Course Syllabus

Data Visualization and Storytelling with AI

R + Web Programming Track — No Programming Background Required

Instructor: Dr. Vikas Thammanna Gowda Semester: Fall 2025
Contact: vthammannagowda@champlain.edu Office Location: West Hall 100
Office Hours: TBD

1 Course Overview

This course introduces students to the principles and practice of data visualization and visual storytelling in an era augmented by large language models (LLMs) and artificial intelligence. Students learn to explore, clean, and visualize real-world data using `R` (tidyverse and `ggplot2`), build interactive dashboards with `React`, and craft narratives. The course emphasizes responsible AI usage, prompt design, and critical evaluation of automated insights. Projects culminate in a portfolio of explainable, interactive visualization stories.

2 Prerequisites and Audience

- **Prerequisite:** 30 credits (general academic maturity). No prior programming experience required.
- **Target students:** Undergraduates from any discipline who want to analyze and communicate data with visual clarity, using R and AI-assisted workflows.

3 Learning Outcomes

By the end of the course, students will be able to:

1. Explain core visualization principles and map them to `ggplot2` constructs.
2. Load, clean, and summarize data using the `tidyverse` in R.
3. Construct effective static and interactive visualizations with `ggplot2` and `plotly`.
4. Design and refine prompts to obtain, debug, and explain R code from LLMs, validating outputs critically.
5. Evaluate visualizations for clarity, accessibility, bias, and ethical concerns.
6. Compose coherent narrative reports embedding visual artifacts via `rmarkdown`.
7. Build simple reactive dashboards using `React`.
8. Reflect on the role of AI/LLMs in their analytic process and document compromises, corrections, and trust decisions.

4 Course Components and Assessment

4.1 Grading Policy

Component	Weight
LLM-assisted case studies & Prompt engineering portfolio	50%
Mini-Project	15%
Final Project: Visualization story on Web	25%
Participation & Attendance	10%

4.2 Grading Scale

Score Range (%)	Grade
≥ 93.00	A
90.00–92.99	A-
87.00–89.99	B+
83.00–86.99	B
80.00–82.99	B-
77.00–79.99	C+
73.00–76.99	C
70.00–72.99	C-
67.00–69.99	D+
63.00–66.99	D
60.00–62.99	D-
≤ 59.99	F

4.3 Grading Notes

- Emphasis is on understanding, iteration, and explanation—not just polished output.
- AI usage must be documented: include original prompts, model outputs, and student edits with rationale.
- Late submission policy: Applicable for **LLM-assisted case studies** only. 50% penalty for late submissions within 1 week; no credit after 1 week unless prior arrangements made.
- Participation includes engagement in discussions, peer reviews, and feedback sessions.

5 Tools and Technologies

- **R / tidyverse:** data import, manipulation, summarization.
- **ggplot2:** declarative static visualization.
- **plotly (via ggplotly):** interactivity layered on ggplot2 outputs.
- **React:** web dashboard building.
- **rmarkdown:** narrative reporting (HTML/PDF).
- **LLM integration:** Students will use LLMs (e.g., ChatGPT or institution-provided models) as collaborators—generating, explaining, and critiquing R code and insights.
- **Environment:** `RStudio Cloud` / `posit.cloud` preferred for zero-install. Starter project templates provided with `renv` lockfile.

6 AI/LLM Usage Guidelines

Responsible AI Collaboration

- Use LLMs as assistants, not oracles: always verify computed statistics manually or with independent code.
- Include the prompt you used, the raw LLM response, your modifications, and a brief commentary for every significant LLM-assisted artifact.
- Ask the model to explain code line-by-line, then cross-check at least one computed value yourself.
- Be wary of hallucinated "insights": if a model claims a trend, validate it using the data.
- Disclose in final project where and how AI contributed (e.g., "The caption was drafted by an LLM then edited for domain accuracy.")

7 Weekly Schedule (Tentative)

Week	Topics / Activities
1	**Onboarding & Data Literacy**: types of data, basic summary statistics, introduction to RStudio Cloud, primer on asking LLMs effective questions, setup project template, "AI teammate" orientation.
2	**Fundamentals of Visualization**: perception, visual encoding, common chart types; critique misleading visuals; mini-lab: revise a bad chart using `ggplot2`.
3–4	**Introductory R for Visualization**: tidyverse basics (`filter`, `mutate`, `summarise`, joins), ggplot2 grammar, constructing bar/line/scatter plots, using LLMs to scaffold code generation and explanation.
5	**Designing Effective Visuals**: color theory (accessibility), labels/annotations, narrative flow, small multiples; peer review exercise.
6-7	**Ethics, Bias, and Trust**: identifying bias in data and visual encoding, misleading AI narratives, privacy concerns, responsible disclosure of AI assistance.
8-12	**Implementing Visualization on Web**: build interactive web-based data visualizations with modern JavaScript tooling (React, Recharts, CSV parsing). critically engaging with LLM-assisted workflows. live demo including peer interview feedback.
13–15	**Domain Application & Capstone Planning**: choose domain, data sourcing/cleaning plan, draft visual story arc, peer feedback, reflect on AI usage.

8 Major Assignments and Deliverables

8.1 LLM-Assisted Case Studies & Prompt engineering portfolio (50%)

Module Description: In this sequence of case studies, students learn core principles of visualization (perception, encoding, design), practical R-based implementation, and critical topics like ethics and bias—all through realistic, data-driven scenarios. Each case presents a dataset and a use case; students work through guided activities to produce visualizations that fulfill analytic requirements and collectively build a coherent data story. LLMs (e.g., ChatGPT) are used as collaborators for ideation, code scaffolding, and critique, with structured documentation and verification required. The module emphasizes iterative refinement: from exploratory progress reports to polished live demos and written narrative reports that explain and justify design and insight.

8.1.1 Learning Objectives

- Apply visualization fundamentals to real-world data problems.
- Implement and refine visualizations in R (e.g., `ggplot2`, faceting, smoothing) to reveal relationships and patterns.
- Detect and mitigate biases in data representation and storytelling.
- Integrate and critically assess LLM-generated suggestions, validating outputs against the data.
- Communicate insights through narrative, interactive demos, and reflective documentation.

8.1.2 Student Activities

- Exploratory analysis and incremental visualization prototyping with LLM-assisted prompt engineering.
- Iterative refinement of code and design informed by peer feedback and model critiques.
- Building a coherent data story that connects multiple views and addresses the use case scenario.
- Documenting LLM interactions: initial prompts, raw outputs, edits, and verification steps.
- Reflecting on how AI involvement shaped understanding and application of course concepts.

8.1.3 Deliverables

1. **Progress report:** interim findings, initial visualizations, and evolution of prompts/code.
2. **Prompt Engineering Portfolio:** comprehensive log of AI/LLM usage including original prompts, raw outputs, refinements, validation steps, and a reflective narrative describing the student's experience with AI—how it helped (or misled) their understanding and application of visualization concepts in a realistic context.
3. **Live presentation/demo:** interactive or static dashboard with storytelling rationale.
4. **Written narrative report:** explanation of key visualizations, insights, design decisions, and responsible AI usage (including what was validated or corrected).

8.1.4 Assessment Snapshot

- **Visualization quality:** correctness, appropriateness, and clarity of visual encodings.
- **Insight depth:** handling of confounders, bias identification, and meaningful interpretation.
- **LLM documentation:** completeness and critical engagement in the Prompt Engineering Portfolio, including verification of model outputs.
- **Communication:** effectiveness of live demo and written storytelling.
- **Reflection:** awareness of model limitations, lessons learned, and responsible AI use.

8.1.5 Workflow Expectations

1. Receive dataset and scenario.
2. Explore data and prototype visualizations, iteratively refining with LLM assistance while documenting prompts and changes.
3. Incorporate peer feedback focused on design and ethics.
4. Submit progress report and Prompt Engineering Portfolio.
5. Finalize dashboard and narrative report.
6. Present live and submit final reflection on AI usage.

8.2 Mini Project (15%)

Project Description: Students will collect two real-world visualizations (e.g., screenshots from news, social media, reports) and critically analyze their design, interpretability, and underlying data choices. The goal is to reverse-engineer the visualization: identify variables, chart type, evaluate clarity and potential misleading encodings, and formulate a substantive inquiry question for the original creator. The analysis is summarized on a single slide and communicated via a live 8–10 minute presentation.

8.2.1 Learning Objectives

- Deconstruct real-world visualizations to identify data dimensions and structural encoding choices.
- Evaluate design effectiveness and potential sources of confusion or bias in visual communication.
- Formulate insightful questions about data provenance, aggregation, and design rationale.
- Synthesize critique into a concise visual artifact (single slide) and deliver a coherent oral presentation.
- Practice timing, audience engagement, and clear verbal explanation of analytic reasoning.

8.2.2 Student Activity

1. Locate and capture two distinct existing visualizations (do not create your own).
2. For each visualization:
   - Describe the variables being visualized.
   - Identify the chart type or form.
   - Evaluate interpretability: what design elements help or hinder comprehension.
   - Pose one thoughtful inquiry question for the creator (e.g., about aggregation, encoding choices, intended audience).
3. Assemble both analyses on a single slide with clear structure.
4. Practice and deliver a live 8–10 minute presentation explaining your critique and reasoning.
5. Attend peers' presentations for full engagement credit.

8.2.3 Deliverables

1. **Single-slide analysis** (`.pptx`): contains both visualization critiques, each with variables, chart identification, interpretability evaluation, and inquiry question.
2. **Live presentation/demo:** 8–10 minute oral presentation of your slide, articulating your analysis and insights.
3. **Engagement:** Attendance and attention during peer presentations (affects participation/engagement score).

8.2.4 Workflow Expectations

1. Identify and capture two real-world visualizations early in the project window.
2. Perform detailed analysis for each, documenting variables, chart type, interpretability, and crafting an inquiry question.
3. Design and assemble the single-slide summary; rehearse the 8–10 minute presentation.
4. Submit the slide by the deadline and deliver the live presentation in class.
5. Attend peer presentations; participation contributes to engagement scoring.

Presentation Length Policy: Presentations must be between 8 and 10 minutes. If a presentation is shorter than 10 minutes, apply a penalty of 10% of the presentation component per full minute below 10.
Example: An 8-minute presentation incurs a 20% deduction on the 10 points allotted for presentation clarity/pacing (a 2-point reduction).

8.3 Final Project: Assessing the Sensitivity of a Physical Sensor

Note: This is just a sample project description. The actual project may vary based on course needs and availability of resources.

Project Description This project combines physical sensing with web-based visualization to give students end-to-end IoT data experience. Students will assemble and calibrate a DHT22 temperature/humidity sensor with a Raspberry Pi, collect data under varying environmental conditions to assess sensor sensitivity, consolidate and clean the data, and build an interactive React dashboard to visualize and interpret the results.

8.3.1 Learning Objectives

- Integrate and interface hardware: connect a DHT22 sensor and I2C LCD to a Raspberry Pi and read real-time environmental data using Python.
- Log, consolidate, and clean time-series sensor data from multiple experimental conditions.
- Set up a modern web visualization environment (React + Recharts + PapaParse) and load CSV data for interactive display.
- Design and implement clear, informative dashboards applying visualization best practices (labels, tooltips, legends, multi-condition encoding).
- Interpret sensor behavior: assess responsiveness, sensitivity, and potential sources of error across conditions.
- Communicate findings through demonstration and written report, and reflect on the end-to-end data workflow.

8.3.2 Student Activity

1. **Hardware Assembly & Testing:** Wire the DHT22 and I2C LCD to the Raspberry Pi, install required libraries, and verify live readings via a Python script (with optional unit conversion extension).
2. **Experimental Design & Data Collection:** Define at least two environmental conditions (e.g., ambient vs. heated), run the sensor to collect timestamped temperature and humidity data, and log it responsibly (adhering to sensor read interval limits).
3. **Data Consolidation:** Merge, clean, and label the collected datasets into a unified CSV for downstream analysis.

4. **Dashboard Development:** Scaffold a React application, integrate PapaParse to load the consolidated CSV, and build interactive visualizations (e.g., time series for temperature and humidity, condition comparisons) using Recharts.
5. **Interpretation:** Analyze how sensor readings vary across conditions, noting delays, magnitude of change, and anomalies.
6. **Preparation & Presentation:** Prepare code, visual assets, and narrative to demonstrate both the physical setup and the analytical dashboard.

8.3.3 Deliverables

1. **Hardware Demonstration:** Live lab demo or short video showing the working sensor/LCD pipeline with real-time readings.
2. **Data Files:** Raw and consolidated CSV logs of sensor readings with appropriate condition labeling and timestamps.
3. **Source Code:** Python script for data collection and the full React dashboard codebase (organized, commented, and shareable via archive or repository).
4. **Dashboard Access:** Deployed or runnable interactive dashboard (plus fallback screenshots if live access is not feasible).
5. **Project Report:** 2–4 page PDF including introduction, experimental setup, at least one visualization with explanation, analysis of sensor sensitivity across conditions, discussion of uncertainty/errors, and conclusions/suggestions.
6. **Prompt Engineering Portfolio:** comprehensive log of AI/LLM usage including original prompts, raw outputs, refinements, validation steps, and a reflective narrative describing the student's experience with AI—how it helped (or misled) their understanding and application of visualization concepts in a realistic context.

8.3.4 Workflow Expectations

1. **Stage 1:** Assemble hardware, install libraries, and validate sensor + display pipeline with Python.
2. **Stage 2:** Design and execute experiments under multiple conditions; collect and log time-series data.
3. **Stage 3:** Consolidate and clean the collected data; initialize React dashboard environment and verify data ingestion.
4. **Stage 4:** Build, polish, and test visualizations; interpret findings; finalize dashboard and supporting artifacts.
5. **Finalization:** Prepare and submit all deliverables, and perform the required demonstration/presentation.

9 Collaboration and Academic Integrity

- Students may discuss approaches and give peer feedback, but code submissions and reflections must be their own.
- AI assistance is allowed and expected, but must be transparently documented. Copy-pasting LLM output without comprehension or attribution counts as academic dishonesty.
- Collaboration on capstone is individual unless group option is explicitly approved; in group cases, responsibilities must be delineated.
- Use of LLM's/AI to generate reports is strictly prohibited. Any such use will be considered a violation of academic integrity.
- All work must adhere to Champlain College's academic integrity policy; violations will result in disciplinary action.

10 Resources

- **Books / References:**
  - Hadley Wickham and Garrett Grolemund, *R for Data Science* (online freely available).
  - Claus O. Wilke, *Fundamentals of Data Visualization*.
  - `shiny` tutorials from Posit (`https://posit.co` recommended as central hub).
- **Cheat Sheets:** tidyverse, ggplot2, prompt engineering templates (provided as PDFs/in-notebook quick reference).
- **LLM Prompt Guide:** curated list of example prompts for R tasks, debugging, critique, and narrative generation.
- **Starter Code:** Versioned RMarkdown templates with sections: Data Import, Cleaning, Visualization, AI Interaction Log, Reflection.
- **Website Starter:** Students get a bootstrapped React starter project preconfigured for local development, including example CSV loading, basic dashboard layout, Tailwind styling, and utility components for filters, loading/error states, and responsive design. The package also includes setup instructions, scripts to run the dev server, LLM integration hooks for logging prompt-assisted insights, and tips.

11 Accessibility & Inclusion

- All visualizations must include descriptive text/alt descriptions in reports.
- Provide high-contrast themes and colorblind-safe palette recommendations (e.g., via `scale_fill_brewer()` with appropriate palettes).
- Multiple submission modalities for final presentations (written summary, narrated video, interactive link).
- Students with accommodations should notify the instructor early to coordinate adjustments.

12 Getting Help

- Office hours (TBD)—drop-in for debugging, prompt refinement help, and project feedback.
- Peer review sessions: scheduled mid-course to exchange feedback using a structured rubric.
- Discussion board / Slack/Canvas: questions, sharing interesting AI prompt results, and showcasing interim visuals.
- "AI Coach" handout available: tips on verification, avoiding overreliance, and asking effective follow-ups.

13 Tentative Schedule of Deliverables

Week	Deliverable
Week 2	Visualization critique: Case Study 1
Week 4	Basic plotting: Case Study 2
Week 5	Mini Project
Week 7	Design improvement: Case Study 3
Week 10	Progress report: Case Study 4
Week 11	Live demo + Final Report: Case Study 4
Week 15	Final project report
Finals Week	Final project presentation

14 Instructor Contact & Communication Protocol

- Preferred communication: institutional email; include course name and brief subject in subject line.
- Expect replies within 48 hours on weekdays.
- For quick debugging: include minimal reproducible example, expected vs actual behavior, and any LLM prompt + response if AI-assisted.

15 Optional Extensions (for motivated students)

- Time-series forecasting visualizations.
- Geospatial mapping (e.g., with `leaflet` in R).
- Automated narrative generation combined with data triggers (scheduled report summaries).
- Incorporating real-time streams or APIs into Shiny dashboards.

This syllabus may be refined slightly during the course of the semester to accommodate class pace and student interests. Significant changes will be communicated in writing.