Pilot or Passenger: Test-Flying an LLM Workflow for the SIB Listening Item Analysis

BILC Testing Workshop, 2026

Marju Laurits
Estonian Military Academy

1. Context

Provide a structured overview of the results of the statistical analysis of the SIB listening items to guide decisions made about anchor items for listening.
The decisions regarding the anchor items were based on the Classical Test Theory (CTT) and Item Response Theory (IRT) results and the Neo-Classical approach of dividing test takers into two groups: Masters and Non-Masters.

1.1 Existing Workflow

Multiple programs (jamovi, jMetrik, TiaPlus, dexter) were used to calculate the statistics. Results across all statistical platforms were synthesised into a single comprehensive spreadsheet.
While these programs offer many advantages, they provide limited options for customizing visual outputs and collating results is time consuming.

- Jamovi - descriptive statistics, CTT, Rasch, by group analysis, variety of plots. Customisation of graphs limited (colour schemes, font). (The jamovi project (2026). jamovi (Version 2.7.31) [Computer Software]. Retrieved from https://www.jamovi.org.
- jMetrik - descriptive statistics, CTT, Rasch, by group analysis, limited plots. The graphs lack visual appeal and are not customisable. To re-run the analysis for a single test version with more data, the user needs to know how to set up and run the code (Meyer, J. P. (2018). jMetrik (Version 4.1.1) [Computer software]. Psychomeasurement Systems. Retrieved from https://itemanalysis.com/jmetrik-download/.
- TiaPlus - descriptive statistics, CTT, by group analysis, basic plots. An older program that may become incompatible with future operating systems. (CITO (1998-2013). TiaPlus).
- Dexter - descriptive statistics, CTT, Rasch, by group analysis for descriptive statistics, range of plots. Designed for RStudio, though GUI exists (this has to be launched from RStudio). Novice users can expect a challenging learning curve; at the same time, quite versatile and can do a variety of analyses.

1.2 Why Look for an Alternative?

It is time-consuming to set up the analysis across the different platforms.
The programs cannot be used to produce graphs that show the comparison in the performance of Masters vs Non-Masters, which is vital in the STANAG 6001 testing context.
Customisation of output is low.

1.3 Possible Solution?

Explore using Large Language Models (LLMs) as coding agents to generate an R script that will produce desired output.
LLMs used:
Gemini (mainly, subscription tier) and Claude (free tier).

2. Test-Flying LLM(s)

2.1 Core Questions

Q1. *To what extent can LLMs be used to create a single comprehensive analysis including a range of plots in RStudio?*

Q2. *To what extent can the script be re-used on new tests?*

Q3. *What can one user's experience reveal about using LLMs to generate R scripts to analyse data for a specific context?*

2.2 User Prerequisites

Familiarity and experience in using statistical software, including RStudio. Familiarity and experience in using LLMs. (Posit team (2026). RStudio: Integrated Development Environment for R. Posit Software, PBC, Boston, MA. URL http://www.posit.co/.

2.3 R Script Generation

2.3.1 Prepare the data before prompting

1. Data validation.
2. Confirm the dataset has unique values for test taker id numbers, item labels across all data sets included in the analysis.
3. Adopt clear naming conventions for files, variables.
4. Re-name variables where needed as dexter expects some specific labels.
5. List all the files used in the analysis.

2.3.2 Process description

1. Specify the problem, list the analyses to be carried out.
2. Identify the platform that meets the needs (in this case RStudio).
3. Find reputable experts in the field who have used RStudio, identify the packages they used and tell the LLM to use those packages.
4. Start with basic analysis, then refine to get the desired output.
5. It may be useful to test the code block with one item before looping in remaining items.
6. Loop the relevant parts of the code in the chat with the error message when a section fails to execute.
7. Ask the LLM to explain parts of the code to make sure the correct variables are used in the calculations.
8. Establish a peer-revision system of the results to double check whether the output matches the results from a reliable software (in this case, TiaPlus/Dexter GUI results, jMetrik, jamovi).
9. Build the script one type of analysis at a time.
10. Include explanations and comments.
11. Automate the script where possible.

3. Lessons Learnt

3.1 Workflow Challenges

1. A basic understanding of RStudio is needed to navigate the interface and troubleshoot errors.
2. Subscription tiers have an impact on the efficiency and time spent on the process.
3. As the chat history grows, the LLM's responses can degrade, occasionally "forgetting" nuances or specific analyses established earlier in the session.
4. The code generated by LLM may mishandle missing data without flagging it; the output needs to be peer-reviewed.
5. High-volume iterative coding can exhaust message caps unexpectedly quickly, even on paid subscription tiers.
6. It is difficult for a novice user to judge if the generated script is computationally efficient.

3.2 Suggested Best Practices

1. Expect to refine, run, compare the output against known values, and prompt again until outputs and visuals meet expectations.
2. Build in sanity checks into the script to ensure the data is reading correctly.
3. When resuming coding across different time slots, verify that the LLM has not altered the variable labels.
4. Ask the LLM to explain the "what and why" of its code. If time allows, regenerate the final script in a fresh chat to solidify the user's own understanding.
5. Test the prompts in a different LLM to compare coding approaches and find the cleanest solution.
6. Ask the LLM which parts of the script can be automated.

3.3 Benefits of Using an R Script Instead of Multiple Platforms

1. Replicability of the results, adds to transparency.
2. The code can be adapted for future needs.
3. Including comments helps to record or remind the user of specific nuances.
4. Automated file name generator means there is no need to go through the whole code and change all the output file labels in case of a new test version where the analysis remains the same.
5. In case of visuals, gives more control over what information is included and how it is presented (density lines for histograms, item data from different analysis included in one output plot).
6. Output is collated and set up according to user specifications (information, colour scheme).

4. Results and Reflections

Since LLMs became widely available, the EMA testing team has integrated them into their test development process. However, the test data analysis workflow has remained the same, as platforms like RStudio require a steep learning curve for non-coders. Given that LLMs excel at writing code, this initiative investigated the feasibility of using them for that purpose.

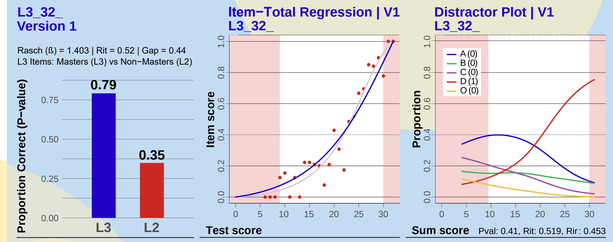
4.1 Fasten Your Seatbelts: Clear Skies and Turbulence Ahead

Q1. *To what extent can LLMs be used to create a single comprehensive analysis including a range of plots in RStudio?*

LLMs can work well in this context, but the script generation takes time for novice users. Users should have a good understanding of statistical concepts (so that they know what analysis to ask for and how to check the output for accuracy) and a basic understanding of RStudio and coding is a must. There were misunderstandings regarding the intent of the user and the LLM forgot relevant features that needed to be included. The user has to be vigilant.

However, the single analysis script produced:

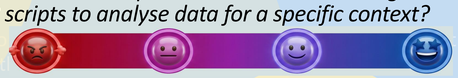
1. Excel file that contains whole group analysis, sub-groups analysis, distractor analysis, Rasch estimates (ENORM and JMLE from TAM), test taker theta values, test taker scores by task and their total score.
2. Pdf files: Masters vs Non-Masters comparison (incl item beta value, rit, the gap between Ms vs N-Ms); ENORM Fit Plots, ENORM Item Difficulties (as a single plot from easiest to most difficult); Item Regression and Distractor Plots (side-by-side); Score Distributions (for whole population and sub-groups); Wright map for all items.



Q2. *To what extent can the script be re-used for other tests?*

It was used successfully to re-analyse the SIB Reading items and then also, with some minor modifications, to analyse a single data set (SIB Listening Version 1). The script for two listening test versions required just five substitutions (input file names and the project name) before it could be used with the SIB reading test data sets. Script validation would be useful.

Q3. *What can one user's experience reveal about using LLMs to generate R scripts to analyse data for a specific context?*



What is the Verdict: Pilot or Passenger?

Novice users act as pilots navigating toward a set destination while they also have to know when to ask for course corrections and route explanations. Yet, they are also passengers, since they must rely on the LLM's suggestions and explanations. Despite this tension, this workflow can unlock opportunities that would otherwise remain out of reach for language testers with limited coding experience.