Journal of Data Science 24 (1), 276–281
January 2026

DOI: 10.6339/26-JDS1208H
Education in Data Science

Discussion of "Addressing the Challenges of AI-Generated Assignment Submissions in Education: Insights and Strategies"*

Aurélien Nicosia*

[1]*Département de mathématiques et de statistique, Université Laval, Québec, QC, Canada*

Keywords *AI-assisted assessment; contextualized assignments; large language models; R programming; reproducible workflows; statistics education*

Wang, Xu, Liu and Zhai (2025) present a timely analysis of how generative AI challenges traditional assessment. They argue that detection-based approaches are neither reliable nor pedagogically aligned, and instead recommend strategies that emphasize process, critical thinking, and responsible human–AI collaboration. Their proposed responses include oral examinations, process documentation, contextualized tasks, explicit AI literacy, and a warning against "counter magic," where both students and instructors outsource too much to AI.

In this discussion, I present an assignment from an advanced R course at Université Laval in which the use of large language models (LLMs) is not only permitted but required and structured. The pedagogical goal is to integrate LLMs into statistical workflows in a transparent and reproducible way and to evaluate students on the quality of that integration. To enable fair, criterion-referenced grading across submissions, the assignment specifies a minimal API (required function names, interfaces, and expected outputs), while leaving substantial design freedom inside the implementation. Grading combines a rubric with a lightweight automated "conformity check" that verifies that each submission installs and runs its required demonstrations in a clean environment.

A Contextual Statistical Assignment with Mandatory LLM Use

The assignment ("Contextual statistical analysis with an LLM") asks students to build a small R package that provides contextual versions of familiar statistical tools. Concretely, students must implement *at least two* "contextual" functions chosen from common workflows (e.g., `lm`, `t.test`, `table`, or a `ggplot`-based visual), together with S3 classes and methods that generate audience-aware interpretations. For example, a student may implement `lm_context()` and `t_test_context()` as wrappers around `lm()` and `t.test()`.

What the Contextual Objects Contain Each contextual wrapper returns an S3 object (e.g., class `"context_lm"` or `"context_ttest"`) that stores:

- the original statistical object (e.g., `lm` or `htest`),
- a context field (dataset documentation and/or user-supplied text),
- metadata required for transparency and reproducibility (call, formula, and LLM options such as provider/model).

☆Main article: https://doi.org/10.6339/25-JDS1208.
* Email: Aurelien.Nicosia@mat.ulaval.ca.

Preset Function Names and Interfaces A frequent challenge in assignments involving software design is that students can spend substantial time varying interfaces, which complicates grading and shifts attention away from the intended learning outcomes. For this reason, the handout provides a minimal set of required function names and interfaces (argument structure and expected return types). Students must implement the internal logic and S3 methods corresponding to this shared API, but may extend beyond it (e.g., additional contextual wrappers, extra metadata, or alternative prompt strategies).

Required S3 Methods and Outputs Students must implement at least three S3 methods across their contextual classes. The `print()` method is required for *each* contextual function and must provide a short numerical summary plus one or two sentences of interpretation. In addition, students implement `summary()` (more detailed explanation in accessible language) and `plot()` (a diagnostic or explanatory graphic whose title/subtitle/labels are adapted to the stored context), depending on their chosen contextual functions. A key requirement is that LLM-generated text is clearly separated from the numerical output and is explicitly framed as an interpretation grounded in that output.

LLM Integration and Robustness Requirements A crucial requirement is that interpretive text (and, optionally, plot annotations) is generated with the help of an LLM through a dedicated function `ctx_llm_generate()`. Students construct prompts by combining a controlled summary of numerical results with contextual information, then send the prompt to an LLM via a local provider (e.g., Ollama) or a hosted API. The assignment explicitly asks students to (i) include an anti-hallucination instruction (e.g., "do not invent values that are not in the numerical summary"), (ii) limit prompt length by extracting only relevant parts of the dataset documentation (e.g., Description/Format), and (iii) implement safe failure modes (clear fallback messages when the LLM is unavailable, rather than crashing). Robust handling of LLM unavailability is important for reproducibility and for practical grading: evaluators should be able to run submissions without depending on a particular external service being reachable at that moment.

While students may implement direct HTTP calls (e.g., via `httr+jsonlite`), they may also use the package `ellmer` (Wickham et al., 2025), which provides a unified interface to local and hosted LLM providers.

An Illustrative Example of Expected Output

To clarify expectations, students receive an illustrative example of the output of a `summary()` method applied to a contextual linear model.

```
mod <- lm_context(body_mass_g ~ bill_length_mm,
                  data    = penguins)
summary(mod)
```

A possible structure for the printed result is:

```
=== LM output ===
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)      -1500.2     250.3   -5.99  3.1e-08 ***
bill_length_mm      98.7       5.2   18.95  < 2e-16 ***

Residual standard error: 410 on 340 df
Multiple R-squared: 0.514, Adjusted R-squared: 0.512

=== Contextual interpretation (LLM) ===
In this sample of penguins, the model suggests a strong positive
linear association between bill length and body mass. According to
the dataset documentation, species with longer bills also tend to be
larger birds. The positive slope indicates that, on average, each
additional millimetre of bill length is associated with an increase
in body mass, and the very small p-value indicates that this pattern
is unlikely to be explained by random variation alone in this study.
```

The exact wording is left to the LLM and to the student, but the pattern is fixed: a conventional numerical summary, followed by a short context-aware explanation that explicitly connects the statistics to the documented context of the data. (The numerical values shown above are illustrative and are provided only to make expectations concrete.)

Assessment, Grading, and Planned Student Feedback

Student-Selected Methods and Extensions A review of the submitted student packages suggests that many teams went beyond the minimum requirement of two contextual wrappers and applied the same design pattern to a broad range of statistical tasks. Implementations frequently targeted classical inferential procedures (e.g., two-sample t-tests, ANOVA-style comparisons, tests for proportions, and chi-squared tests), and some groups extended the approach to multivariate, predictive, and time-series methods (e.g., principal component analysis, k-nearest neighbors classification, and ARIMA-style modeling). Importantly, students typically did not reimplement these algorithms from scratch; rather, they wrapped existing R functions and focused on the intended learning outcomes of the assignment: selecting which results to surface, constructing prompts grounded in those results and in dataset context, and producing audience-aware interpretations through an explicit LLM layer. This diversity of method choices provides initial evidence that the "contextual" wrapper approach can generalize across multiple families of statistical tools while keeping communication and responsible interpretation at the center of the workflow.

How the Assignment Is Graded The assignment is graded with a criterion-referenced rubric that evaluates statistical correctness and software quality while treating LLM integration as a first-class outcome. Table 1 summarizes the rubric used for grading (100 points).

This rubric makes the assessment targets explicit: students are not evaluated on whether they used AI "secretly" or "too much," but on whether they used it in a disciplined, transparent way that supports statistical reasoning, communication, and reproducibility.

Academic Integrity, Transparency, and Privacy Guidance To align with Wang et al.'s emphasis on responsible human–AI collaboration, the handout includes explicit guidance that is evaluated indirectly through the rubric:

Table 1: Rubric for "Contextual statistical analysis with an LLM" (100 points).

Criterion	**What is assessed (examples)**	**Pts**
Core functionality	Two contextual functions implemented and operational; correct wrapping of base functions; S3 objects store results, context, and metadata.	20
S3 methods	Correct S3 dispatch; `print()` for each contextual function; additional `summary()` and/or `plot()` methods produce useful, readable outputs.	20
LLM integration	`ctx_llm_generate()` implemented; prompts grounded in results + context; anti-hallucination instructions; provider/model parameters explicit; robust fallbacks when LLM is unavailable.	20
Contextual visualization	Plots are appropriate and honest; titles/subtitles/labels are context-aware; optional LLM-generated annotations remain faithful to numerical results.	15
Software quality & reproducibility	Package structure (DESCRIPTION/NAMESPACE); dependencies declared; no hard-coded API keys; informative error messages; metadata recorded to support transparency.	15
Documentation & demo	README explains installation and LLM prerequisites; reproducible examples; `examples/demo.R` demonstrates use on at least two datasets.	10
Total		**100**

- **Disclosure:** the README and/or object metadata should make clear *where* LLM calls occur and which provider/model were used.
- **Verification:** LLM outputs are treated as drafts; students are expected to prevent hallucinated numbers or claims by grounding prompts in the computed results and by including explicit "do not invent values" instructions.
- **Data minimization:** prompts should include only what is needed (e.g., a numerical summary and short dataset documentation snippets), not raw data tables.
- **Sensitive data:** students are instructed to avoid sending personally identifiable or confidential information to hosted APIs; local inference is encouraged when privacy is a concern.
- **Reproducibility:** provider/model choices should be recorded so that differences across runs can be interpreted, and failures should be handled transparently rather than hidden.

Student Feedback After submission, we invited students to complete an anonymous post-assignment questionnaire (in French, matching the course language). Twelve students responded. Table 2 summarizes the six Likert items (1 = strongly disagree to 5 = strongly agree). Overall perceptions were strongly positive regarding global learning (M = 4.83) and the contribution of S3 object design to understanding object-oriented programming in R (M = 4.75). Perceived relevance of LLM-generated explanations was also positive on average (M = 4.08).

Open-ended responses (5–8 respondents per question) highlighted two recurring themes. First, several students reported that embedding a contextual interpretation directly next to

Table 2: Student feedback on the assignment (Likert 1–5; $n = 12$).

Item	M	SD	% (4–5)
Instructions were clear.	4.08	1.24	66.7
LLM use helped me understand the statistical function better.	3.33	1.30	50.0
Creating S3 objects helped me understand OOP in R.	4.75	0.45	100.0
LLM-generated explanations were pertinent.	4.08	1.24	75.0
Using `ellmer` felt realistic and professional.	3.83	0.83	58.3
This assignment contributed to my overall learning.	4.83	0.39	100.0

numerical output helped them connect p-values, coefficients, and uncertainty statements to meaning in context (e.g., "Embedding a textual interpretation directly into the statistical output helped me link numerical results to their meaning," translated by the author). Second, the most frequently reported difficulties concerned technical integration of a local model (e.g., Ollama connectivity, latency, availability) and early ambiguity about expected S3 method requirements (e.g., "Making the LLM work was the most difficult part; the connection between my local model and RStudio failed," translated). Suggested improvements included providing a more minimal starter template and clearer guidance on package structure and interpretation expectations.

Finally, 11 of 12 respondents authorized the use of anonymized excerpts and code for teaching or research dissemination, and 9 of 12 indicated interest (yes/maybe) in contributing to a shared, collective version of the package (primarily through code contributions, testing/validation, and documentation).

From Individual Submissions to a Collaborative `ContextR` Package Following informal in-class discussions and the strong student engagement observed around this assignment, we decided to consolidate the most promising ideas into a shared, collaborative GitHub project aimed at developing a common `contextR` package. This repository will serve as a living, reusable implementation of the "context-aware" wrapper pattern, allowing iterative improvement through standard open-source workflows (issues, pull requests, and code review) and enabling future cohorts to contribute. In post-assignment feedback, most respondents authorized the use of anonymized excerpts and code for dissemination, and many expressed interest in contributing to a collective version of the package, which further motivated this transition from course artifacts to a community resource. The `contextR` package has been released to GitHub (Nicosia, 2025, https://github.com/AurelienNicosiaULaval/contextual-statistics-with-llm).

Connection to the Strategies of Wang et al. (2025)

This assignment aligns with the strategies outlined by Wang et al. (2025) by treating AI use as an intended learning outcome rather than a threat. Storing context and metadata with the statistical object supports process documentation and makes AI involvement explicit. Requiring students to validate and edit LLM-generated text promotes disciplined human–AI collaboration, turning model errors into opportunities to clarify statistical concepts.

Because explanations must reference both numerical results and dataset documentation, the task operationalizes the authors' recommendation for contextualized and open-ended assignments where generic AI output is insufficient. Embedding the LLM call inside a package

function also creates a natural entry point for discussing reproducibility, disclosure of AI use, and ethical AI use (e.g., data minimization and privacy-aware choices between local and hosted inference).

From an assessment perspective, the assignment shifts attention from the final product to students' judgment: how they design the interface with the LLM, justify prompting choices, and diagnose problematic outputs. This echoes the move toward process-focused evaluation advocated by Wang et al. (2025). The planned student feedback questionnaire is intended to complement rubric-based assessment by documenting perceived learning value and realism of structured LLM use in an R workflow.

References

Wang S, Xu L, Liu J, Zhai Y (2025). Addressing the Challenges of AI-Generated Assignment Submissions in Education: Insights and Strategies. *Journal of Data Science*, 24(1): 1–7. https://doi.org/10.6339/25-JDS1208

Wickham H, Cheng J, Jacobs A, Aden-Buie G, Schloerke B (2025). ellmer: Chat with Large Language Models. R package. https://CRAN.R-project.org/package=ellmer. https://doi.org/10.32614/CRAN.package.ellmer

Nicosia A (2025). contextR: Contextual statistics with large language models. R package, version v0.1.0. Zenodo. https://doi.org/10.5281/zenodo.18010065