My title[*]

My subtitle if needed

Allyson Cui[†] Rohan Alexander[‡]

2025-06-30

Abstract

First sentence. Second sentence. Third sentence. Fourth sentence.

Introduction

The increasing adoption of large language models (LLMs) for data analysis presents a new frontier in the study of analytical variation—a topic long explored through many-analysts studies in the social sciences. Prior research has established that different human analysts, when presented with the same dataset and research question, often produce markedly different results due to differences in data cleaning, preprocessing, and interpretation (Silberzahn et al., 2018; Botvinik-Nezer et al., 2020). Recent extensions of this work have begun to explore whether LLMs, when tasked with identical analytical workflows, replicate this variability, thereby raising concerns about the reproducibility of AI-assisted research (Zhang et al., 2024; Huntington-Klein et al., 2025).

This paper introduces a benchmark designed to systematically evaluate variation across LLMs in applied data science tasks. We draw on a novel dataset of appointments made by the Government of New Brunswick from 2013 to 2024, curated by the Investigative Journalism Foundation. Our empirical question—"Which government branch most frequently reappoints past appointees, and is this trend increasing or declining?"—requires a multi-step data analysis process that includes combining datasets, identifying reappointments, calculating organization-level rates, and fitting regression models on temporal trends.

[Might change later] We test three leading LLMs—Claude Sonnet 4, Claude Opus 4, and GPT-4o—on two formally specified benchmark documents that detail step-by-step procedures for this task. Each model is evaluated on both protocols to assess how consistently they follow structured instructions, how they interpret ambiguous data elements, and whether their final outputs converge or diverge. This design allows us to isolate sources of LLM variation and quantify the extent to which different models reach comparable conclusions when given the same analytical blueprint.

By adapting the many-analysts paradigm to LLMs and focusing on a real-world administrative dataset, our study finds [conclusion on accuracy, replicability, and other limitations].

Literature Review

The credibility of empirical research in the social and behavioral sciences has faced increasing scrutiny over the past decade, giving rise to innovative methodological approaches designed to examine the reliability and robustness of scientific findings. One particularly revealing approach - the "many-analysts" paradigm - has illuminated how researcher variability can dramatically influence analytical outcomes, even when working with identical datasets and research questions. This methodology, where multiple independent analysis teams

[*] Code and data are available at: https://github.com/AllysonCui/LLM-Automation.

[†] University of Toronto. Work done during an internship at the Investigative Journalism Foundation.

[‡] University of Toronto, rohan.alexander@utoronto.ca.

tackle the same research task, has uncovered substantial variation in results that challenges traditional notions of scientific objectivity and highlights the role of subjective decision-making throughout the research process.

Silberzahn et al. (2018) conducted a pioneering many-analysts study that invited 29 research teams to analyze the same dataset, investigating whether soccer referees disproportionately give red cards to dark-skinned players. Despite working with identical data, teams reported effect sizes ranging from 0.89 to 2.93 in odds ratio units, with 20 teams finding a statistically significant effect and 9 finding no significant relationship. This striking disparity demonstrated that equally defensible analytic choices could yield substantially different conclusions—raising fundamental questions about the certainty with which any single analysis should be interpreted. Similarly, Botvinik-Nezer et al. (2020) showed that when 70 independent teams analyzed the same functional neuroimaging dataset, no two teams used identical workflows, and the teams reached varying conclusions about which brain regions showed significant task activation.

Sources of Analytical Variation

Several sources of researcher variation have been identified across many-analysts studies. Auspurg and Brüderl (2021) reanalyzed the Silberzahn et al. (2018) data and identified at least four different ways researchers interpreted the same research question, leading to fundamentally different analytical approaches. They argued that the resulting variation stemmed not just from subjective choices within a shared analytical framework, but from researchers answering entirely different questions. When they performed a "multiverse analysis" (systematic exploration of many reasonable specifications) with a clear research question, they found a much narrower range of effects, suggesting that precise question formulation significantly constrains analytical variability.

Most critically for understanding data cleaning as a source of variation, Huntington-Klein et al. (2025) employed an innovative three-stage design where 146 research teams completed the same task three times: first with minimal constraints, then with a specified research design, and finally with pre-cleaned data. This approach isolated different sources of variation, revealing that while substantial variation existed even under tight constraints, data cleaning decisions particularly influenced results. Interestingly, specifying the research design improved agreement on sample sizes but not on effect estimates—suggesting that researchers' interpretation and implementation of even well-specified designs varies considerably. Only when pre-cleaned data was provided did variation decrease substantially (to an IQR of 2.4 percentage points from 4.0), highlighting data preparation as a crucial source of researcher variation.

Breznau et al. (2021) further demonstrated this "hidden universe of uncertainty" in a study where 73 research teams tested the same hypothesis using identical cross-national survey data, producing widely divergent results and identifying 1,261 unique analytical decisions with no two teams making identical choices. Menkveld et al. (2024) found similar patterns when 164 research teams analyzed financial market data, with the greatest divergence occurring for the most complex research questions. Kummerfeld and Jones (2023) conceptualized this problem in terms of three critical pitfalls: lack of an actionable research question, failure to explicitly identify a formal question, and insufficient relevant expertise. Gould et al. (2023) demonstrated similar patterns in ecology and evolutionary biology, while Trübutschek et al. (2023) found remarkable variation in EEG preprocessing pipelines even for standard tasks.

Extending Many-Analysts to Artificial Intelligence

Given that data cleaning and preprocessing decisions emerge as such critical sources of variation among human analysts, the increasing use of large language models (LLMs) in data analysis raises important questions about whether AI systems exhibit similar patterns of variation. As LLMs become increasingly capable of performing data analysis, understanding whether their approaches to data cleaning and analysis are replicable and reliable becomes crucial for the future of empirical research.

Initial evidence suggests that different LLMs exhibit distinct analytical strengths and weaknesses, indicating that the choice of AI system could introduce variation analogous to human analyst differences. Agarwal et al. (2024) evaluated multiple LLMs across diverse tasks commonly performed by undergraduate computer science students, finding that no single LLM excelled across all task types. Different models showed distinct strengths: Microsoft Copilot performed best for code explanation, GitHub Copilot Chat excelled at programming tasks, ChatGPT dominated email writing, and Google Bard was superior for learning new concepts. This finding that different LLMs excel at different aspects of analytical workflow suggests that the choice of AI analyst could influence research outcomes in ways that parallel human analyst variation.

The complexity of this variation becomes more apparent when examining specific analytical tasks. Nejjar et al. (2025) evaluated LLMs for code generation and data analysis tasks commonly performed in scientific research, finding that while basic programming tasks showed relatively consistent performance across models, complex data analysis tasks requiring iterative refinement and contextual understanding yielded highly diverse results. This pattern mirrors the findings from human analyst studies, where more complex analytical decisions produce greater variation.

Several specific sources of LLM analytical variation have been identified. Zhang et al. (2024) systematically evaluated how LLMs handle data preprocessing tasks, examining models including GPT-4 and GPT-4o across error detection, data imputation, schema matching, and entity matching tasks. While GPT-4 achieved perfect accuracy (100%) on 4 out of 12 datasets, suggesting considerable potential for LLMs in data preparation workflows, their framework also highlighted critical limitations regarding computational expense and domain specification for specialized fields. Importantly, LLMs sometimes generate text that is plausible-sounding but factually incorrect, as they lack fundamental understanding and rely solely on learned patterns.

The preprocessing stage represents a particularly important source of variation. Qi et al. (2024) demonstrated that even when provided with standardized data cleaning APIs, LLMs make varying decisions about data standardization, including how to handle missing values, which column types to apply specific cleaning functions to, and how to interpret ambiguous data formats. Their CleanAgent framework revealed that LLMs can make different interpretations of data cleaning requirements, leading to different preprocessing outcomes.

Prompt sensitivity adds another layer of variation. Sclar et al. (2023) found that widely used open-source LLMs are extremely sensitive to subtle changes in prompt formatting in few-shot settings, with performance differences of up to 76 accuracy points when evaluated using LLaMA-2-13B. This sensitivity persists even when increasing model size, the number of few-shot examples, or performing instruction tuning.

Perhaps most significantly, LLMs introduce a novel source of variation through their dynamic analytical planning capabilities. Hong et al. (2024) demonstrated through their Data Interpreter framework that LLMs construct hierarchical graph models where analytical tasks are decomposed into interconnected subtasks, with the model dynamically adjusting both task graphs and action graphs based on intermediate results. This dynamic planning capability means that two identical requests to the same LLM can result in entirely different analytical workflows, introducing a source of variation that is largely absent in human analysts who typically commit to a specific analytical approach early in their process.

These findings suggest that as human analysts increasingly use LLMs as research tools in their empirical research workflows, systematic differences between LLMs combined with different prompts could significantly influence research conclusions and hinder scientific reproducibility. Understanding these patterns of AI analytical variation becomes essential for maintaining research quality.

Benchmark to Evaluate Analytical Variation

The extension of many-analysts paradigms to LLMs necessitates robust benchmarks that can systematically evaluate how different AI systems approach complex data science workflows, particularly focusing on data preprocessing and cleaning decisions that have been identified as critical sources of variation in human analyst studies.

Some developments in data-centric benchmarks evaluate LLMs' capabilities in the preprocessing stages that precede analysis. Mazumder et al. (2023) developed DataPerf, a comprehensive benchmark suite specifically designed to evaluate data-centric AI operations across multiple domains. DataPerf includes challenges for training set selection, data cleaning and debugging, data acquisition, and quality assessment—precisely the types of decisions that drive variation in many-analysts studies. The benchmark's data debugging challenge requires AI systems to identify and prioritize the most detrimental data points in noisy training sets, achieving accuracy improvements through strategic data cleaning.

Building on code generation benchmarks, recent work has focused on end-to-end data analysis capabilities. DataSciBench provides a comprehensive evaluation framework for LLMs in data science tasks, utilizing a semi-automated pipeline to generate ground truth through LLM-based self-consistency strategies. The benchmark employs a Task-Function-Code (TFC) evaluation framework with carefully crafted complex questions and predefined aggregated metrics. Similarly, InfiAgent-DABench introduced 311 data analysis questions derived from 55 CSV files, specifically designed to evaluate LLM-based agents in data analysis tasks using format-prompting techniques to ensure closed-form questions that can be automatically evaluated.

For evaluating machine learning experimentation workflows, MLAgentBench introduced 13 tasks ranging from improving model performance on CIFAR-10 to recent research problems, evaluating agents' ability to perform actions like reading/writing files, executing code, and inspecting outputs throughout the ML experimentation process. DSEval provides an evaluation paradigm for assessing data science agents throughout the entire data science lifecycle, incorporating a bootstrapped annotation method to streamline dataset preparation and expand benchmarking comprehensiveness.

Advanced scientific coding benchmarks demonstrate the challenges LLMs face with domain-specific analytical tasks. Tian et al. (2024) developed SciCode, containing 338 subproblems decomposed from 80 challenging scientific research problems across 16 natural science subfields. Problems naturally factorize into multiple subproblems involving knowledge recall, reasoning, and code synthesis.

Beyond data analysis, computational reproducibility benchmarks provide important foundations for evaluating LLMs in research contexts. Siegel et al. (2024) introduced CORE-Bench (Computational Reproducibility Agent Benchmark), consisting of 270 tasks derived from 90 scientific papers across computer science, social science, and medicine. This benchmark evaluates AI agents' ability to reproduce research results by installing dependencies, executing code, and extracting findings from outputs. The benchmark demonstrates that even basic computational reproducibility—a prerequisite for novel research—remains challenging for current AI systems, particularly due to dependency resolution issues and difficulty navigating complex file structures to extract relevant results.

While existing benchmarks like DataPerf (Mazumder et al., 2023), DataSciBench, and MLAgentBench have made significant advances in evaluating LLMs' data science capabilities, they primarily focus on either computational execution, isolated data operations, or specific coding tasks rather than end-to-end analytical workflows that require sequential decision-making throughout the complete data science pipeline. These benchmarks often evaluate discrete capabilities—such as data cleaning accuracy, code generation quality, or model performance optimization—but do not systematically examine how LLMs handle the types of ambiguous analytical decisions that drive the greatest variation in many-analysts studies among humans. Specifically, existing approaches do not adequately assess how different LLMs interpret research questions, make data preprocessing choices, handle missing or inconsistent data, and resolve analytical ambiguities when multiple defensible approaches exist. Our benchmark addresses this gap by presenting a multi-step analytical workflow that mirrors the complex decision-making processes where human analysts demonstrate the greatest variation: from initial data exploration and cleaning through variable definition, analysis design, and results interpretation. By requiring LLMs to navigate the same types of sequential, interdependent analytical choices that create divergent outcomes in human many-analysts studies—particularly around data cleaning and preprocessing decisions identified as critical sources of variation by Huntington-Klein et al. (2025)—our approach enables systematic measurement of the extent and sources of analytical variation across AI systems in realistic research contexts.

Such a benchmark should evaluate LLMs' ability to handle the types of ambiguous analytical decisions that drive variation in many-analysts studies: how to handle missing data, which variables to include in models, how to define and clean key variables, and how to interpret borderline statistical results. The benchmark tasks should be designed to have multiple defensible analytical paths, allowing different LLMs to make different choices while still producing valid analyses, thus enabling systematic measurement of the extent and sources of analytical variation across AI systems.

Benchmark: Evaluating Analytical Variation Across LLMs

Objective

This benchmark is designed to systematically evaluate how different large language models (LLMs) perform when given the same structured data science task, under identical instructions. Our central research question is: *Which government branch in New Brunswick most frequently reappoints past appointees, and is this trend increasing or declining over the past 12 years?*

We operationalize this through a 9-step analysis pipeline and observe whether distinct LLMs, when tasked with identical code-generation and analysis requests, produce consistent results.

Dataset

We use data on public appointments by the Government of New Brunswick from 2013 to 2024, curated by the Investigative Journalism Foundation. The dataset includes individual-level appointment records and contains 17 columns with 5 crucial ones that will be used in the analysis:

- `name`: full name of the appointee
- `position`: the official title of the appointment
- `org` (organization): the department or agency
- `reappointed`: whether the appointment is a reappointment
- `date`: the post date of the appointment

Each year's data is stored as a separate CSV file: `appointments_2013.csv` through `appointments_2024.csv`.

Analytical Workflow

The benchmark is structured around a nine-step analytical pipeline designed to evaluate LLMs' capacity to carry out a full data analysis from raw files to statistical inference. Each step builds logically on the previous one, providing intermediate outputs that serve both as checkpoints for correctness and as lenses into potential sources of variation. The steps are intentionally detailed to reflect a realistic, multi-stage workflow common in empirical research.

1. **Combine Annual Datasets (2013–2024)**: Load twelve raw CSV files, append a year column to each, and concatenate them into a unified dataset.

2. **Select Key Columns**: Extract and standardize the essential variables (name, position, org, reappointed, year) while resolving column name inconsistencies.

3. **Mark Reappointments**: Identify repeat appointments by detecting multiple instances of the same individual holding the same position at the same organization across years. Flag all but the first as reappointments.

4. **Count Total Appointments**: Compute the total number of appointments per organization per year to serve as denominators for subsequent rate calculations.

5. **Count Reappointments**: For each organization-year pair, count how many appointments are marked as reappointments.

6. **Compute Reappointment Rates**: Divide the number of reappointments by total appointments for each organization in each year to generate reappointment rates.

7. **Identify Max-Reappointing Organizations**: Determine which organization had the highest reappointment rate each year and produce a time series of these peak rates.

8. **Calculate Government-Wide Trends**: Compute the overall reappointment proportion across the entire government for each year from 2013 to 2024.

9. **Trend Analysis via Regression**: Perform linear regression to assess whether the government-wide reappointment proportion has increased or decreased over time, reporting slope, statistical significance, and diagnostic statistics.

Models and Conditions

Three LLMs are evaluated under two benchmark protocols: Claude Sonnet 4, Claude Opus 4, and GPT-4o. All prompts are given in the "project" mode of the AI platform.

There are two versions of prompt protocols to examine how robust LLMs are to prompts with different levels of detail. Each LLM is tasked with producing 9 Python steps to complete the 9-step workflow under each version. Twenty executions are tested per LLM per version.

Version 1 provides a rigorously modular, step-by-step framework. Each prompt corresponds to a discrete Python step that builds on the previous step—from raw data ingestion to final regression and visualization—allowing precise inspection of intermediate outputs and LLM decision-making. `appointments_2024.csv` is given as initial project knowledge before the prompts. Complete prompts are provided in the Appendix.

Version 2 offers a compact, instruction-driven pipeline using unified "Code Generating Guidance" to direct the LLM through a complete 9-step workflow. The Guidance embeds environmental setup rules (directory paths, file naming, libraries) and the prompts only have one sentence for each step without specific hints such as column variation or missing values. `appointments_2024.csv` and `Code Generation Guideline` are given as initial project knowledge before the prompts. Complete prompts are provided in the Appendix.

Version 3 provides a single, end-to-end prompt that instructs the model to complete the full 9-step analysis in one script, using only appointments_2024.csv as an example and relying on the code generating guideline to infer all remaining structure, file paths, and naming conventions. `appointments_2024.csv` and `Code Generation Guideline` are given as initial project knowledge before the prompts. Complete prompts are provided in the Appendix.

In addition to issuing structured prompts, our benchmark permits LLMs to iteratively revise their code in response to runtime errors. If a step fails due to a blocking error (e.g., `FileNotFoundError`, `KeyError`, `TypeError`), the error message is returned to the model, which is then prompted to generate a corrected version of the step. This refinement cycle continues until the code executes successfully without critical errors or until five consecutive attempts have failed. If the fifth attempt does not produce a working step, the dialogue is marked as a `fail`; otherwise, it is marked as a `success`. Non-blocking issues—such as stylistic formatting inconsistencies or warnings that do not interrupt execution—are not considered failures.

Evaluation Matrix

Our evaluation framework assesses LLM performance across four key dimensions: accuracy, reproducibility, interpretive variation, and output variation. Each metric is designed to capture distinct aspects of analytical consistency and reliability.

Accuracy Metrics **Average Maximum Organizations per Year Correct Rate**: For the result to be considered correct, for each year, both the "organization" and the "maximum reappointment rate" that the organization has must be correct. This metric ranges from 0 to 1.

Average Regression Slope Correct Rate: For the result to be considered correct, both the regression's "slope coefficient" and whether the regression has "statistical significance" must be correct. This metric ranges from 0 to 1.

Reproducibility Metrics **Average step Initial Success Rate**: The average rate of steps that are successful in the first attempt. If 160 steps out of 180 total steps (from 20 executions) are successful on the first attempt, the rate would be 160/180 = 0.889. This metric ranges from 0 to 1.

Average Execution Success Rate: The average proportion of executions that produced a final result, regardless of correctness or the number of initial errors. This metric ranges from 0 to 1.

Interpretive Variation Metrics **Number of Unique Reappointment Definitions**: Different models may employ varying definitions of reappointment. Some might use looser definitions such as "Match only on name, regardless of position/org," while others might use tighter definitions such as "Require an exact match including additional fields like appointment date or status." This metric ranges from 0 to infinity.

Number of Tie-Breaking Strategies: Count of distinct approaches used by models to resolve ties when multiple organizations share the same maximum reappointment rate. This metric ranges from 0 to infinity.

Output Variation Metrics **Average Final Correct Step**: The final correct step is the last step where the result is fully correct, including the correct number of rows/columns for CSV output and accurate calculations. This metric ranges from 0 to 9.

Standard Deviation of Slope Coefficient: Calculate the standard deviation of all slope coefficients from the 20 executions. This metric ranges from negative infinity to positive infinity.

Code Running Metrics **Time to Generate Full Pipeline**: For executions that succeed, measure the total time required to generate the complete analytical pipeline.

Average Script Length by Line: Calculate the average number of lines of scripts across 10 executions per model-version.

Debug Metric Average number of re-prompt attempts before a failed step executes successfully. Lower is better.

Version1:

Dimension	Metric	GPT-4o	Claude Opus 4	Claude Sonnet 4
Accuracy	1. Avg. max-orgs correct rate	0.60	0.30	0.70
	2. Avg. regression slope correct rate	0.60	0.60	0.60
Reproducibility	1. Avg. step initial success rate	0.87	0.91	0.90
	2. Avg. execution success rate	1.00	1.00	1.00
Interpretive Variation	1. # of unique reappointment definitions	7	6	6
	2. # of tie-breaking strategies	-	-	-
Output Variation	1. Avg. final correct step	-	-	-
	2. SD of slope coefficient	-	-	-
Code Running	1. Time to generate full pipeline (seconds)	11.36	11.37	12.77
	2. Average Script Length (line)	68.58	199.56	313.58
Debugging	Avg. # of retries per failed step	1.13	1.13	1.33

Version2:

Dimension	Metric	GPT-4o	Claude Opus 4	Claude Sonnet 4
Accuracy	1. Avg. max-orgs correct rate	0.70	-	-
	2. Avg. regression slope correct rate	0.60	-	-
Reproducibility	1. Avg. step initial success rate	0.99	-	-
	2. Avg. execution success rate	1.00	-	-
Interpretive Variation	1. # of unique reappointment definitions	5	-	-
	2. # of tie-breaking strategies	-	-	-

Dimension	Metric	GPT-4o	Claude Opus 4	Claude Sonnet 4
Output Variation	1. Avg. final correct step	-	-	-
	2. SD of slope coefficient	-	-	-
Code Running	1. Time to generate full pipeline (seconds)	8.47	-	-
	2. Average Script Length (line)	68.58	-	-
Debugging	Avg. # of retries per failed step	1.00	-	-

Version3:

Dimension	Metric	GPT-4o	Claude Opus 4	Claude Sonnet 4
Accuracy	1. Avg. max-orgs correct rate	0.00	-	-
	2. Avg. regression slope correct rate	0.00	-	-
Reproducibility	1. Avg. step initial success rate	1.00	-	-
	2. Avg. execution success rate	1.00	-	-
Interpretive Variation	1. # of unique reappointment definitions	4	-	-
	2. # of tie-breaking strategies	-	-	-
Output Variation	1. Avg. final correct step	-	-	-
	2. SD of slope coefficient	-	-	-
Code Running	1. Time to generate full pipeline (seconds)	3.11	-	-
	2. Average Script Length (line)	68.58	-	-
Debugging	Avg. # of retries per failed step	0.00	-	-