Teaching Portfolio

Aurélien Nicosia

Aurélien Nicosia

2026-01-30

Table of contents

Teaching Portfolio

Introduction and Structure

The purpose of this teaching portfolio is to present a holistic and documented view of my contribution to university teaching in statistics and data science. It is not a simple collection of syllabi, but an evidence-based argument for my teaching philosophy: training autonomous, critical, and technically competent practitioners.

What You Will Find in This Portfolio

This document is organized as a coherent narrative spanning the main dimensions of my teaching work. You can navigate directly to the following sections:

1. **Foundations**: alignment between my teaching philosophy and its concrete implementation (learning by doing, immersion).
   - Teaching Philosophy
   - Teaching Practice
2. **Innovation and Improvement**: case studies on AI integration and impact analysis with students.
   - Pedagogical Innovation
   - Evaluation and Continuous Improvement
3. **Leadership and Outreach**: action beyond my own courses, shared tools, and future vision.
   - Teaching Tools for the Academic Community
   - Dissemination and Impact
   - Future Trajectory and Vision
   - Student Support and Inclusion

Navigation

Chapters are accessible from the sidebar or through the links above.

Teaching Philosophy

My pedagogical approach unfolds across a wide range of courses and through my coordination of the Centre de Dépannage et d'Apprentissage (CDA). It is structured around five core pillars.

Learning by doing (project-based learning and authentic situations). My teaching rests on a simple principle: statistics and data science are learned by practicing tasks close to real-world work, rather than by stacking decontextualized techniques. In my courses, this takes the form of projects, case studies, written and oral deliverables, and a constant effort to connect statistical results to a concrete context and audience. This orientation is explicit in my teaching philosophy and in the design of STT-1100, where students take on professional roles and produce artifacts (reports, GitHub repositories, presentations).

The classroom as a collective workshop (collaboration -> autonomy). I design the classroom as a workshop: peer interaction and small-group problem solving are part of learning, with progression toward greater autonomy (more open structure, freer choices). In STT-1100, this progression is reflected in the evolution of the Challenges and in a final team project, with assessment including peer review and GitHub repository traces. In STT-4230/6230 (R for Scientists), collaboration is supported by GitHub and software development practices (documentation, version control).

Feedback as a core learning component. I prioritize frequent feedback, often formative and sometimes automated, to support progress without unnecessarily increasing assessment load. My teaching philosophy explicitly mentions the use of learnr and Quarto notebooks, as well as feedback mechanisms through R or GitHub. In STT-1100, feedback is also embedded directly into Adventures through interventions and immediate-correction questions.

Reproducibility as a language of thought. Reproducibility is not an add-on technical feature: I treat it as an intellectual skill (organization, explicitness, justification) required for applied statistics. This is implemented through the use of Quarto (documents integrating code and text), Git/GitHub (work traceability), and related assessment criteria (repository organization, commit messages).

Generative AI: a tool to frame and make visible. I integrate generative AI within a responsible framework: it is a support tool (error clarification, documentation translation, example generation), but it does not replace reasoning or the obligation to verify. In STT-1100, a course-specific GPT assistant is described as a support resource, with instructions and reminders about verification. In an advanced R course, I designed an assessment where LLM use is mandatory and assessed, with explicit criteria for transparency, anti-hallucination safeguards, and data minimization.

Teaching Practice

My teaching practice is defined by a practice-first approach: I do not prepare students to do statistics "later"; I have them act as statisticians from day one. This philosophy relies on four pillars that define my pedagogical signature.

Pedagogical Signature

1. Situational authenticity: assessments are not school exercises, but simulations of professional mandates (consultant, analyst, developer).
2. Reproducibility by default: technical rigor (Git, Quarto) is not a peripheral skill; it is the language of scientific evidence itself.
3. Structured collaboration: the classroom is a workshop where code is written, read, and critiqued collectively, reproducing the dynamics of a modern data team.
4. Technological responsibility: the use of tools (AI, libraries) is encouraged but always framed by requirements for validation and transparency.

Learning Through Projects: The "Learning by Doing" Strategy

I prioritize project-based learning as the main driver of skills development. This strategy adapts across all levels, from introductory courses to specialized seminars, by adjusting the level of autonomy.

Immersion at Universite Laval: The Junior Data Scientist (STT-1100)

In this first-year course, the project-based approach is scripted to reduce code-related anxiety.

- Intention: transform the perception of failure ("I am bad") into iteration ("my code has a bug").
- Implementation: students take on the role of a "junior analyst" in a fictional institute. Each module is a mission (for example, "Antarctic Mission" for data cleaning) that ends with an executable deliverable.

Technical Projects at UQAC: Theory in Action

At Universite du Quebec a Chicoutimi, I apply the same logic in courses with more theoretical or specialized content.

- Data Visualization and Interface Design (8INF416): instead of in-class exams, students complete a critique and redesign of an existing media visualization, justifying their choices through graphical perception theory.
- Statistical Methods for Massive Data (8STT108): the course is built around architecture mini-projects where students must not only analyze data, but select and justify the tool that matches data volume.
- Introduction to Data Science (8INF404): statistics course for data science (official title).
- Pedagogical gain: this approach forces students to go beyond the "how" (syntax) and address the "why" (analysis strategy).

R for Scientists (STT-4230/6230)

For advanced students, the project becomes a full software development exercise.

- Example deliverables: creation of a functional R package, interactive web app (shiny), interactive tutorial (learnr), flexdashboard, etc.
- Professional practice: continuous integration (CI/CD) through GitHub Actions; the code must not only produce the correct answer, but pass automated tests on a remote machine.

> **Note**
>
> Project as alignment: Across all these courses, the project is not a peripheral activity ("if there is time"), but the core of pedagogical alignment. Lectures are resources to support project success, not the opposite.

Supporting the Next Generation of Researchers

My involvement at UQAC extends beyond the classroom; it is part of a broader commitment to building an autonomous and rigorous scientific community in the region.

Technical mentoring and reflective posture I support students in their first steps toward research or the labor market by emphasizing posture. Beyond code, I work with them on:

- the ability to read and critique technical documentation,
- autonomy in debugging (moving from "it does not work" to "here is the error and what I already tried"),
- communication of results to non-specialists.

This close supervision of internships and projects helps identify talent and guide students toward graduate studies or key industry roles, strengthening the local scientific ecosystem.

Assessment and Integrity in the AI Era

The rise of LLMs (ChatGPT, Claude) has made assessments based on knowledge recall or simple code production obsolete. I redesigned my assessments to measure the process rather than only the final product.

1. Mandatory traceability: in STT-1100, assessment includes Git commit history. A "perfect" project that appears suddenly without history is suspicious; an iteratively built project is valued.
2. AI as a supervised partner (STT-4230):
   - I do not ask students to "avoid AI".
   - Assessment requires integrating an LLM to generate a contextual interpretation of statistical results.
   - Success criterion: the student's ability to preprocess the data sent to the model (minimization) and validate model output (hallucination detection).

> ⚠ Warning
>
> Paradigm shift: We no longer assess only "Does the code run?", but "Is the process robust, transparent, and verified?" Academic integrity becomes a professional verification skill.

Pedagogical Innovation

This section presents three innovation initiatives I led to address contemporary teaching challenges: student engagement in data science, assessment in the era of generative AI, and modernization of academic support.

Immersive Approach: STT-1100 Introduction to Data Science

Context and Pedagogical Intent

STT-1100 was redesigned as part of the update of the undergraduate program in statistics, now the undergraduate program in statistics and data science. The goal was twofold: update the content so it reflects modern data science practice, and renew the learning experience to strengthen student motivation and engagement.

This redesign addresses a common limitation of introductory courses: progression centered on isolated techniques, often disconnected from a full workflow. STT-1100 instead provides direct entry into real practice: learning data science by doing data science, with reproducible deliverables and explicit attention to tools, collaboration, ethics, and communication.

A Stable and Explicit Architecture: Plan, Adventure, Challenge

The course is structured around a recurring three-part framework visible throughout the course notes:

- Learning Plan: objectives, readings, preparation
- Adventure: guided lab session, professional scenario, integrated activities
- Challenge: autonomous work, graded deliverable, demonstration of mastery

This structure is intentional: it stabilizes cognitive load (students always know how a module works) while enabling gradual growth in autonomy over the semester.

Link to course notes (Quarto website): https://aureliennicosiaulaval.github.io/STT-1100_notes_de_cours/

Immersive Design: Learning Through Roles and Missions

A central feature of the course is narrative immersion: students take on concrete roles (junior analyst, data journalist, data engineer, etc.) and produce deliverables for an explicit audience (client, supervisor, partner team, management).

This scenario-based design serves three pedagogical functions:

1. Give immediate meaning to tools and concepts (the why comes before the how).
2. Strengthen communication skills (writing for an audience, not only for the instructor).
3. Reproduce realistic project dynamics (constraints, traceability, revisions, collaboration).

Module Overview

The course includes ten thematic modules plus a final project. Each module corresponds to a mission and a role.

Module	Theme (course notes)	Scenario and role	Main deliverable
1	Diving into data science	Tool discovery, exploratory mindset	First Quarto HTML report
2	GitHub and data visualization	Antarctic mission, data scientist role	Quarto notebook + GitHub repository
3	Categories in all forms	Data journalism	Quarto article for a newspaper
4	Factors and data cleaning	Junior data engineering	Script, cleaning log, clean CSV
5	Exploring relationships between variables	Statistician	Report addressing business questions
6	Collaboration and reproducibility in GitHub	Team collaboration (rotating roles)	Cross-review using GitHub issues
7	Visualization, ethics, and security	Data ethics specialist	Critical analysis and redesign of a visualization
8	Automation and web exploration	Web extraction consultant	Scraping function and metadata
9	Prediction and bias	Data scientist	Short capsule or synthesis deliverable
10	Text and dashboard	Business analysis	Dashboard and web sharing

Note: exact titles and associated resources (plans, adventures, challenges) are available on the course website.

Tools and Environment: Reproducibility as a Learning Objective

The course adopts a toolset that reflects contemporary practice:

- R and tidyverse for data manipulation, visualization, and analysis
- Quarto for reports, notebooks, and teaching material (computational literacy)
- Git and GitHub for version control, collaboration, and traceability
- A course R package grouping datasets and utility functions (uniform resource distribution)
- A course-specific AI assistant, framed by responsible-use guidelines

The originality is not only using these tools, but placing them at the core of success criteria: repository organization, code clarity, workflow consistency, and ability to produce a reproducible artifact.

Assessments and Progression

Assessments are aligned with the targeted skills:

- Weekly challenges: regular production of reproducible deliverables
- Exams designed in the spirit of adventures and challenges (solve tasks, do not recite procedures)
- Final project: full team-based analysis, GitHub repository, written and oral communication

The final project is designed as a first complete mission: data selection, question formulation, exploration, possible simple modeling, visualization, communication, and repository reproducibility.

Responsible Integration of Generative AI

The course treats AI neither as a shortcut nor as a taboo. It is integrated as one resource among others, with one central instruction: verify, understand, document.

Examples of encouraged uses:

- clarifying an error message and suggesting diagnostic leads
- rephrasing or translating technical documentation
- generating a code prototype to adapt and annotate

Examples of explicitly discouraged uses:

- delegating an entire deliverable
- accepting an answer without validation
- using AI without keeping a trace of the process

Scholarship of Teaching and Learning (SoTL) Approach

The course redesign includes an evaluation process: an anonymous questionnaire administered after the end of the semester to document perceptions of the learning design, motivation, engagement, and material clarity.

The intent is twofold:

- improve the course based on structured feedback
- disseminate results and pedagogical design in academic venues (presentation, article), in alignment with a SoTL approach

STT-4230 / STT-6230 - R for Scientists: Mandatory and Supervised LLM Integration

Context and Motivation

The arrival of language models (LLMs) has made part of traditional programming and analysis assessments obsolete, especially when only a final product (script, report, answer) is assessed and can be generated quickly. In STT-4230/6230, I chose to treat this reality as an explicit learning objective: students must learn to use an LLM in a disciplined, transparent, and reproducible way, rather than trying to bypass or hide its use.

The approach is based on a simple principle: LLM use is not assessed in itself; it is assessed as a professional competency integrated into a statistical and software workflow. Students are evaluated on the quality of this integration: design, traceability, robustness, prompt quality, risk control (hallucinations, bias, confidentiality), and ability to produce an executable artifact in a clean environment.

Pedagogical Approach: Make AI Unavoidable, but Verifiable

Rather than telling students "do not use AI," the assessment makes AI mandatory while imposing constraints that force human judgment:

- the LLM must produce an interpretation grounded in numerical outputs computed in R
- generated text must remain separate from numerical results and must be explicitly presented as interpretation
- the design must be reproducible, testable, and robust to LLM failures
- students must implement safeguards against invented values and arguments not present in the results

This approach turns potential LLM errors into pedagogical objects: they become opportunities to discuss validation, statistical communication, and tool limitations.

> **Note**
>
> Core pedagogical objective: move students from "answer consumer" mode to "workflow engineer" mode, able to orchestrate an LLM as a software component with traceability, validation, and risk management.

Flagship Assessment: Contextual Statistical Analysis with LLM as an R Package

The key course assessment requires students to develop an R package that extends common statistical functions to produce contextualized outputs. The package format was chosen for several reasons:

- requirement for a standard software structure (DESCRIPTION/NAMESPACE, dependencies, export)
- possibility of evaluation in a clean environment (installation, execution, demonstration)
- emphasis on design, documentation, and reproducibility

The main deliverable is a package implementing "contextual" functions (for example wrappers around base functions) and associated S3 methods.

Contextualized Functions and S3 Objects

Students implement at least two contextualized functions selected from common workflows (modeling, tests, tables, visualization). Each function returns an S3 object that stores:

- the original statistical object (e.g., lm, htest)
- a context field (free text or excerpt from dataset documentation)
- reproducibility and transparency metadata (call, formula, LLM parameters, model, provider)

A "Minimal API" to Standardize Assessment Without Limiting Creativity

A frequent challenge in software design assessment is interface variability, which increases grading cost and reduces fairness. To address this, the assignment defines a minimal interface to follow:

- required function names
- argument structure
- return object type
- expected S3 methods

Students still retain broad freedom in internal implementation, prompting strategy, additional functions, and metadata enrichment. This strategy enables coherent criterion-based grading and encourages relevant extensions.

LLM Integration: Technical Requirements and Safeguards

LLM integration is implemented through a dedicated function, for example `ctx_llm_generate()`, which is mandatory. This function combines a controlled summary of numerical results with context, then queries an LLM through a local provider (e.g., Ollama) or a remote API.

Assessment enforces explicit requirements:

- anti-hallucination instruction in the prompt (do not invent values absent from results)
- context length and relevance control (e.g., extract only relevant documentation sections)
- robust fallback mode when the LLM is unavailable (clear message, no crash)
- visible separation between numerical output and interpretive text
- explicit provider and model parameters recorded in metadata

Student Feedback and Continuous Improvement

An anonymous post-assessment questionnaire documents perceptions of the approach. Results indicate very positive perceptions of overall learning and strong perceived value of S3 work. The relevance of LLM-generated explanations is rated positively on average, with recurring technical difficulties related to local models (connection, latency, availability). Suggested improvements converge on providing a minimal starter template and clearer expectations on package structure and interpretation quality.

This feedback feeds an iterative logic: clarify the minimal skeleton, better frame reproducibility requirements, and provide shorter but canonical examples.

Outcomes and Transferability

This mandatory LLM integration in STT-4230/6230 produces three major outcomes:

- training in professional LLM use (tool of work, not oracle)
- strengthened reproducibility learning (package, clean execution, fallback modes)
- more robust assessment in the face of generative AI, because it evaluates process, design, and judgment

In the medium term, this approach also supports consolidation of student contributions toward a collective project (shared package), based on open-source practices (issues, pull requests, code review), and transforms course artifacts into a sustainable resource.

Academic Dissemination

This pedagogical approach is discussed in an invited contribution to the *Journal of Data Science*, in response to Wang et al. (2025) on challenges raised by AI-generated assignment submissions. The discussion, entitled *Discussion of 'Addressing the Challenges of AI-Generated Assignment Submissions in Education'*, details how this assignment operationalizes recent literature recommendations: make AI an explicit, transparent, and verifiable learning partner.

GPT-CDA - Specialized AI Assistant for the Centre de Dépannage et d'Apprentissage (CDA)

Context and Pedagogical Problem

The Centre de Dépannage et d'Apprentissage (CDA) in the Faculty of Science and Engineering provides support in mathematics and statistics for about twenty courses, relying on teaching assistants who support students in conceptual understanding, problem solving, and study methods.

Since 2024, two signals have been documented:

- decreased attendance: the CDA historically received about 3,000 visits per semester; since 2024, attendance has dropped by about one third
- a change in request profile: students increasingly arrive with solutions generated by AI tools (e.g., ChatGPT) that they do not always understand and that are often outside course scope

This situation creates an opportunity for pedagogical innovation: instead of competing with public AI tools, the CDA can offer a specialized, supervised assistant aligned with course expectations and designed to preserve human support.

Design Vision: A Hybrid AI + Human Model

The GPT-CDA project aims to develop a specialized conversational chatbot to support students in solving mathematics and statistics problems, with two guiding principles:

- disciplinary contextualization: answers aligned with expectations of courses supported by the CDA (syllabi, theory, exercise sets, past exams, internal resources)
- hybrid support: AI provides a first pedagogical mediation, then hands off to a teaching assistant when conceptual or methodological difficulty requires it

The goal is not to replace tutoring, but to better distribute pedagogical effort: provide immediate and structured support while preserving human intervention for high-value cases.

> **Note**
>
> Pedagogical stance of the tool GPT-CDA is designed as a learning mediator: explanations, detection of misunderstanding, and redirection to a teaching assistant when needed.

Objectives and Target Population

Main objectives:

- provide self-service access to GPT-CDA in the CDA physical space
- support understanding through detailed explanations and complementary resources

- detect situations requiring human intervention and redirect to a teaching assistant
- promote responsible AI use in learning
- stimulate student engagement and motivation, and support academic success

Target population:

- students enrolled in mathematics or statistics courses supported by the CDA
- for university courses: about 2,000 students per semester, across 32 programs and 6 faculties
- added scope: remedial courses (STT-0150, MAT-0130, MAT-0150, MAT-0250, MAT-0260) reaching a broader population on campus

Technopedagogical Design

1) Alignment with Courses, Not with General-Purpose AI

The core design choice is integration of CDA-relevant pedagogical materials (syllabi, content, exercises, exams, internal resources), so responses remain coherent with each course framework.

This contextualization addresses a common limitation of general assistants: they may provide mathematically correct solutions that are systematically outside the course framework, which can harm understanding and learning.

2) Detection and Redirection to a Teaching Assistant

An explicit project requirement is the assistant's ability to detect situations where human support is needed and then redirect the student to a teaching assistant.

This anchors the system in a service logic: AI serves as a first support layer but does not try to close every case autonomously.

3) Access and Support Outside Regular Hours

A key expected impact is improved access to academic support, especially outside regular CDA hours.

Governance, Ethics, and Responsibility

Deployment includes usage guides and integration protocols covering:

- role and limits of the tool
- redirection toward a human
- traceability of support interactions

The intention is to support autonomy without replacing human pedagogical interaction, and to promote responsible AI use in learning.

> Warning
>
> Explicit watch point The value of GPT-CDA depends on mediation quality and governance, not only model performance. This is why the project includes guides, protocols, and monitoring and evaluation tools.

Deliverables

The project's core deliverables are:

- a functional platform integrating the GPT-CDA chatbot
- a user guide for students and teaching assistants

Evaluation and Continuous Improvement

Evaluation of my teaching relies on data triangulation: quantitative institutional evaluations, detailed qualitative reports, and direct feedback on pedagogical innovations.

Quantitative Synthesis of Teaching Evaluations

My portfolio shows consistent student appreciation, recognized through repeated receipt of the "Enseignant Etoile" award (2014, 2023, 2024, 2025) at the Faculty of Science and Engineering.

Overall Trend

Scores remain high and often above the departmental average. The figure below normalizes results (as a percentage of the maximum) to enable cross-comparison despite changes in scoring scales (3, 4, or 100).

Course-Level Detail (Most Recent Session Available)

Course	Title	Semester	Score	Scale
MAT-1905	Arithmetic (Elementary Education)	Summer 2024	2.98	/3
MAT-1906	Geometry (Elementary Education)	Winter 2025	2.82	/3
STT-1100	Intro. Statistical Software	Winter 2024	2.91	/3
STT-2200	Data Analysis	Fall 2024	2.96	/3
STT-4230	R for Scientists	Fall 2024	2.97	/3
8INF416	Visualization (UQAC)	Winter 2025	3.81	/4
8STT108	Massive Data (UQAC)	Winter 2025	3.86	/4

(Consolidated table from CV and institutional reports)

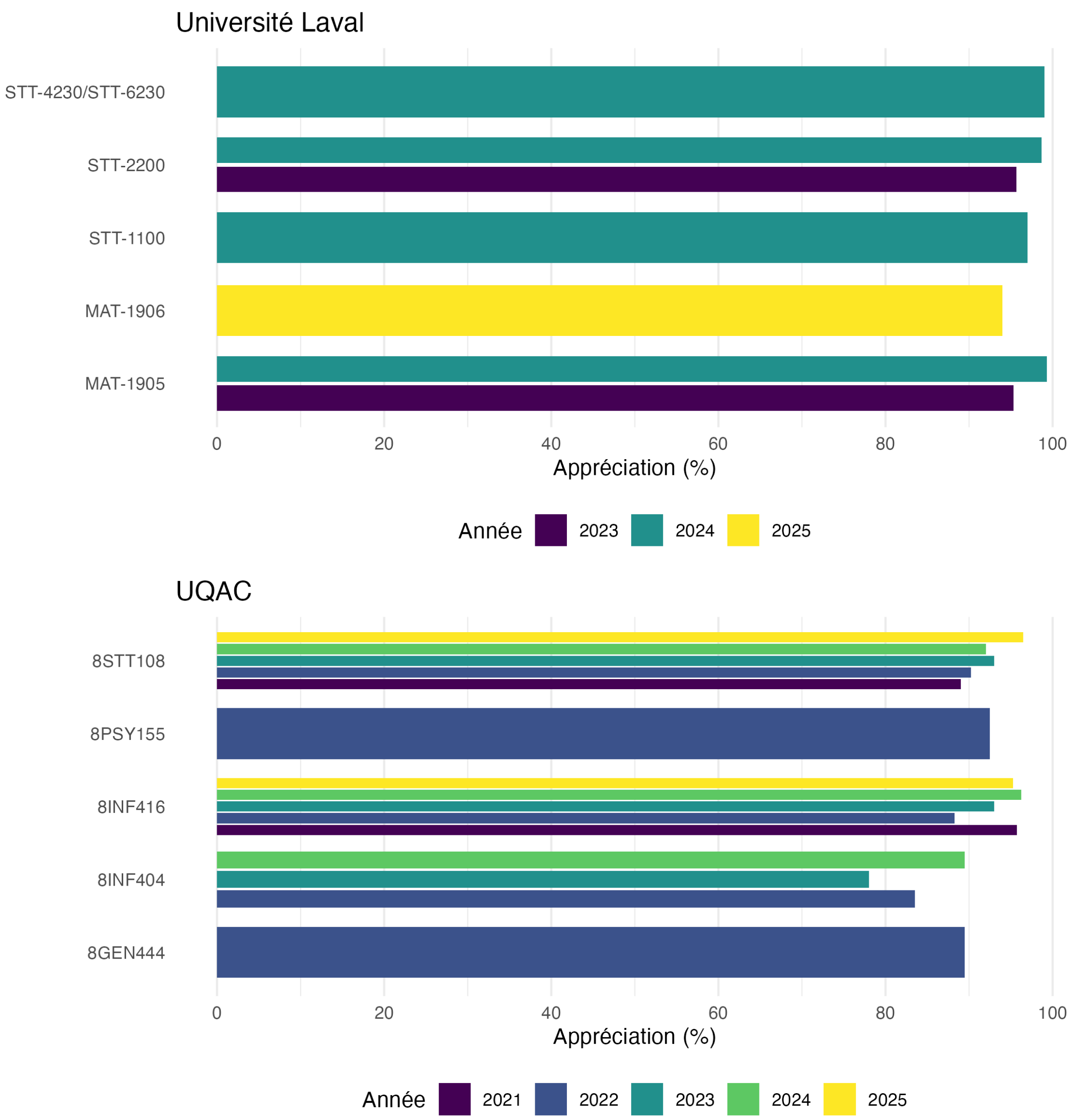

Figure 1: Teaching evaluations by course and year

Qualitative Analysis and Feedback on Innovation

Beyond scores, I focus on understanding the learning experience, especially when introducing new methods.

Case Study: "Mandatory AI" Assessment (STT-4230)

Within the LLM innovation described in Section 4, I conducted a specific survey to measure perceived pedagogical impact.

- Perceived success: students report that the exercise strongly helped them understand object-oriented programming (score 4.75/5).
- Technical friction: qualitative comments highlighted the difficulty of installing local models (Ollama), creating technical inequality.
- Corrective action: this feedback directly led to creating a starter template for the next iteration, to reduce setup-related cognitive load.

Case Study: Cognitive Load in STT-1100

The simultaneous introduction of Git, R, and Quarto was identified as a barrier for complete beginners.

- Corrective action: the course plan was adjusted to introduce tools more sequentially (GitHub only from module 2 onward).

Philosophy of Learning Assessment

My approach to student assessment mirrors my approach to my own teaching: formative and iterative.

- Automated feedback: in programming courses, unit tests provide immediate feedback, enabling trial-and-error without penalty.
- Peer assessment: used in final projects, it forces students to adopt a critical stance and read code other than their own.

Tools for the Teaching Community

Beyond my own courses, I develop and maintain several technical tools aimed at supporting university instructors in statistics and data science. These open-source projects are designed to reduce technological friction for adopting reproducible practices.

Teaching Resources for Statistics and Data Science Instruction

I provide the community with a central repository that serves as a "starter kit" for building modern course websites with Quarto. This project offers a simple entry point for colleagues who want to modernize course material without needing to become web development experts.

- Website structure and available resources:
  - Ready-to-use templates: Quarto (`.qmd`) templates for Reveal.js presentations, formatted assignments, and reproducible exams.
  - Technical tools: libraries of useful R functions for teaching and guides for integrating GitHub in class.
  - Step-by-step tutorials: detailed guides for getting started with R, RStudio, Quarto, and Git in a teaching context.
- Philosophy: enable instructors to focus on pedagogical content (substance) by providing robust, turnkey technical infrastructure (form).
- Access and documentation: https://aureliennicosiaulaval.github.io/site_ressources_SSD/

R Package `UlavalSSD`

To support data science teaching at Universite Laval, I created the R package `UlavalSSD`.

- Problem: students (and instructors) lose valuable time managing file paths, character encoding, and multiple dependency installations.
- Solution: a single package that distributes:
  - all pedagogical datasets used in courses (cleaned and documented),
  - helper functions for tool installation,
  - preconfigured RStudio project templates.
- Impact: this package is now used not only in my courses, but also by colleagues teaching related courses, helping standardize the software environment.
- Access: GitHub - UlavalSSD

Package `contextR`: AI Pedagogy

Currently in development, the `contextR` package is an experimental tool for instructors who want to integrate LLMs locally in R (see Section).

- Objective: provide a simple interface so students can obtain a contextual interpretation of statistical output (e.g., `lm` regression summary) generated by AI, but strictly framed by pedagogical prompts defined by the instructor.
- Teaching use: supports activities where students critique AI interpretations, turning the tool into a Socratic debate partner.
- Access: GitHub - ContextR

Package `tutorizeR`: Automation of Interactive Tutorials

The `tutorizeR` package is a pedagogical productivity tool designed to quickly transform static content into interactive learning experiences.

- Problem: creating interactive tutorials (such as `learnr`) often requires time-consuming rewriting of existing course material.
- Solution: this package automatically converts existing Quarto (`.qmd`) or R Markdown documents into interactive modules, generating exercises and validation tests from source code.
- Features:
  - Supports conversion to `learnr` (Shiny) or `quarto-live` (WebAssembly, serverless).
  - Transforms code blocks into exercises with solutions and automatic validation.
- Access: GitHub - tutorizeR

Dissemination and Outreach

This section documents dissemination activities related to my teaching practices within academic and professional communities. These contributions show peer recognition of my expertise in both university pedagogy and data science.

Invited Conference: Association des statisticiennes et statisticiens du Quebec (ASSQ) (2025)

- Context: ASSQ annual conference, held on May 15, 2025, at Universite Laval (Quebec City).
- Title: "What's New in Data Science Tools?"
- Content: overview of recent environments (Posit/RStudio), reproducibility with Quarto and analysis publishing, plus a survey of generative AI and agent use in Shiny applications.
- Resources: GitHub - ASSQ presentation materials

Training the Next Generation of Researchers: Interdisciplinary School Tools & Methods (EIOM) (since 2022)

- Context: Interdisciplinary School Tools & Methods (EIOM), Universite Laval.
- Title: workshops on statistical modeling and good practices in R programming.
- Content: training in collaborative project management with GitHub, scientific publishing with Quarto, data analysis, visualization, and modeling (linear and logistic regression).
- Resources: GitHub - EIOM workshops

Invited Academic Seminar: HEC Montreal (2026)

- Institution: HEC Montreal, Department of Decision Sciences.
- Title: "Integrating Artificial Intelligence into Teaching, Academic Support, and Data Science Research" (January 28, 2026).
- Content: presentation of GPT-CDA and integration of language models into pedagogical and research activities, with emphasis on validation, reproducibility, and scientific integrity.
- Resources: Seminar slides

Institutional Outreach: Delegation Visits (November 2025)

The innovative initiatives implemented at the Centre de Depannage et d'Apprentissage (CDA) have attracted interest from other major institutions.

- Visitors: joint delegation from HEC Montreal and McGill University.
- Purpose: on-site observation of GPT-CDA deployment.
- Outcome: exchange of expertise on implementing specialized AI assistants for academic support.

Future Trajectory and Vision

My vision for the coming years is structured around three horizons: consolidating the support ecosystem, creating a new advanced course offering, and formalizing my pedagogical research (SoTL).

Horizon 1: GPT-CDA Next Step (Phase 2, 2026-2027)

Why a Phase 2

Phase 1 made it possible to develop the agent, integrate it into CDA facilities, train staff, test in real conditions, and progressively deploy the tool.

Phase 2 now targets sustainability and service expansion: instead of depending on an external interface, the objective is to build an institutional in-house web platform that accesses the model via an API and enables a controlled, scalable experience integrated with CDA needs.

Vision: Hybrid Online Platform (AI + Teaching Assistants)

The core of Phase 2 is native integration of a hybrid online pathway:

- students use GPT-CDA in self-service mode
- during opening hours, they can be transferred to an online teaching assistant (chat)
- transfer preserves relevant history and context to make human intervention more effective

This step is also a direct response to accessibility challenges: supporting students who cannot physically come to the CDA and better covering needs beyond in-person support.

> **Note**
>
> Pedagogical positioning Phase 2 does not aim to replace human support. It organizes help into two layers: handling simple or recurrent requests through the agent, then human intervention when needed, with continuity of context.

Operational Objectives of Phase 2

1. Design and deploy an in-house API-based web platform, including UL authentication, access control, logging, and administration tools.
2. Integrate a hybrid pathway: interaction with GPT-CDA, then transfer to an online teaching assistant with context preservation during opening hours.
3. Implement and document a fully remote version of the service (agent + human interaction) in one course, then extend to all courses supported by the CDA.
4. Evaluate acceptability and large-scale adoption conditions: preferences (human, AI, mixed), barriers, user and non-user profiles, usage trace analysis.
5. Produce reusable deliverables: responsible-use guide, technical documentation, implementation recommendations.

Implementation Plan (May 2026 to April 2027)

Phase 2 is planned from May 1, 2026 to April 30, 2027.

1. Analysis and specifications (May to June 2026)
   - consolidate Phase 1 feedback, clarify student pathway
   - confidentiality requirements, monitoring needs, transfer modalities to teaching assistants
2. In-house platform development (June to September 2026)
   - authenticated web interface (UL account)
   - conversational model integration via API
   - access management, minimal logging, admin dashboard
3. Online human support module (September to October 2026)
   - teaching assistant chat with queue management
   - transfer mechanism from GPT-CDA preserving relevant context
4. Operational pilot and testing (October to December 2026)
   - controlled deployment in one fully remote course
   - student and assistant feedback, fixes, stabilization
5. Expanded deployment and evaluation (January to April 2027)
   - extension to all CDA-supported courses
   - training and documentation
   - acceptability study (survey, focus groups, usage analysis)

Evaluation: What Will Be Measured and Why

Phase 2 explicitly documents adoption conditions to guide deployment decisions and continuous improvement.

Indicators and typical questions:

- platform usage volume and recurrence
- rate of transfer to teaching assistants and when transfer occurs
- satisfaction, perceived utility, preferences (human, AI, mixed)
- profiles of students who adopt or do not adopt, reported barriers

Risk Management and Responsible Implementation

Phase 2 includes explicit attention to:

- limiting out-of-scope responses (guidance toward course concepts and methods, pedagogical framing, redirection in case of uncertainty)
- confidentiality and data minimization (clear student information, validation of retention parameters)
- academic integrity: support for process rather than generation of final answers when inappropriate
- data collection under required authorizations, including ethics process when needed

> ⚠ Warning
>
> Watch point The added value of Phase 2 depends on two difficult but essential elements: governance (access, logging, confidentiality) and context transfer to humans. The in-house platform is precisely the lever that makes these elements controllable.

Expected Outcomes and Sustainability

Expected outcomes:

- increased accessibility to support (including for remote students)
- reduced waiting times and better support continuity
- better use of teaching assistants' time (higher-value interventions)
- reusable model adaptable to other contexts

Sustainability:

- the in-house platform reduces dependency on an external interface and facilitates maintenance
- technical documentation and user guide support continuity beyond grant funding

Horizon 2: STT-4230 / STT-6230 - Programming for Data Science: Development of the New Course (Summer 2026)

Context and Pedagogical Intent

In summer 2026, I am developing a new version of STT-4230/6230 Programming for Data Science. This redesign aims to align the course with current practice: students should not only write code that runs, but learn to design, structure, test, optimize, document, and deliver reproducible artifacts in a collaborative setting. The course is explicitly designed for students who already have R experience, so class time can focus on intermediate-to-advanced competencies: project engineering, performance, software architecture, and full model lifecycle.

Course Vision: Moving from "Script" to "Product"

The redesign is guided by a progressive transition:

- from one-off scripts to structured projects
- from local results to shareable deliverables (reports, dashboards, web apps)
- from individual code to collaborative code (versioning, review, documentation)
- from "I can run something" to "I can deploy and maintain a pipeline"

The objective is that students leave the course able to deliver a mini data-science product: a clean repository, a clear pipeline, reproducible analyses, and the ability to diagnose, maintain, and evolve the project.

Modular Architecture (Planned Progression)

The course is organized in modules following a real production logic, where each block enables the next.

1. Structuring data-science projects
    - directory conventions, file organization, reproducibility principles
    - dependency and environment management
    - versioning best practices and commit strategy
2. Data handling and management
    - import, validation, cleaning
    - robust and traceable transformations
    - output structuring (clean tables, exchange formats)
3. Visualization
    - exploratory visualization and communication
    - principles of readability, coherence, and graphical honesty
    - graphics as components of a reproducible pipeline

4. Reports, web apps, and dashboards
    - reproducible reports (generation, parameters, templates)
    - decision-oriented dashboards
    - web apps for interaction and dissemination
5. Object-oriented programming
    - motivation in data science (stable interfaces, rich objects, methods)
    - object design for analytical workflows
    - user-oriented design: prints, summaries, diagnostics
6. Functional programming with tidyverse and pipelines
    - transformation composition
    - readable, testable, reusable pipelines
    - "data in, data out" principles and pure functions when possible
7. Profiling, optimization, and performance management
    - identify bottlenecks
    - optimize without harming readability
    - tradeoffs: performance, maintainability, reproducibility
8. Asynchronous and parallel programming
    - when parallelization is relevant
    - structuring parallelizable tasks, avoiding common errors
    - robustness: error handling, logs, result reproducibility
9. End-to-end model pipeline and hyperparameters
    - structuring full machine-learning pipelines
    - train/validation/test separation
    - hyperparameter tuning and data-leakage prevention
10. Model deployment and production

- turning a model into a service or reusable component
- producing consumable outputs (API, report, app)
- minimum requirements: documentation, tests, versioning

11. Model lifecycle

- continuous monitoring: drift, degradation, updates
- logging, monitoring, and quality maintenance
- reflection on model governance and responsibility

12. Collaboration and documentation: package

- turning a project into a reusable package
- user and developer documentation
- team collaboration: issues, code review, conventions, continuous integration

Assessments and Deliverables (Orientation)

Assessments will be aligned with delivery competencies:

- recurring deliverables as structured, reproducible repositories
- artifact production: report, dashboard, or web app
- software components: objects, functions, pipeline, tests, documentation
- final integrative project: a complete mini-product (data -> pipeline -> deliverable -> lightweight deployment), with collaboration and traceability requirements

Expected Outcomes

This redesign aims to produce a clear increase in autonomy and programming maturity: less trial-and-error, more intentional design; fewer fragile scripts, more maintainable projects; and a concrete understanding of what it means to put analysis, models, and code into production in real environments.

Horizon 3: Pedagogical Research (SoTL)

I plan to formalize my approach through academic publication on university pedagogy.

- STT-1100 analysis: an article documenting course design is in preparation. A longitudinal study on skill retention is also envisioned to evaluate whether students immersed in the STT-1100 narrative retain better programming habits throughout their program.
- Competition club: launch of a student club for Kaggle/hackathon competitions, which will also serve as an observation site for informal and peer-based learning dynamics.

Conclusion: A Pedagogy of "Doing Together"

Across these projects, my trajectory remains guided by one conviction: teaching statistics is not the transmission of inert knowledge, but an invitation to join a living, tool-enabled, and ethical community of practice.

Student Support, Accessibility, DEI (Diversity, Equity, Inclusion)

My commitment to student success goes beyond my own courses and is embodied in major departmental structures that I lead.

Centre de Dépannage et d'Apprentissage (CDA)

The CDA is the core support system for mathematics and statistics at Universite Laval. As coordinator, I manage a team of about 30 teaching assistants who provide daily support to students from 32 different programs.

A Large-Scale Support Ecosystem

This service represents approximately 3,000 visits per semester. To maintain consistent quality at this scale, I implemented a professionalization project (funded by PAIP) including:

- Structured training: onboarding pathway for new teaching assistants focused not only on mathematical content but on pedagogical approach (how to guide without giving the answer).
- Teaching assistant hub: an internal platform for sharing resources and best practices.
- Hybrid innovation: GPT-CDA deployment helps offload repetitive questions so assistants can focus on complex conceptual support.

Inclusion and Diversity: Concrete Actions

Welcome and Integration

Each year, I run a dedicated welcome session for new students with an immigration background. This session aims to demystify the Quebec university system and present available resources (CDA, tutoring) to support fast integration and reduce isolation.

Universal Accessibility

A major barrier to inclusion is the cost of pedagogical materials. All my courses (STT-1100, STT-4230, etc.) are built on open educational resources (OER):

- No expensive mandatory textbook.
- Course notes available as a Quarto website, accessible on mobile and tablet.
- Exclusive use of free software (R, Python, Git) so students can work on their own devices without paid licenses.

Ethics and AI: An Inclusive Approach

The rise of generative AI may widen gaps between those who know how to use it and those who do not (or cannot pay for "Pro" versions).

- Democratized access: by deploying tablets with GPT-CDA at the support center, I ensure equitable tool access for all students regardless of personal equipment.
- Ethical guidance: in my courses, AI use is framed by clear guidelines on data protection and transparency, turning a potential source of anxiety ("Am I allowed?") into an explicit professional skill.