future projects to build on this one to develop a larger database of items. As mentioned previously, one of the major advantages of the IRT is that parameter estimation is independent of the population from which the estimation is derived and subjects can be tested with different items, appropriate to their skill level, while maintaining the comparability of their scores.

It is hence possible to add items by doing future surveys with other subjects, containing just a few of our items plus many others, then estimate the IRT parameters of all these new items and add them to our item pool.

This would create over time a big item database that can be used to improve the precision of the assessment test over the whole continuum of the possible item parameters.

5. References

A. Anastasi. 1982. Psychological Testing. New York: Mac Millan. 6th edition.

Assessment Systems Corporation. 1995. User‘s manual for the MicroCAT testing systems. St. Paul, MI.

H.-H. Chang & Z. Ying. 1996. A Global Information Approach to computerized Adaptive Testing. Applied Psychological Measurement, 20-3, 213-229.

S.E. Embretson & S.P. Reise. 2000. Item Response Theory for Psychologists. Mahwah NJ: Erlbaum.

G. Fischer. 1974. Einführung in die Theorie psychologischer Tests. Bern: Hans Huber.

M.F. Lord. 1980. Robbins-Monro procedures for tailored testing. Educational and Psychological Measurement, 31, 3-31.

D. Magis, G. Raiche & J. Barrada. 2022. Generation of IRT Response Patterns under Computerized Adaptive Testing, 1-120.

D. Magis & J. Barrada. 2017. Computerized Adaptive Testing with R: Recent Updates of the Package catR, 1-19.

G. Rasch. 1960. Probabilistic models for some intelligence and attainment tests. Copenhagen: The Danish Institute for Educational Research.

J. Rost. 1995. Testtheorie, Testkonstruktion. Bern: Hans Huber.

W.J. van der Linden & C.A.W. Glas. 2000. Computerized Adaptive Testing: theory and Practice. Dordrecht: Kluwer.

H. Wainer, N. Dorans, D. Eignor, R. Flaugher, B. Green, R. Mislevy, L. Steinberg & D. Thissen. 2000. Computerized adaptive testing: A primer (2nd. ed.). Mahwah NJ: Erlbaum.

D.J. Weiss. 1982. Improving measurement quality and efficiency with adaptive testing. Applied Psychological Measurement, 6, 473-492.

6. Appendix

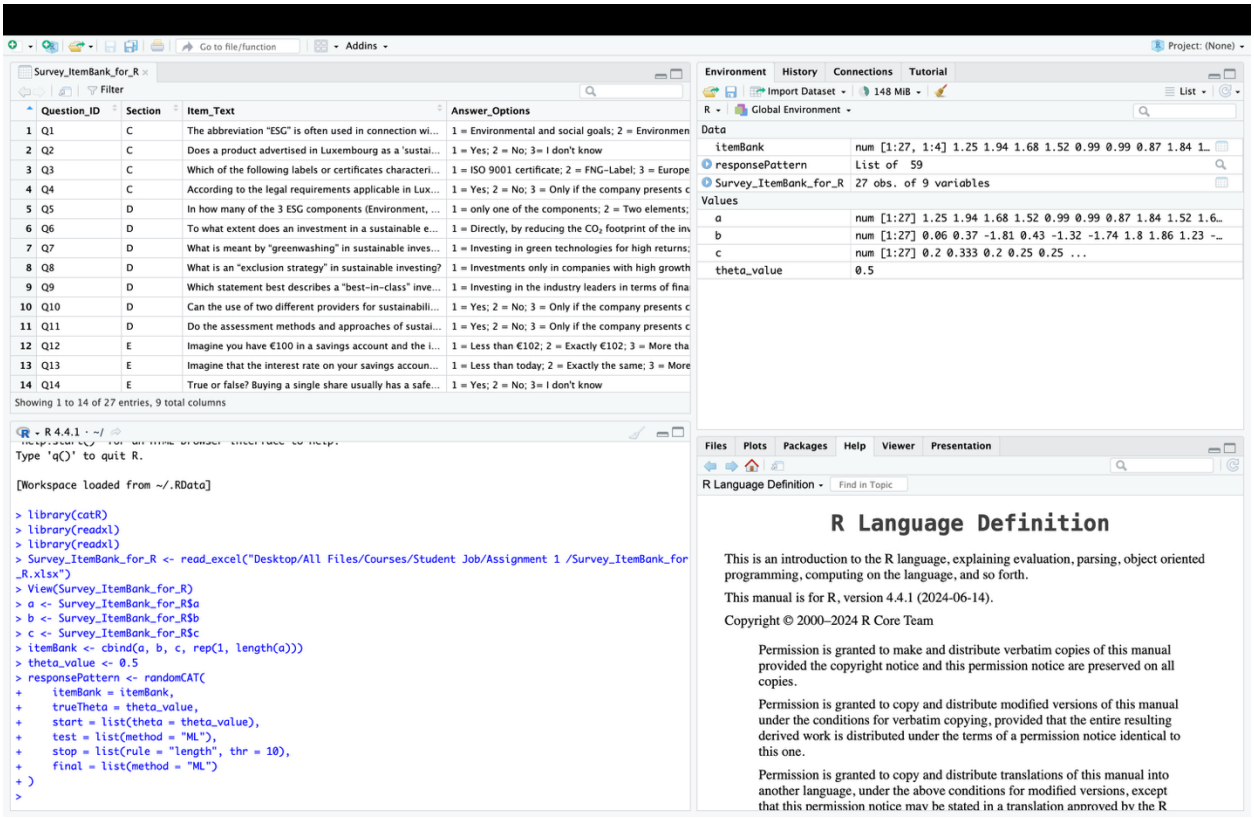

Figure 1: Simulation Output

This image shows the full cat simulation code and successful execution using the randomCAT() function. The item bank (a b, c, and fourth column for D parameter) was properly loaded, and the response pattern was generated based on a true theta value of 0.5.

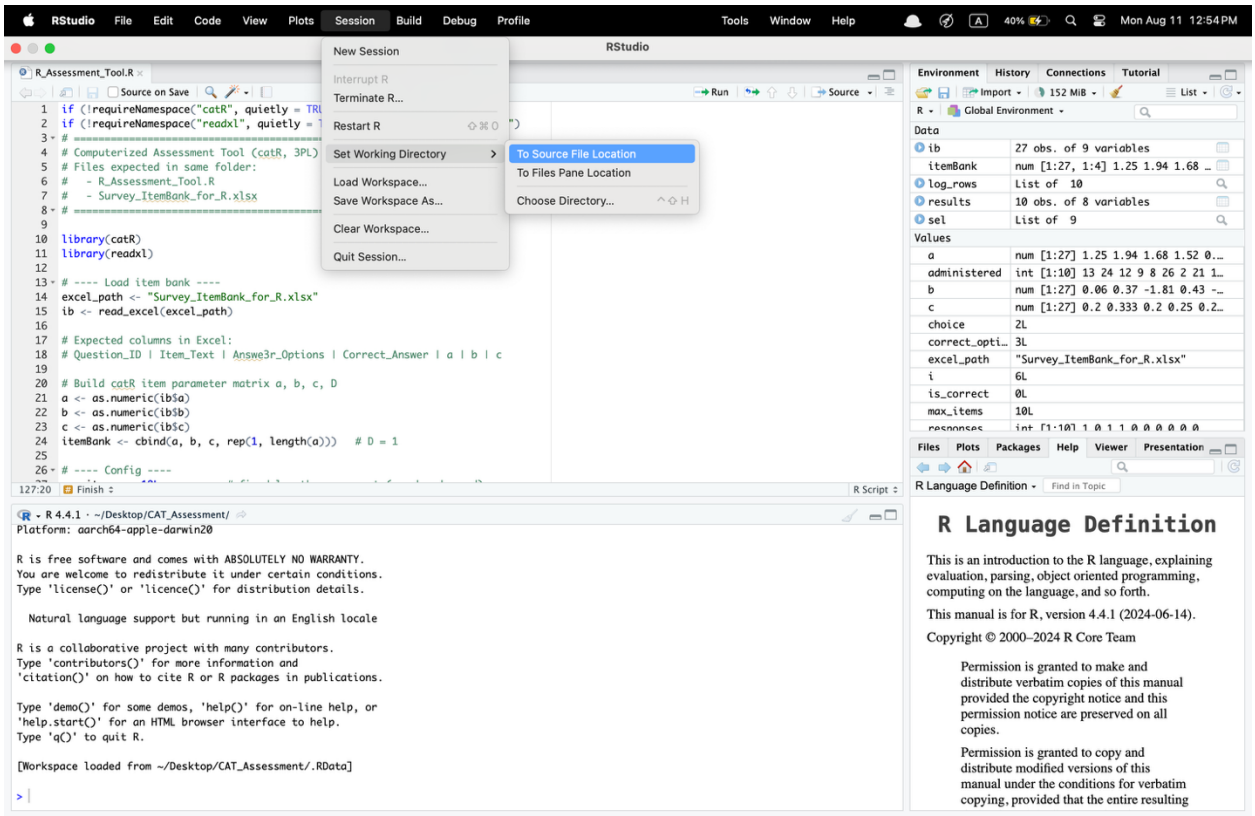

Figure 2: Source Step (load the tool)

a) Session > Set working directory > To source file location

Sets R's working directory to the folder where the currently open script (R_Assessment_Tool.R) is saved. Thereafter, any relative file paths such as Survey_ItemBank_for_R.xlsx are resolved in that folder. Please note that if the working directory points elsewhere, R will not find the file. Save the script first and use at the start of each run.

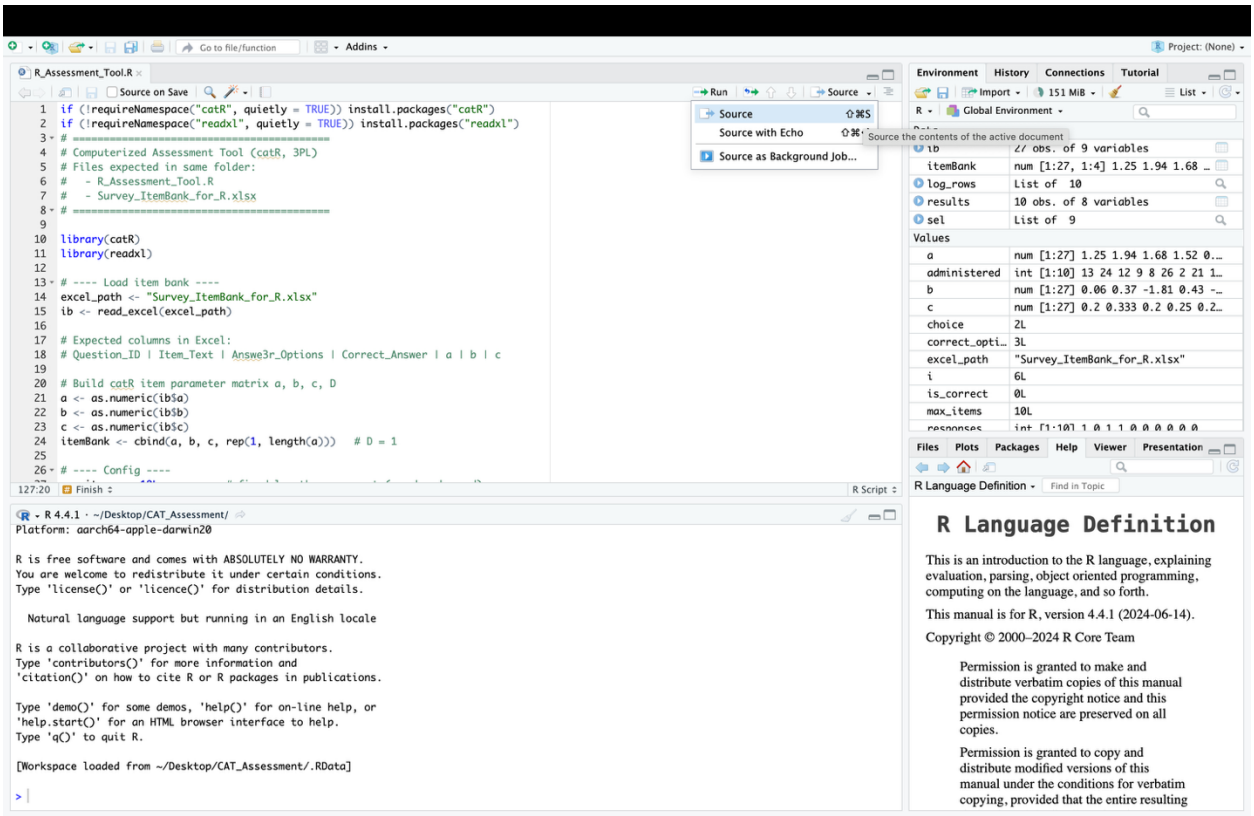

Figure 3: Source Step (load the tool)

b) Source button (top right)

Executes the entire open script in the current R session (loading packages, reading the excel item bank, building the catR matrix, and starting the interactive loop). Sourcing initializes the assessment and immediately begins the question-answer sequence in the console. Please note that it does not change the user's working directory. If the directory is wrong, sourcing will fail to read the excel file.

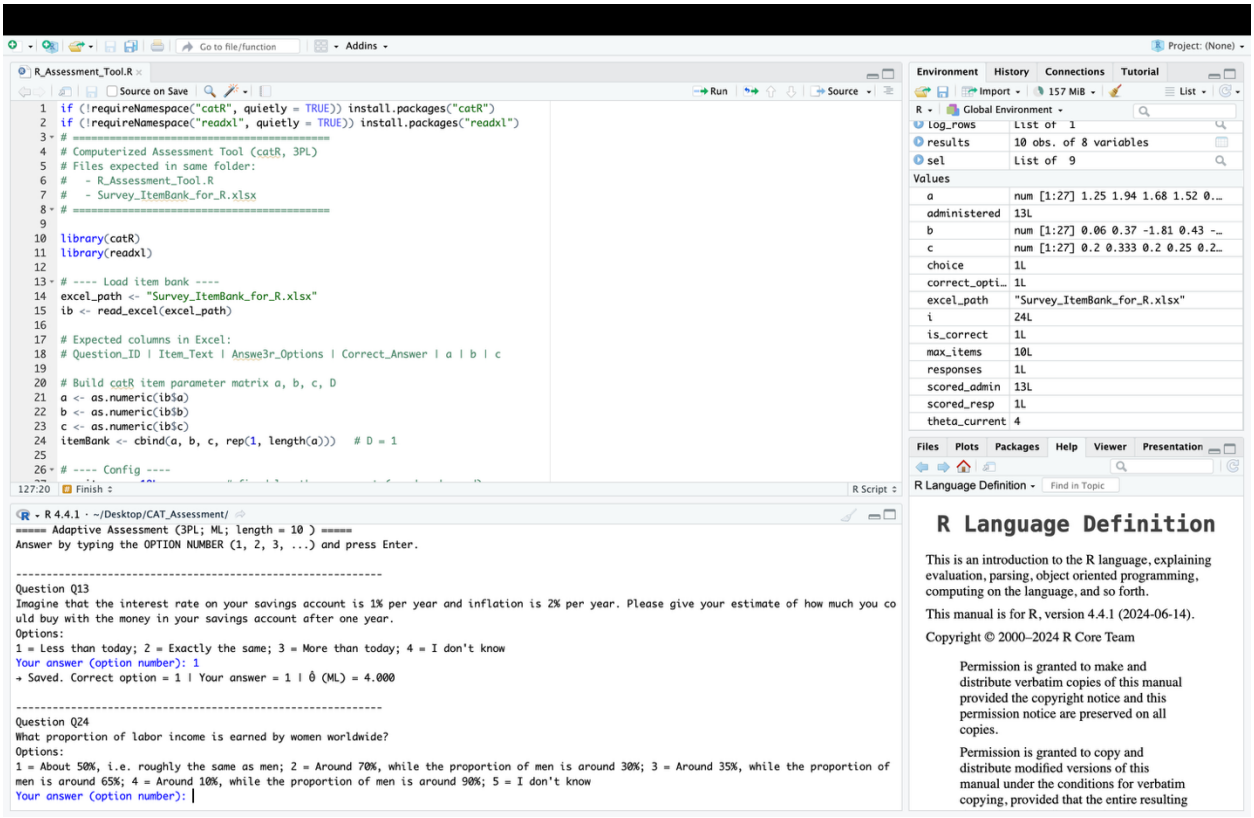

Figure 4: Session Step (take the assessment)

After setting the working directory and sourcing, the user can start taking the test by typing the number of the answer they believe is correct to the multiple-choice question asked and press enter to start the new question.

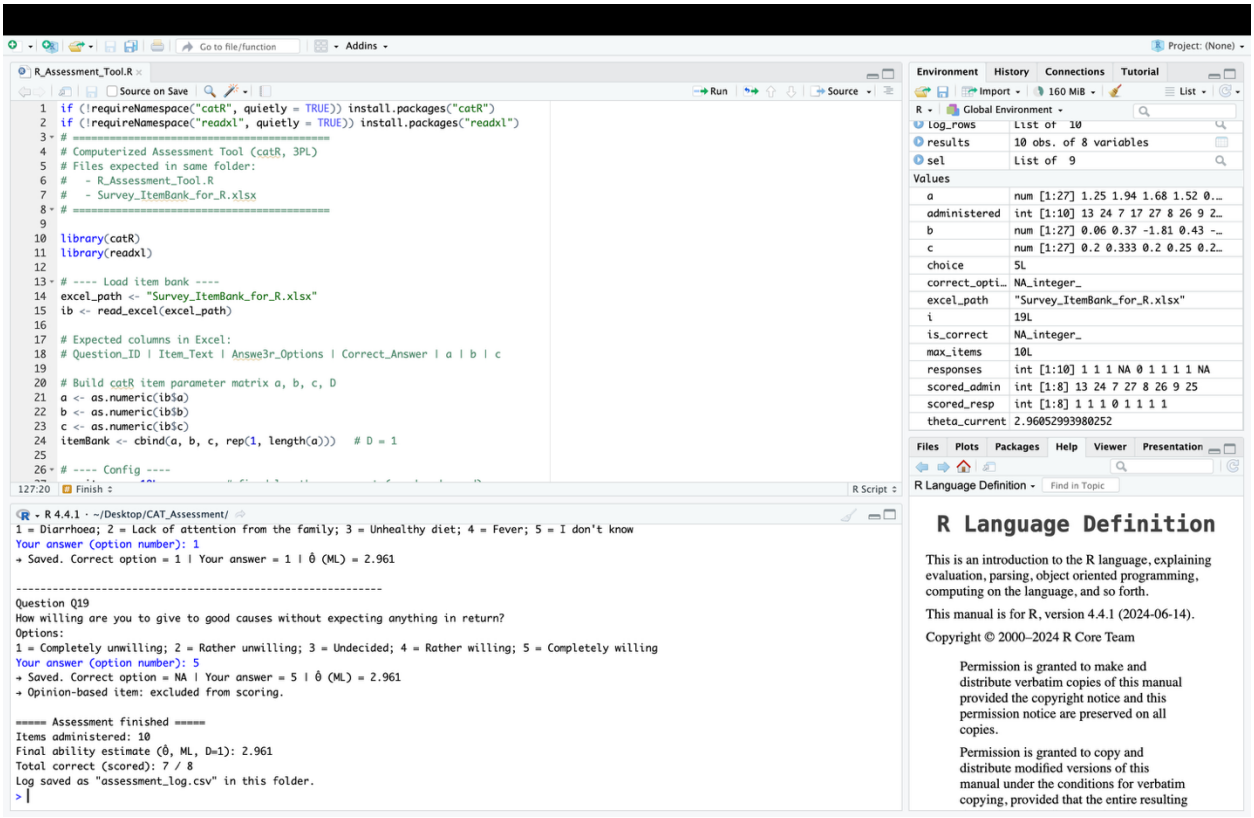

Figure 5: Assessment Output

After completion of the fixed-length assessment (10 questions), the script prints a run summary in the console which are: the total number of items administered, the final ability estimate computed by Maximum Likelihood Method with D=1, and the total correct answers scored (opinion/Likert are excluded by design).