Evaluating Analytical Variation in Large Language Models: A Many-Analysts Approach to Data Science Tasks[*]

Evidence from New Brunswick Government Appointments Data

Allyson Cui[†] Rohan Alexander[‡]

2025-07-14

Abstract

As large language models (LLMs) increasingly assist in data analysis, understanding their analytical variation becomes crucial for research reproducibility. We adapt the many-analysts paradigm to evaluate how three leading LLMs—Claude Sonnet 4, Claude Opus 4, and GPT-4o—approach identical data science tasks. Using government appointment data from New Brunswick (2013-2024), we task models with identifying reappointment patterns across government branches. . . .

Introduction

The integration of large language models (LLMs) into data analysis workflows represents a paradigm shift in empirical research. As researchers increasingly rely on AI assistants for data cleaning, analysis, and interpretation, a critical question emerges: do these tools enhance or undermine the reproducibility of scientific findings?

This question gains urgency in light of the well-documented "many-analysts" phenomenon, where equally qualified human researchers analyzing identical datasets often reach substantially different conclusions. If LLMs exhibit similar analytical variation, their widespread adoption could amplify rather than mitigate reproducibility concerns in empirical research.

We address this gap by adapting the many-analysts paradigm to systematically evaluate analytical variation across LLMs. Our study examines how three leading models approach a realistic data science task: analyzing reappointment patterns in New Brunswick government appointments from 2013-2024. By providing identical data and instructions to each model, we isolate and quantify sources of analytical variation in AI-assisted research.

Our contributions are threefold. First, we develop a benchmark specifically designed to evaluate analytical variation in LLMs, focusing on the ambiguous decisions that drive divergence in human analyses. Second, we document substantial variation in how different LLMs approach data cleaning, variable definition, and statistical analysis—even under identical instructions. Third, we identify data preprocessing as the primary source of variation, mirroring patterns observed in human analyst studies.

The remainder of this paper proceeds as follows. Section 2 reviews the literature on analytical variation and existing LLM benchmarks. Section 3 presents our methodology, including the dataset, analytical pipeline, and evaluation framework. Section 4 reports results across accuracy, reproducibility, and variation metrics. Section 5 discusses implications for AI-assisted research, and Section 6 concludes.

[*] 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.

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 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.

Methodology

Research Design

We adapt the many-analysts paradigm to evaluate LLMs, asking: *Which government branch in New Brunswick most frequently reappoints past appointees, and is this trend increasing or declining?* This question requires data integration, variable construction, and statistical analysis—tasks where human analysts typically diverge.

Dataset

We use appointment records from the Government of New Brunswick (2013-2024), curated by the Investigative Journalism Foundation. Each year contains a separate CSV file with:

- `name`: Appointee's full name
- `position`: Official appointment title

- `org`: Department or agency
- `reappointed`: Reappointment flag
- `date`: Appointment date

The dataset contains natural ambiguities (inconsistent column names, missing values, unclear reappointment definitions) that mirror real-world analytical challenges.

Analytical Pipeline

We decompose the analysis into nine steps:

1. **Data Integration**: Combine 12 annual CSV files
2. **Variable Selection**: Extract and standardize key columns
3. **Reappointment Identification**: Flag repeat appointments
4. **Aggregation**: Count appointments by organization-year
5. **Rate Calculation**: Compute reappointment rates
6. **Maximum Identification**: Find highest-rate organization annually
7. **Trend Calculation**: Calculate government-wide rates over time
8. **Statistical Analysis**: Test for temporal trends via regression
9. **Visualization**: Create summary plots

Experimental Design

Model Selection

We selected three leading large language models representing the current state-of-the-art in code generation and data analysis capabilities. GPT-4o, developed by OpenAI, represents one of the most widely adopted models in research and industry, known for its strong performance across diverse analytical tasks and robust code generation abilities. Claude Opus 4 and Claude Sonnet 4, both from Anthropic's Claude 4 family, were chosen to examine within-family variation—Opus 4 as the most powerful model optimized for complex reasoning tasks, and Sonnet 4 as a more efficient model balancing capability with speed. This selection allows us to evaluate both between-provider variation (OpenAI vs. Anthropic) and within-provider variation (Opus vs. Sonnet), providing insights into whether analytical differences stem from fundamental architectural choices or model-specific training approaches. All three models have demonstrated strong capabilities in data science tasks and are actively used by researchers for computational analysis, making them representative of real-world AI-assisted research scenarios.

Prompt Conditions

Version 1 - Detailed Step-by-Step Instructions: This version provides the most comprehensive guidance, with each of the nine analytical steps presented as a separate, detailed prompt. Each prompt includes specific instructions about file handling, error management, data validation, and expected outputs. For example, the reappointment counting prompt explicitly instructs models to "include debugging output to show sample grouped data before creating the pivot table" and to "ensure year columns are handled as integers." This granular approach mirrors how a senior researcher might guide a junior analyst through a complex analysis, providing scaffolding at each decision point. The version includes only the target CSV file (`appointments_2024.csv`) as project knowledge, requiring models to infer the structure of the remaining files. This design tests whether detailed procedural instructions can standardize analytical approaches across different models. See Appendix A for the complete prompt.

Version 2 - Concise Instructions with Coding Guidelines: This version represents a middle ground, providing both a comprehensive "Code Generating Guideline" document and streamlined single-sentence prompts for each step. The guideline document establishes the analytical context, file conventions, required libraries, and data structure expectations upfront, while individual prompts are reduced to their essential elements (e.g., "Count the total number of employees for each 'org' in each year"). This approach tests whether establishing clear conventions and standards can maintain consistency while allowing models more interpretive freedom in implementation details. The key difference from Version 1 is the shift from procedural instructions to declarative specifications, examining whether models can translate high-level requirements into appropriate implementations when given adequate context. See Appendix B for the complete prompt.

Version 3 - Single Unified Prompt: This version provides maximum autonomy to the models by condensing the entire nine-step analysis into a single prompt. While it includes the same "Code Generating Guideline" as Version 2, all analytical steps are presented together in one request, requiring models to decompose the complex task independently. This design mirrors real-world scenarios where researchers might describe their analytical goals holistically rather than procedurally. The critical distinction from Version 2 is the removal of step-by-step scaffolding—models must determine the appropriate sequencing, intermediate outputs, and validation steps autonomously. This tests whether models can maintain analytical coherence and correctness when given minimal structural guidance, revealing their inherent biases and default approaches to complex analytical workflows. See Appendix C for the complete prompt.

Iterative Refinement

Models receive up to 5 attempts to correct execution errors, mimicking real-world debugging. We track both initial success and final outcomes.

Evaluation Framework

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.

Output Variation Metrics

- **Average Correct Step**: Count the number of steps that produce correct intermediate results out of the total nine steps. Take the average across the 10 executions. 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

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

Results

Table 1: LLM Benchmarking Results by Version

Dimension	**Metric**	**GPT-4o**	**Claude Opus 4**	**Claude Sonnet 4**
Panel A: Version 1				
Accuracy	1. Avg. max-orgs correct rate	0.60	0.30	0.50
	2. Avg. regression slope correct rate	0.40	0.50	0.10
Reproducibility	1. Avg. step initial success rate	0.87	0.91	0.90
	2. Avg. execution success rate	1.00	1.00	1.00
Output Variation	1. Avg. correct steps	-	-	-
	2. SD of slope coefficient	0.000436	0.000617	0.000592
Code Running	1. Time to generate full pipeline (s)	10.95	15.63	11.57
	2. Average Script Length (line)	68.58	199.56	313.58
Debugging	Avg. # of retries per failed step	1.33	1.33	1.14
Panel B: Version 2				
Accuracy	1. Avg. max-orgs correct rate	0.00	0.00	0.00
	2. Avg. regression slope correct rate	0.00	0.00	0.00
Reproducibility	1. Avg. step initial success rate	0.99	0.82	0.84
	2. Avg. execution success rate	1.00	1.00	1.00
Output Variation	1. Avg. correct steps	-	-	-
	2. SD of slope coefficient	0.011928	0.001643	0.001925
Code Running	1. Time to generate full pipeline (s)	8.42	23.13	19.59
	2. Average Script Length (line)	45.21	301.18	437.37
Debugging	Avg. # of retries per failed step	1.00	1.78	1.00
Panel C: Version 3				
Accuracy	1. Avg. max-orgs correct rate	0.00	0.00	0.00
	2. Avg. regression slope correct rate	0.00	0.00	0.00
Reproducibility	1. Avg. step initial success rate	1.00	0.96	1.00
	2. Avg. execution success rate	1.00	1.00	1.00
Output Variation	1. Avg. correct steps	-	-	-
	2. SD of slope coefficient	0.001000	0.001227	0.001242
Code Running	1. Time to generate full pipeline (s)	3.96	4.10	5.40
	2. Script Length (line)	171.3	432.5	513.1
Debugging	Avg. # of retries per failed step	0.00	1.33	0.00

Discussion

Limitations

Conclusion

References