Linear Models Project in R

M1–MIDO, 2025–2026

Table of contents

Context, dataset, and objectives

Context

Student performance can be associated with multiple dimensions (individual characteristics, learning environment, study habits, constraints, and health-related factors). This project uses a dataset to study how these factors relate to exam performance and to build a multiple linear regression model that is both interpretable and predictive.

Dataset

The dataset contains *n* observations (one row per student). In the dataset, the response variable `y` represents the exam score. The remaining variables are potential predictors (quantitative and categorical; see the codebook).

You will produce a clear, reproducible, and statistically rigorous analysis, including EDA, model construction, model checking, and predictive evaluation.

Global requirements (apply to the entire project)

- **Reproducibility:** use **relative paths only**, do not manually edit the raw dataset, and ensure your work runs from a fresh R session without interactive steps.
- **Randomness:** whenever you use random procedures (e.g., train/test split, resampling), you must set `set.seed(42)`.
- **Associations, not causality:** this is an observational dataset. Interpret estimated effects as **associations**, not causal effects.
- **No data leakage:** any preprocessing/feature construction used for validation must be learned on training data only and then applied to validation/test data.

Objective of the project

Your goal is to identify a **final multiple linear regression model** that relates exam score `y` to the available predictors in the dataset, with the dual aim of **interpretation (inference)** and **prediction**. The final model should be **parsimonious**, **statistically defensible**, and **reproducible**, while respecting the key assumptions of the linear model.

Minimum expected deliverables in your report (high level):

1. **A final model specification** (R formula) and a brief justification of the modeling strategy used (criteria from class/labs).
2. **Model results for the final model:** coefficient table with standard errors and **95% confidence intervals**, plus clear interpretation of the most important effects (in meaningful units and with correct factor reference categories).
3. **Model adequacy:** required diagnostic plots (residuals vs fitted, residuals vs predictors, residuals vs omitted variables, Q–Q plot, and influence/leverage diagnostics) and a brief discussion of limitations.
4. **Prediction performance (out-of-sample):** a reproducible evaluation using **`set.seed(42)`**, reporting a baseline model and your final model, with *Mean Square Error (MSE)* plus additional performance measures beyond Mean Squared Error (e.g., test R^2, MedAE, calibration plot).
5. **Scenario-based interpretation:** define at least **two realistic student profiles** and translate your model into predicted scores (or predicted differences).

Detailed guidance on expected contents is provided in Sections 1–6 below.

Codebook (variables and coding)

You will receive the dataset file `project.csv` and this codebook. Use these labels and codings in your analysis.

Variable name	Numeric values	Label
`id`		Student ID
`y`		Exam score
`age`		Age (years)
`agecat`		Age (years)
	1	[14.0,15.1[
	2	[15.1,16.1[
	3	[16.1,17.1[
	4	[17.1,18.1[
	5	[18.1,19.0]
`sexe`		Gender
	1	Female
	2	Male
	3	Other
`school_type`		School type
	1	Public
	2	Private
`parent_educ`		Parental education
	1	No Formal
	2	High School
	3	Graduate
	4	Post Graduate 1
	5	Post Graduate 2
	6	PHD
`study_hrs`		Weekly study hours
`sleep_hrs`		Sleep duration (hours)
`sleep_qual`		Sleep quality
	1	Poor
	2	Average
	3	Good
`attend_pct`		School attendance (%)
`attend_pct_cat`		School attendance (%)

(continued)

Variable name	Numeric values	Label
	1	[50, 62[
	2	[62, 72[
	3	[72, 81[
	4	[81,100]
`web_access`		Internet access
	1	No
	2	Yes
`trav_time`		Commute time
	1	<15 Min
	2	15-30 Min
	3	30-60 Min
	4	>60 Min
`extra_act`		Extracurricular activities
	1	No
	2	Yes
`study_method`		Study method
	1	Online Videos
	2	Coaching
	3	Notes
	4	Textbook
	5	Group Study
	6	Mixed

Important note on "duplicate representations": Some concepts appear twice (continuous and categorized versions), e.g. `age` vs `agecat`, and `attend_pct` vs `attend_pct_cat`. You should **justify** whether you use the continuous version, the categorical version, or (rarely) both (and then explain how you avoid multicollinearity and interpretability issues).

Expectations for the project (recommended contents)

The list below is indicative. You may change the order and add relevant elements.

1) Data management and variable preparation

- Import the dataset and report its basic structure (sample size, variable names, variable types, factor levels).
- Convert coded categorical variables into properly labelled factors. Choose sensible **reference categories**.
- Verify data integrity and plausibility: check duplicate IDs/rows, confirm plausible ranges, and identify extreme values that may affect estimation (outliers or high-leverage cases).
- Identify potentially redundant representations (e.g., continuous vs categorized versions of the same concept) and justify which version(s) you keep. Avoid including both unless you provide a clear rationale and address interpretability and collinearity.
- Keep preprocessing reproducible and organized (e.g., create a single `data_clean` object from the raw data).

2) Exploratory data analysis (EDA)

- Provide descriptive statistics for quantitative variables (mean, SD, median, IQR, etc.).
- Provide frequency tables (counts and proportions) for categorical variables.
- Visual exploration (EDA must inform modeling decisions):
  - Distribution of `y` (histogram/density and a boxplot); comment briefly on skewness and potential outliers.
  - `y` versus each predictor:
    - quantitative predictors: scatterplots with a smooth trend to assess linearity and potential heteroskedasticity,
    - categorical predictors: boxplots, violin plots or similar plots.
  - Summarize associations between predictors (numeric–numeric, categorical–categorical, and mixed) to anticipate collinearity and confounding.
- For ordinal predictors, include at least one visualization that helps assess monotone trends in `y` across ordered levels.
- Conclude EDA with 3–6 key findings that motivate: (i) candidate predictors, (ii) any transformations/nonlinear terms, and (iii) a small number of interaction hypotheses to test.

3) Building regression models

- Begin with simple regressions (one predictor at a time) as descriptive baseline associations and to build intuition.
- Build multiple linear regression models with a justified set of predictors (substantive motivation and evidence from EDA and class/lab criteria).
- Consider transformations or re-expressions only when justified by EDA/diagnostics; explain how interpretation changes.
- Interactions: include them only if motivated by a **clear, testable hypothesis**. We suggest keeping them few.
- Compare candidate models using criteria covered in class/labs (e.g., nested F-tests when appropriate, AIC/BIC, adjusted R^2, etc.). Summarize comparisons in a short model-selection table and justify the final choice.

  You must compare at least **three** candidate models and justify the final choice using at least **two** distinct criteria (e.g., one inferential and one predictive/selection criterion).

4) Diagnostics, assumptions, and robustness

- Provide and comment briefly on diagnostics for your final model (include at least: residuals vs fitted, residuals vs predictors, residuals vs omitted variables, Q–Q plot, and influence/leverage diagnostics such as Cook's distance).
- If issues are detected (heteroskedasticity, nonlinearity, influential observations), discuss implications for inference and prediction.
- Do not remove observations unless you can justify that they are data errors. Conclude with a short statement on whether your main findings are stable under the sensitivity check(s).

5) Interpretation and inference

For the final model:

- Write the model equation clearly (including factor coding and reference categories), as in lab session.
- Present a coefficient table with estimates, standard errors, and **95% confidence intervals**.
- Interpret 3–6 key terms with attention to magnitude and units (meaningful increments for continuous predictors; correct contrasts for categorical/ordinal predictors).
- Provide **scenario-based interpretation**: define at least two realistic student profiles and report predicted scores (or predicted differences), with confidence intervals.
- Discuss practical significance versus statistical significance (do not rely soly on p-values).
- If you include interactions, interpret effects conditionally and illustrate them appropriately.
- Briefly summarize overall model fit (e.g., R^2, adjusted R^2, residual standard deviation, etc.).

6) Predictive performance (graded)

- During model development, use a validation strategy to compare candidate models (train/test split, repeated splits, or k-fold CV). Whenever randomness is involved, set **`set.seed(42)`** and report your protocol (split proportion and, if relevant, number of repeats/folds).
- Avoid data leakage: any preprocessing/feature construction used for validation must be learned on training data only and applied to validation/test data.
- Report predictive performance on your validation/test data using **Mean Squared Error** (MSE) plus the following:
  - **out-of-sample** R^2,
  - **Median Absolute Error** (MedAE),
  - a **calibration check** (predicted vs observed plot with a brief comment).
- Always include a **baseline** model (e.g., predicting the training mean of `y`) and report improvement over baseline.
- Include a short table comparing baseline, main candidate model(s), and the final model under the same validation protocol.
- The instructor will also evaluate each submitted final model on a separate held-out test dataset (**n = 1000**). Ranking and bonus points will be based primarily on predictive performance on that instructor test set (typically Mean Squared Error). If performance is very similar, more parsimonious specifications may be preferred as a tie-breaker.

Deliverables and submission

Recommended structure of the report

- Introduction: problem, dataset, hypotheses
- Methods: data management, modeling strategy, selection approach
- Results: EDA, model comparisons, final model estimates and interpretation
- Diagnostics and limitations
- Predictive evaluation (train/test)
- Conclusion

What to submit (mandatory)

A Quarto HTML template (`project_template_minimal.qmd`) is provided (see Quarto for documentation). You may use it as a starting point and rename it for your project.

Submit a single folder (or a single compressed file) containing:

1. A **rendered report** that the instructor can read (**at least one** of: HTML, PDF, or DOCX). The report must include all results, tables, figures, and written explanations.
2. The **source file** used to generate the report: either a Quarto document (`.qmd`) or an R Markdown document (`.Rmd`).
3. **At least one R script file** (**mandatory**) that reproduces the full analysis end-to-end from a clean R session (e.g., `R/run_analysis.R`). Running `source("R/run_analysis.R")` must execute without error and (re)generate the analysis.

Reproducibility rules (mandatory)

- Use **relative paths only** (no absolute paths specific to your computer).
- Do not manually edit the raw dataset file.
- Your project must run from a fresh R session without interactive steps.

Running Quarto from RStudio

You can render your Quarto report directly from RStudio in several standard ways:

- **Using the Terminal inside RStudio (recommended):**
  - Open the **Terminal** tab in RStudio and run:

    ```
    quarto render report.qmd
    ```

- **Using the R Console (recommended):**
  - Render the current file:

    ```
    quarto::quarto_render("report.qmd")
    ```

  - Render to a specific format:

    ```
    quarto::quarto_render("report.qmd", output_format = "html")
    ```

- **Using the Render button (not recommended):**
  1. Open your `.qmd` file in RStudio.
  2. Click **Render** (top of the editor pane).

Notes

- Work inside an RStudio Project (`.Rproj`) and use **relative paths** (e.g., `data/project.csv`).
- If you render to PDF, you may need a LaTeX installation (tinytex is a lightweight, LaTeX distribution).

Academic integrity and use of generative AI

- You may use generative AI tools as *support* (e.g., coding assistance, debugging, improving wording, or suggesting alternative analyses). However, the **statistical choices**, **interpretation**, and **final writing** must reflect your own reasoning and understanding.
- You are responsible for verifying the correctness of **all** code, outputs, and interpretations. Do not copy AI-generated text or conclusions without checking them against your results.
- **No fabrication:** do not invent numerical results, tables, figures, references, or methodological steps.

Disclosure (mandatory, short): Include a brief "AI use statement" (3–6 lines) specifying:

- the tool(s) used,
- what you used them for (e.g., plotting code, debugging, language editing),
- which part(s) of the project they affected,
- and a sentence confirming that you verified all results.

Do not paste long AI transcripts into the report.