LLM-Assisted Replication for Statistical Research in Social Science

So Kubota[1], Hiromu Yakura[2], Samuel Coavoux[3],
Sho Yamada[4], and Yuki Nakamura[5]

[1]Graduate School of Economics, Tohoku University

[2]Max Planck Institute for Human Development

[3]CREST, ENSAE, Institut Polytechnique de Paris

[4]Graduate School of Public Policy, The University of Tokyo

[5]Faculty of Engineering, The University of Tokyo

January 26, 2026

Abstract

The replication crisis, the failure of ambitious new scientific claims to be validated by further research, is one of the most pressing issue for social science. This is in part an incentive problem: replication is costly and less rewarded than "original" research. Large language models (LLMs) have accelerated scientific production by streamlining writing, coding, and reviewing. We propose to leverage these models to reduce the cost of replication. In this paper, we present a working prototype that replicates statistical analyses from social science papers and flags potential problems. Population science is particularly well-suited to automation because it relies on standard statistical models, shared public datasets, and uniform presentation formats such as regression tables and summary statistics. We introduce a method that iterates LLM-based text interpretation, code generation, execution, and discrepancy analysis, and we demonstrate its capabilities by reproducing key results from a seminal sociology paper. We also outline deployment scenarios including pre-submission checks, peer-review support, and meta-scientific audits, positioning AI verification as assistive infrastructure that strengthens research integrity.[1]

1. Introduction

Large language models (LLMs) are rapidly changing how scientists produce research. They now assist with literature screening, survey design, data cleaning, code generation, and even peer review (Gilardi et al., 2023; Chen et al., 2021; Du et al., 2025; Conroy, 2023; Hosseini et al., 2025). Agentic systems extend these capabilities by chaining retrieval, planning, and execution, enabling models to run analyses, generate figures, and iteratively revise outputs (Wang et al., 2024; Yao et al.,

[1] Code and replication outputs are available at https://github.com/kubotaso/AI_Social_Replication. We thank Chishio Furukawa for helpful comments. This work was supported by JST CREST Grant JPMJPR2364 and by a grant from Hi! PARIS C23HIP-PR04-ENSAE.

2023; Wang et al., 2025). Researchers have also begun using LLMs as simulated participants in experiments and as components in measurement workflows (Aher et al., 2023; Argyle et al., 2023; Brand et al., 2023; Park et al., 2023) despite a growing literature warning of large, unexplained biases and a low variance in LLM answers to survey (Boelaert et al., 2025); in the natural sciences, related systems can design experimental procedures and interface with lab automation (Bran et al., 2024). Recent work has even proposed "AI Scientists" that autonomously generate research ideas and draft papers (Lu et al., 2024).

This acceleration promises major productivity gains, but it also creates a verification bottleneck. If AI reduces the cost of producing papers faster than it reduces the cost of checking them, the supply of scientific claims may outpace what the research community can credibly evaluate, threatening scientific integrity (Ioannidis, 2005; Naudet et al., 2018), at a time when the peer review system is under dire stress. This asymmetry amplifies what is often framed as a "replication crisis" (Open Science Collaboration, 2015; Freese and Peterson, 2017), that is the failure of ambitious new scientific claims to be validated by further research.

Social science faces closely related problems, now documented across economics, political science, psychology, and sociology (Camerer et al., 2016, 2018; Brodeur et al., 2024b; Open Science Collaboration, 2015; Freese and Peterson, 2017). In psychology, large multi-laboratory projects show that some influential findings replicate robustly while others fail, often with substantial heterogeneity across settings (Klein et al., 2014, 2018; Hagger et al., 2016). In economics, replication has sometimes surfaced problems in papers that were both widely cited and politically salient (Reinhart and Rogoff, 2010; Herndon et al., 2014; Toner-Rodgers, 2024).

These problems persist in part because verification produces public goods while imposing private costs. Replication attempts can be time-consuming, hard to publish, and professionally risky, because replication is not perceived as original research (Vanpaemel et al., 2015; Freedman et al., 2015; Gherghina and Katsanidou, 2013). Editorial policies requiring data and code have improved availability, but compliance and usability vary widely, and the remaining effort to run and understand another team's workflow is still substantial (Trisovic et al., 2022; Hardwicke et al., 2018). Even reproducing a published table from the same data can fail due to missing code, ambiguous documentation, proprietary data, or brittle software environments (Peng, 2011; Sandve et al., 2013; Goodman et al., 2016). As a result, producing a complex analysis is often easier than checking whether it actually runs.

This paper proposes a workflow, assisted with generative large language models, for assessing the computational reproductibility of a study in quantitative social science. We present a prototype and test it on a classical paper from cultural sociology Bryson (1996). Our tool has various uses that all contribute to producing better research, and to reducing the cost of replicating older papers. Given a published article, its dataset, and a codebook, our prototype translates the methods section into executable code, runs that code, and compares outputs to the published tables and figures. When results differ, the system generates a discrepancy report and iteratively debugs the code. The value lies not only in successful reproduction but also in informative failure: when the system cannot match published outputs, it surfaces both errors in statistical code implementation and underspecified elements in the main article (such as undocumented preprocessing steps, ambiguous variable definitions), all of which make independent verification difficult.

Quantitative social science is particularly well-suited to automated verification. The field's reliance on standardized workflows, such as linear regression and its extensions, familiar covariates from publically available individual or household surveys, and conventional reporting formats like summary and regression tables, makes statistical analyses legible to machines. More importantly, this standardization potentially enables verification at scale: the same automated logic that checks one paper can be applied to hundreds or thousands. In a recent large-scale experiment, Brodeur et al. (2025) examined how much AI should be involved with humans in social science replication. Our paper pushes further by proposing a fully AI-led replication system.

The remainder of this paper proceeds as follows. We clarify the conceptual boundary between computational reproducibility and replicability, using case studies to illustrate where LLM verification can help and where it cannot. We then describe the design of our automated system, which integrates large language models with a code-execution sandbox to iteratively translate methods text into executable code. Next, we present the full case study replicating Bryson (1996). We conclude by assessing current technical limitations, outlining deployment scenarios for authors and journals, and discussing how verification infrastructure can reshape scientific incentives.

2. Background: The Replication Crisis and AI Capabilities

Discussions of the "replication crisis" often conflate failures that have different causes and require different solutions. Following Goodman et al. (2016) and the National Academies report (National Academies of Sciences, Engineering, and Medicine, 2019), we distinguish between *replicability* and *computational reproducibility*. Replicability asks whether a finding holds when researchers collect new data, perhaps in different populations or settings. Computational reproducibility asks a simpler question: given the *same* data and the same procedures, can other researchers get the same results?

Freese and Peterson (2017) offer a finer breakdown for quantitative social science, identifying four types of replication:

- **Verifiability:** Checking whether reported results can be reproduced from the same data and code. This matches computational reproducibility in Goodman et al. (2016)'s terms.
- **Robustness:** Testing whether findings hold under different analytic choices.
- **Repeatability:** Collecting new data using the original procedures to see whether the effect reappears.
- **Generalizability:** Testing whether similar findings appear with different methods or in different settings.

Each type faces different challenges. *Verifiability* failures usually stem from everyday technical problems: missing or incomplete code, unclear data-cleaning steps, undocumented software settings, changing software versions, or simple coding errors (Peng, 2011; Sandve et al., 2013; Collberg and Proebsting, 2016). Fixing these problems requires better infrastructure, incentives, and tools. *Robustness* failures happen when results depend heavily on particular analytic choices (the many small decisions researchers make that may not be fully reported) (Gelman and Loken, 2013; Simonsohn et al., 2014). *Repeatability* failures often reflect studies with too few subjects or publication bias that inflates early effect estimates (Ioannidis, 2005; John et al., 2012). *Generalizability* failures occur when findings do not hold for new populations, time periods, or measurement approaches. Solutions for these last three types focus on better study design, stronger theory, preregistration, and coordinated replication projects.

Data fabrication. At the extreme end of replicability (or repeatability) failures lies data fabrication. High-profile cases span many fields (Campos-Varela and Ruano-Raviña, 2019a,b), including social science (Data Colada, 2021). These fabrication cases are conceptually important because they clarify a boundary: even flawless code cannot rescue invalid data. Forensic methods can sometimes flag suspicious patterns, but they are inherently limited. In psychology, tools such as GRIM and SPRITE evaluate whether reported means and related summary statistics are arithmetically compatible with integer-valued data (Brown and Heathers, 2017; Heathers et al., 2018). In microbiology and related fields, image-integrity systems such as Proofig AI and Imagetwin use computer vision to detect duplicated or manipulated blots and microscopy images (Proofig Ltd., 2024; Imagetwin, 2024). These approaches can highlight internal inconsistencies or apparent manipulation, yet they cannot verify that data collection occurred as described. However, in most data fabrication cases, raw data are purposely made unavailable. For now, our proposed workflow only applies to publicly available data.

Verifiability failures. This paper focuses on verifiability (computational reproducibility failures where the data are, at least in principle, genuine) when the reported results cannot be regenerated from the shared materials. Survey evidence suggests that such failures are common and that confidence in the reproducibility of published work is low (Baker, 2016). In biomedicine, concerns about irreproducible preclinical findings have motivated systematic replication projects and methodological reforms (Begley and Ellis, 2012; Errington et al., 2021). A vivid illustration comes from Baggerly and Coombes (2009), who reconstructed the computational methods behind influential gene-expression signatures intended to guide cancer treatment. Their reanalysis uncovered misaligned samples, mislabeled arrays, and other data-handling errors that invalidated key conclusions and, in some cases, posed direct risks to patients. More broadly, empirical audits across fields show how frequently shared artifacts fail to run or fail to support published analyses (Collberg and Proebsting, 2016; Trisovic et al., 2022; Hardwicke et al., 2018).

Verifiability failures are especially consequential in empirical social science research, where quantitative results routinely inform policy debates and public narratives. The Reinhart–Rogoff paper on public debt and economic growth (Reinhart and Rogoff, 2010) became central to austerity debates by claiming that debt-to-GDP ratios above 90% are associated with sharply lower growth. A graduate-student replication by Herndon et al. (2014) revealed that the headline conclusion depended on a spreadsheet error, selective exclusion of available observations, and unconventional weighting of country-year data; correcting these issues substantially weakened the result. The influential abortion–crime hypothesis (Donohue III and Levitt, 2001) likewise faced serious challenges when Foote and Goetz (2008) identified coding errors and specification choices that attenuated the estimated effect. In development economics, the colonial-origins literature (Acemoglu et al., 2001) provoked methodological criticism when Albouy (2012) showed that core results were sensitive to how historical variables were coded and which colonies were included. Large, coordinated replication efforts in psychology also reveal heterogeneous replication rates and substantial variation in effect sizes across labs (Open Science Collaboration, 2015; Klein et al., 2014).

Why LLM for social science? Quantitative social science offers an ideal environment for automated verification because applied work across economics, political science, sociology, and related disciplines follows standardized patterns. First, researchers rely on widely shared datasets, such as public opinion surveys (GSS, WVS, ANES), census and demographic data (Population Census, DHS), and household surveys (CPS, ATUS), with organized data structures and comprehensive documentation. These data are also accessible through public data dissemination systems, for example, IPUMS and the GSS Data Explorer. Second, analyses employ conventional, explicitly specified models: ordinary least squares, logit and probit models, and other generalized linear models. Third, results appear in canonical formats, such as regression tables with coefficients and standard errors, summary statistics panels, and standardized figures, that machines can systematically parse and verify.

These features substantially reduce the semantic gap between natural-language methods descriptions and executable workflows. The remaining ambiguities (missing-value treatment, sample restrictions, variable transformations) are precisely the tacit choices that cause reproducibility failures. An LLM verification system can make these ambiguities explicit by attempting execution and reporting discrepancies. This approach is feasible because modern large language models are increasingly competent at reading and writing statistical code and at repairing code when given concrete execution errors (Chen et al., 2021; Jimenez et al., 2024; Zhou et al., 2024). They can also act as flexible interpreters between modalities, mapping method descriptions to variable definitions and synthesizing step-by-step execution plans (Yao et al., 2023; Wang et al., 2025). Compared to data fabrication, computational reproducibility failures are both more common and more tractable: they arise from incomplete specification and technical brittleness rather than deliberate deception. This motivates a verifier that attempts to rerun analyses, records where and how execution fails, and produces structured discrepancy reports anchored in observable outputs.

Recent experimental evidence supports this reasoning but also reveals important limitations. Brodeur et al. (2025) compared human-only, AI-assisted, and AI-led teams in a large-scale replication exercise for quantitative social science research. They found that AI-led teams performed substantially worse at identifying coding and data errors, particularly those involving conceptually flawed code leading to incorrect data handling. Our approach addresses this limitation by grounding AI reasoning in concrete execution outcomes. Following the iterative structure of the

"AI Scientist" framework (Lu et al., 2024), our system writes code, runs it, and feeds any errors or output mismatches back to the model for revision. This loop of generation, execution, and refinement continues until outputs match published results, allowing the system to catch errors that a single-pass review would miss.

Although the conceptual distinction between replicability and computational reproducibility is important, the terms *replication*, *replicability*, and *reproducibility* are often used inconsistently across fields (Rougier et al., 2017; Plesser, 2018). In this article, we use "replication" in its broader, colloquial sense to encompass computational reproduction, relying on context to preserve the narrower distinctions when needed.

3. System Design: An Automated Replication System

We developed a prototype method to test whether fully automated replication is possible. Unlike simple bots that run a script once, our system uses an agentic workflow that mimics a researcher who plans, codes, checks their work, and fixes errors. The entire system is implemented as a single Python script (the *Main Script*) that orchestrates interactions between a local code execution environment and an LLM API.

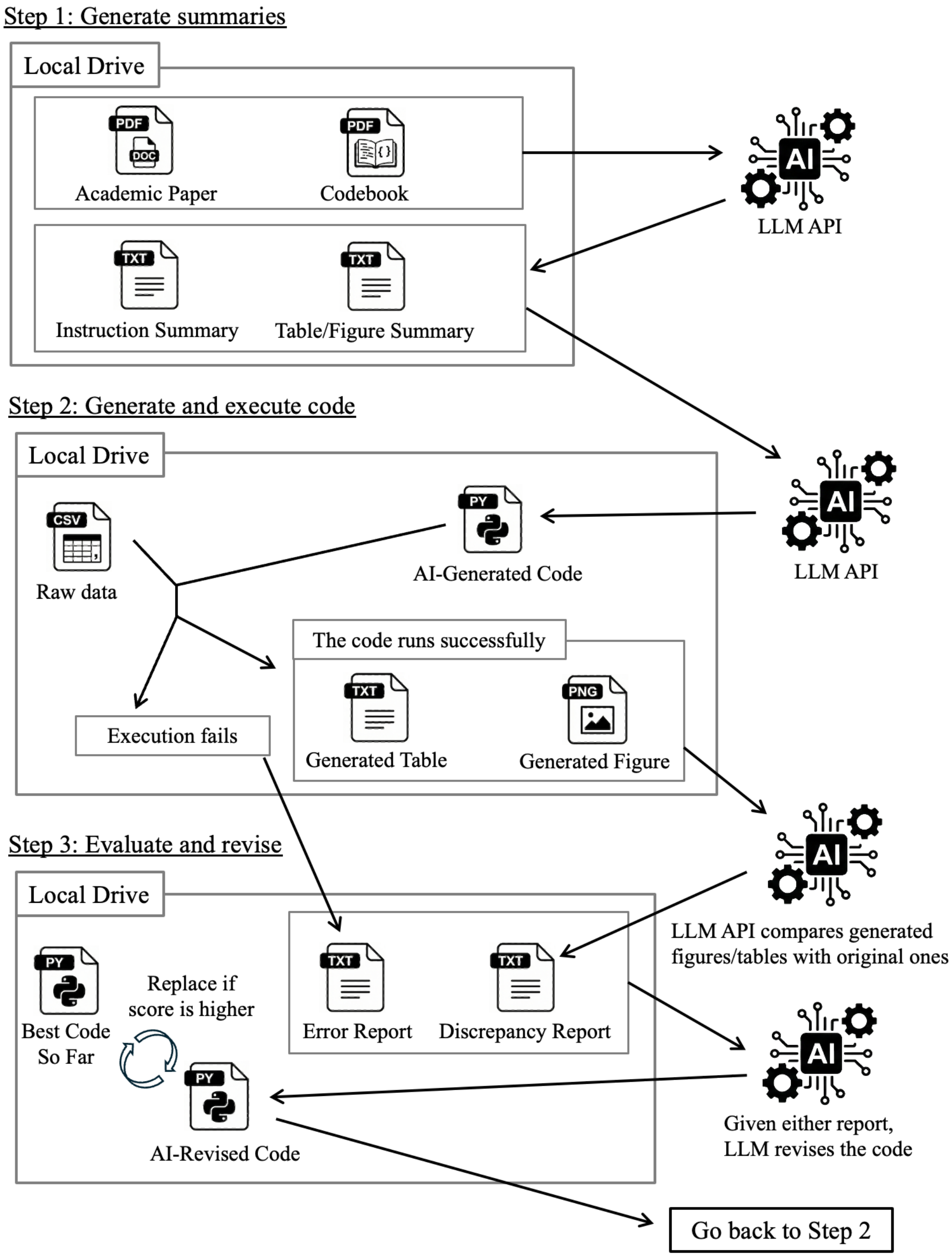
Figure 1 summarizes the proposed automated replication procedure as a three-step loop: (1) generate structured specifications from the paper and codebook, (2) generate and execute analysis code against the local dataset, and (3) evaluate outputs using a numeric alignment score and revise iteratively. The system tracks the best-scoring attempt throughout execution and terminates when alignment exceeds a threshold or the maximum number of iterations is reached.

Step 1: Generate summaries and extract target results. The Main Script's directory houses three primary inputs: (i) the target academic article (typically a PDF), (ii) a dataset codebook or variable documentation, and (iii) the locally stored replication dataset (in formats such as CSV, Excel, R, Stata, or SPSS). In Step 1, the Main Script converts these unstructured research materials into structured specifications by making several LLM API calls.

First, the Main Script uploads the paper PDF to the LLM API and requests a *Table/Figure summary*: a detailed specification of the target table or figure. For regression tables, this includes dependent and independent variables, model specifications, variable transformations (e.g., standardization, index construction), sample restrictions, and missing-value handling rules; for summary tables, it specifies the statistics to compute and any grouping or stratification criteria. For figures, this includes the underlying statistical analysis, axis specifications (labels, ranges, ordering of categories), and visual elements such as reference lines and annotations. Second, the Main Script sends the codebook along with the analysis summary to generate an *Instruction summary*: a structured mapping from the paper's variable names to specific dataset column names, value recoding rules derived from questionnaire response options, construction rules for derived variables (e.g., summing across items to create an index), and missing-value codes to exclude for each variable.

These specification documents are critical because, even when the statistical model is standard,

Figure 1: Overview of the Automated Replication Procedure



details about missing-value handling, weighting, item coding, and index thresholds are often unclear until implementation attempts begin.

Step 2: Generate and execute code. The Main Script sends the analysis summary and instruction summary from Step 1 to the LLM API, along with a preview of the dataset (column names and the first five rows in this prototype), to generate executable analysis code. This LLM-generated code is written as a callable function (in Python, for this prototype) that returns structured results as text or data frames.

The Main Script then executes this code in the local environment, accessing the raw data from the local drive. When the code runs successfully, it produces generated tables (as structured text) or figures (as image files). If execution fails, the Main Script captures the exception and stack trace, producing an *error report* that includes the failing module, the exception message, and relevant context (e.g., variable names present, dimensions of the current dataframe).

Step 3: Evaluate alignment and revise. When execution in Step 2 succeeds, the Main Script evaluates the generated outputs through two mechanisms. First, it computes a numeric *alignment score* on a scale from 0 to 100 by sending the generated results and the true results to the LLM API. The alignment score quantifies how well the generated outputs match the original paper, considering variable names, sample sizes, coefficient estimates, standard errors, and significance levels. A score of 100 indicates perfect reproduction, while 0 indicates complete mismatch.

Second, if the alignment score falls below the threshold (95 in this prototype), the Main Script requests a qualitative *discrepancy report* from the LLM API. This report details specific differences between the regenerated outputs and the original paper: for regression tables, it records mismatches in sample size N, coefficient values, standard errors, R^2 statistics, and whether specifications include the intended covariates; for figures, it compares axis labels and ranges, data series patterns and values, reference lines and annotations, and overall layout properties such as line styles and tick formatting.

Best result tracking. Throughout the iteration loop, the system maintains a *best result tracker* that stores the highest-scoring attempt's code, results, and metadata. Whenever a new attempt achieves a higher alignment score than the previous best, the tracker is updated. This ensures that even if the system cannot achieve perfect alignment, the best approximation is preserved for human review.

Iteration and termination. The Main Script sends either the discrepancy report (if code executed successfully) or the error report (if execution failed) back to the LLM API, along with the best result so far, its score, and the best code. The LLM then produces revised code that attempts to improve upon or maintain the quality of the best result while addressing the identified issues. To prevent context overflow from accumulating feedback across many iterations, the system truncates the feedback history to a maximum length (default: 10,000 characters), retaining only the most recent information.

The loop returns to Step 2, where the Main Script executes the revised code and re-evaluates alignment. The iteration continues until one of two termination conditions is met:

1. **Success:** The alignment score meets or exceeds the threshold (e.g., ≥ 95), indicating successful reproduction.

2. **Maximum iterations:** The configured maximum number of attempts is reached (e.g., 100 iterations).

In either case, the system returns the best result achieved across all attempts, along with accumulated discrepancy reports that document remaining differences for human review.

4. Case Study: Replicating Bryson (1996)

We evaluate feasibility using Bryson (1996), a classic sociology paper on musical dislikes and symbolic exclusion, using the 1993 General Social Survey. This paper is representative of a common empirical pattern in social science: it relies on a widely used shared dataset, applies standard regression models, and constructs non-trivial variables (including multi-item indices) that require careful interpretation of documentation. This highly cited paper establishes that despite a general opening in cultural tastes, strong dislikes for marginal cultural genres remain. Later research has shown an evolution in the pattern of dislikes (Lizardo and Skiles, 2016) but the empirical results have mostly gone unchallenged and the theoretical contribution is regarded as central by contemporary sociologists of culture. Bryson's main results include regression tables with multiple specifications and a figure that effectively repeats a similar model across many music genres and then aggregates the results into a comparative visual summary. From the perspective of Figure 1, the task stresses all three stages: extracting a correct target specification, compiling it into code that runs, and reconciling discrepancies that arise from underspecified preprocessing and variable construction.

In Step 1, we place three files on the local drive: (i) the Bryson paper PDF, (ii) the GSS 1993 data file (CSV format), and (iii) the relevant GSS codebook documentation (TXT format). We use the OpenAI API for all LLM interactions (GPT-5.2 for summarization and extraction, GPT-5.2-codex for code generation). The full GSS dataset is too large for practical use, so we create a compact replication dataset containing only variables potentially relevant to the replication, along with a corresponding codebook file that includes variable descriptions, question wording, and value labels. We use R's gssr package (Healy, 2024) for this preprocessing step.

The Bryson paper contains three tables and one figure. We write separate and independent Python programs for each target. Consider Table 1 as an example. It reports standardized OLS coefficients from three nested specifications predicting musical exclusiveness (the count of music genres a respondent dislikes) from socioeconomic status (SES), demographics, and political intolerance. The dependent variable is constructed from 18 GSS items asking respondents to rate genres on a five-point scale; responses of "dislike" or "dislike very much" count toward the index, while "don't know" responses are treated as missing (dropping the respondent from the analysis). Model 1 includes only SES variables (education, household income per capita, occupational prestige); Model 2 adds demographic controls (gender, age, race/ethnicity indicators, religion indicators, region); Model 3 adds a political intolerance scale constructed from 15 items about civil-liberties asked of two-thirds of the sample.

The LLM-generated Table/Figure summary extracts a detailed specification from the published output: it records the dependent variable definition, lists each independent variable for all three model specifications, documents the variable transformations required (standardization to obtain beta coefficients), and notes sample restrictions and missing-value handling rules. The Instruction summary, by contrast, maps these analysis concepts to specific GSS variable names and recoding rules derived from the codebook. It specifies, for example, that musical exclusiveness equals the sum of 18 binary indicators (one per genre, coded 1 if the response is 4 or 5 on the Likert scale), that the sample should be restricted to 1993, and that listwise deletion should be applied within each model.

Table 1: Original vs. LLM-Generated Table 1 from Bryson (1996)

	Model 1: SES		**Model 2: Demographic**		**Model 3: Intolerance**	
Variable	**Original**	**LLM**	**Original**	**LLM**	**Original**	**LLM**
Education	-0.322***	-0.292***	-0.246***	-0.265***	-0.151***	-0.155*
Income (Per Capita)	-0.016	-0.039	-0.038	-0.051	-0.021	-0.052
Occ. Prestige	0.009	0.020	-0.002	-0.011	-0.038	-0.015
Female	–	–	-0.106**	-0.085*	-0.126***	-0.127*
Age	–	–	0.129***	0.103*	0.093*	0.091
Race (Ref: White)						
Black	–	–	0.037	0.100	0.041	0.060
Hispanic	–	–	-0.037	0.074	-0.043	-0.030
Other	–	–	0.010	-0.027	0.014	0.053
Religion/Region						
Cons. Protestant	–	–	0.038	0.087	0.014	0.036
No Religion	–	–	-0.025	-0.015	-0.011	0.023
Southern	–	–	0.076*	0.061	0.085*	0.068
Political Intolerance	–	–	–	–	0.231***	0.184**
Model Statistics						
Sample Size (N)	787	747	647	507	353	286
R^2	0.10	0.088	0.14	0.139	0.16	0.149
Constant	11.235	10.638	8.941	8.675	7.744	7.999

Note: Table reports standardized coefficients. Significance: *** $p < .001$, ** $p < .01$, * $p < .05$.

In our execution with the GPT-5.2 API, the system iterated 100 times through Steps 2 and 3. Although it did not converge to fully reproduce the exact numbers in Table 1, the program nevertheless replicated the table at an acceptable level. See also Table 2 for a summary of execution statistics. The total API cost for this replication exercise was approximately USD 60.

Of 100 iterations, 95 produced valid output while the remaining 5 terminated with runtime errors during code execution. Interestingly, the program generated reasonable results for the first and second specifications on the second trial. After that, the program attempted minor improvements to these specifications but struggled with the third specification, particularly with constructing the Political Intolerance variable correctly. The discrepancy reports primarily identified inconsistencies in the number of observations, statistical significance levels, and model fit.

For Table 2 and Table 3, detailed results and discussion are provided in the Appendix.

We also test the image handling capabilities of this LLM replication by replicating Figure 1 in

Table 2: Execution Summary for Replication Attempts

				Score		Duration (s)		
Target	**Trials**	**Errors**	**Scored**	**Mean**	**Max**	**Mean**	**Max**	**Best Attempt**
Table 1	100	5	95	28.3	54	130.6	255.4	#27
Table 2	100	37	63	18.5	45	123.7	234.8	#6
Table 3	100	46	54	67.4	91	46.9	140.2	#29
Figure 1	22	0	22	64.5	98	89.5	280.1	#22

Note: Score ranges from 0 (no match) to 100 (perfect reproduction). Duration measured in seconds per attempt.

Bryson (1996). This figure plots logistic regression coefficients for musical tolerance and overlays average education levels for each genre's audience. The Procedure Manual is similar to those for tables, but the Table/Figure summary adds the figure structure in addition to the statistical models. The figure structure describes visual properties such as the labels and scales of the axes, the properties of lines, and the annotation texts with arrows. The system successfully replicated Figure 1 after 22 iterations, achieving a final score of 98/100. Early attempts encountered several issues: the x-axis category order was reversed, the two data series were assigned to the wrong y-axes with incorrect line styles (solid vs. dash-dot), and annotation arrows pointed to the wrong lines. These problems were iteratively corrected through discrepancy reports that identified the mapping errors between statistical series and visual elements. The final replication accurately reproduces both the statistical content and the visual presentation, demonstrating that our method can handle figure replication tasks that combine statistical computation with graphical formatting.

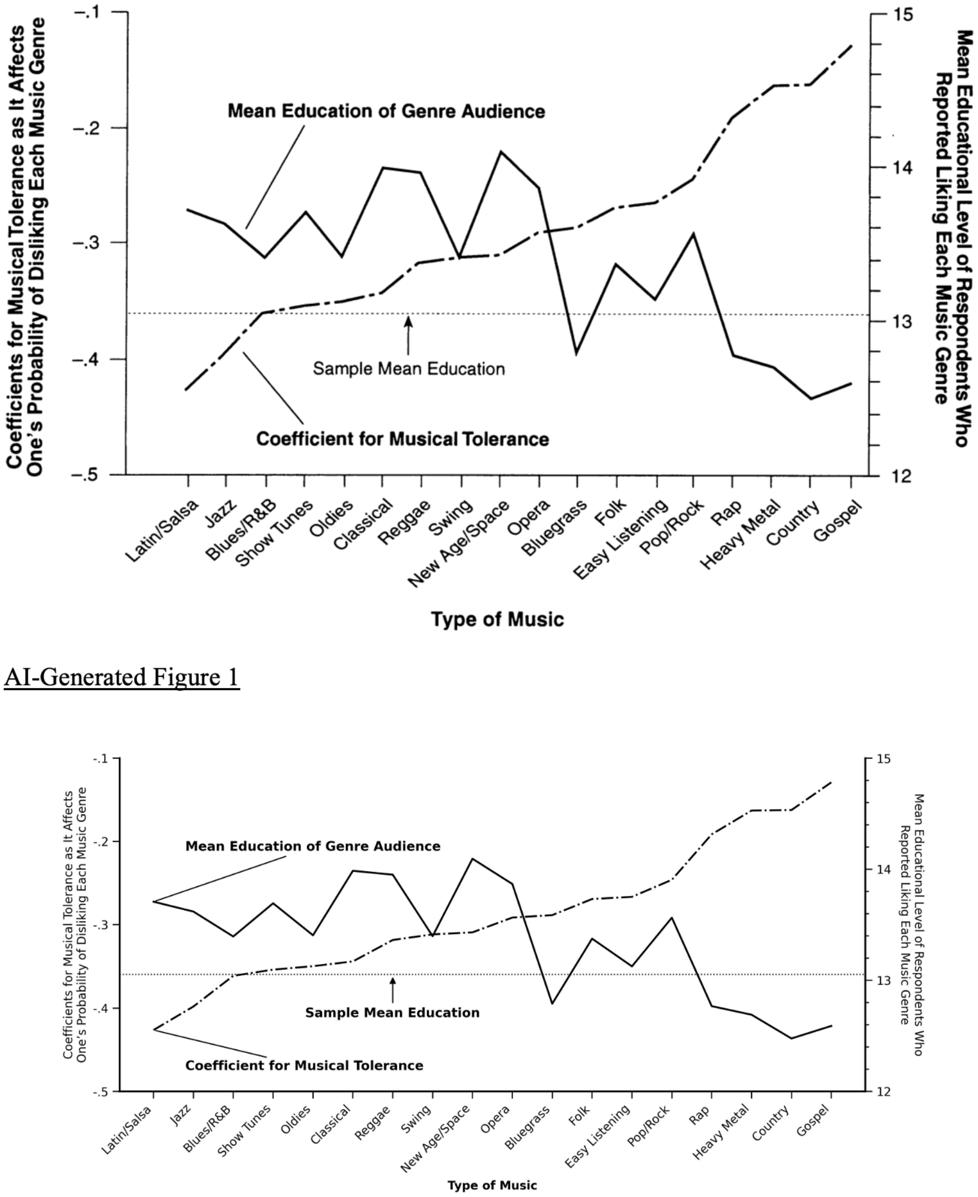
Overall, the exercise of Bryson (1996) stresses the system's ability to interpret methods text, map concepts to variable definitions, handle non-trivial variable construction (multi-item indices), and resolve ambiguities through iterative debugging. The system successfully navigates these challenges to produce results that closely match the published outputs.

5. Current Capabilities and Limitations

The exercise illustrates that LLM-assisted verification is feasible for replicating major statistical results common in social science papers: summary tables of data, regression tables, and figures. We emphasize that while this is a preliminary step, it is applicable to a wide range of empirical social science papers. The data source, GSS, is a standard public microdata source sharing a basic structure common to many other social science datasets, such as WVS, CPS, and ANES. The statistical models (ordinary least squares and logistic regression) are canonical in applied social science. Because our AI-driven method builds on techniques that constitute the empirical backbone of quantitative social science, the approach demonstrated here is inherently scalable.

Technical limitations. However, several limitations remain. First, the LLM-driven system struggles with *underspecified methodological details.* For example, in Bryson (1996), the coding of the Hispanic variable is not explicitly described. It is likely created from an ethnic variable

Figure 2: Original Figure 1 vs. LLM-Generated Replication



Original Figure 1 from Bryson (1996)

AI-Generated Figure 1

about the country of family origin; however, the paper does not specify which countries are treated as Hispanic. Second, our prototype currently operates on *single rectangular datasets*, whereas many research designs require merging multiple data sources or linking distinct survey waves. For example, many studies using WVS create country-level variables as the mean of individual-level responses within each country and merge them with external macroeconomic or demographic data. Such multi-dataset workflows introduce additional complexities in data integration, matching logic, and provenance tracking that are not yet addressed in the current system. Third, the present workflow is *optimized for cross-sectional data*; adapting it to panel data would introduce complexities related to time-series structure and repeated measures that are not yet fully addressed. Analyzing Panel Study of Income Dynamics (PSID) or National Longitudinal Survey of Youth (NLSY) data would be a future challenge. Fourth, our prototype focuses on *canonical statistical models* (OLS, logit, probit, and their extensions); extending it to handle more complex models (e.g., hierarchical models, structural equation models, machine learning algorithms) would require additional development. Finally, while the system can identify discrepancies in outputs, it does not yet provide detailed diagnostics or explanations for why mismatches occur, limiting its utility for debugging complex workflows.

Epistemic boundary of LLM-assisted reproducibility. Aside from these technical limitations, it is more important to clarify the epistemic boundary of LLM-assisted reproducibility. If this system succeeds in reproducing published results, there are two possibilities: the LLM system replicates by following the main text, or it engages in a trial-and-error process that deviates from the workflow in the main text but still arrives at the results. The former is ideal, but the latter is also valuable because it reveals underspecification in the original article. We also emphasize that even in the former case, successful reproduction does not guarantee the validity or ground truth of the scientific claim. It only verifies the procedural consistency given the dataset and chosen method within the paper's scope. Large-scale replication projects have shown that many influential results fail to replicate when new data are collected, even when the original analysis was computationally sound (Open Science Collaboration, 2015; Klein et al., 2018; Camerer et al., 2016). This distinction is particularly relevant for archival data research. Delios et al. (2022) attempted to replicate findings from 110 strategic management papers using archival sources; they found that while many results could be computationally reproduced, their generalizability to extended time periods or alternative specifications was much lower. Our position is that computational reproducibility is not the endpoint of scientific evaluation, but it is a necessary prerequisite for credible debate. An AI verifier should be understood as assistive infrastructure that checks mechanical consistency rather than as an arbiter of scientific truth (Freese and Peterson, 2017).

The meaning of a failed reproduction is more nuanced, and does not automatically imply that the original finding is wrong. Reproduction failures can arise for many reasons, and the most common one is the incapability of LLM. The technical limitations described above are still too substantial to overcome. In some cases, the failure simply reflects underspecification: the methods section omitted details about variable coding, sample restrictions, or estimation options that were obvious to the original authors but not recoverable from the text.

However, some reproduction failures do signal deeper issues. High-profile cases such as the Reinhart–Rogoff spreadsheet error (Herndon et al., 2014) show that seemingly minor computational errors can alter substantive conclusions. When an LLM verifier fails to reproduce a result and the discrepancy persists across multiple debugging iterations, this should trigger closer human investigation. The value of automated verification is not that it pronounces papers "correct" or "incorrect," but that it systematically surfaces inconsistencies that warrant further investigation. A discrepancy report becomes a starting point for dialogue: authors can clarify their workflow, reviewers can assess whether differences matter substantively, and readers can make informed judgments about how much weight to place on the findings.

In short, reproducibility is a reliability check, not a validity proof. The goal of LLM-assisted verification is to make the former cheaper and more routine, thereby freeing human attention for the latter.

6. Use Cases and Deployment Scenarios

If LLM-assisted reproducibility is to matter for scientific practice, it must fit into real workflows. We outline three deployment scenarios, each with distinct governance needs.

Local pre-submission checks by authors. The lowest-friction use is as a local "reproduce-and-compare" tool for authors. Authors run the system before submission to detect missing files, inconsistent seeds, or silent sample-size changes. If the LLM cannot reproduce results from the manuscript and data, it signals that documentation is incomplete, allowing authors to fix ambiguities before peer review. This turns reproducibility into a pre-submission quality check rather than a post-publication crisis.

Institutional verification systems. The LLM-based verifier can also be integrated into institutional workflows. The most straightforward case is journal-run verification during peer review. AI systems have supported the peer review process, from screening for reporting quality to generating reviewer comments (Saito et al., 2024; Hosseini et al., 2025; Zhang et al., 2024). The LLM-based verifier adds one more layer to verify that the statistical procedure matches the main text. Academic journals increasingly require data and code, and some are experimenting with formal code review as part of peer review (Nature Human Behaviour Editorial, 2021). For example, the American Economic Association has built repositories and editorial processes that require replication packages (American Economic Association, 2024). However, this effort only confirms that the submitted program is runnable and produces the same results. The review process can focus on substance after the methodological correctness is confirmed by the LLM-based system. It helps reviewers focus on interpretation and scientific judgment.

This system can also be extended to a service that routinely attempts to reproduce published papers, creating a continuous audit of the scientific record. This aligns with recent proposals such as the "Replication Engine" by the Institute for Progress, which envisions AI agents automatically verifying results at the moment of publication (Institute for Progress, 2025). Such automated infrastructure complements the work of the Institute for Replication (I4R), which organizes large-scale human replication efforts and is increasingly moving toward routine checks (I4R, 2024). As Brodeur et al. (2024b) argue, institutionalizing these checks is critical to solving the supply problem of replication; automated tools can scale this institutional capacity by handling the mechanical verification tasks that currently bottleneck human replicators.

Forensic verification. A distinctive application of LLM-assisted verification is as a *forensic tool* for legacy research. The vast majority of social science papers published before the mid-2000s lack replication packages: the American Economic Association adopted its first data availability policy only in 2005, and most sociology journals have no such requirements even today (Freese and Peterson, 2017). A recent study of papers using the German Socio-Economic Panel found that only 6% have replication code available, with availability sharply lower for older publications (Fink et al., 2025). For influential papers from this era, the original code may be lost, stored on obsolete media, or written in software versions that no longer run. Yet many of these studies used publicly available datasets that remain accessible. An LLM verifier can attempt to reconstruct the analysis workflow from the methods section alone, generating code that approximates what the original authors likely ran.

In addition to recovering the past, this infrastructure serves as a forensic tool for disputed findings. When results are questioned, an automated reproduction attempt can quickly determine whether concerns are about simple rerun failures (missing code, wrong file versions) or about deeper inconsistencies. When errors are suspected, as in the Reinhart-Rogoff case, automated tools can provide rapid forensics.

7. Conclusion

Large language models are transforming scientific production, creating a risk that the supply of plausible-sounding claims will outpace the community's capacity to verify them. This paper argues that the same technologies driving this acceleration can be harnessed to strengthen scientific integrity. We introduce an automated verification system that functions as a replication compiler, translating natural-language methods into executable code. By applying this system to a classic sociology study (Bryson, 1996), we demonstrated that current LLMs can successfully reproduce key statistical results from widely used public datasets, while also surfacing the ambiguities and tacit knowledge that often hinder human replication efforts.

Several technical directions appear promising. One is tighter integration with research repositories and data providers, including standardized metadata and executable environments. Another is extending the method to handle common but more complex structures (survey weights, panels, and multi-source merges) by combining LLMs with domain-specific templates. A third is community-driven libraries of procedure-manual patterns for canonical datasets, analogous to shared codebooks but focused on analysis recipes. Finally, as models improve, verifiers may become capable not only of reproducing tables, but also of checking robustness specifications and sensitivity analyses in a standardized way (Brodeur et al., 2024a,b).

Ultimately, because verification is a public good (Freedman et al., 2015), it should not be an act of heroism by individual researchers but a routine feature of the scientific infrastructure. If we can lower the cost of checking basic consistency, we free human attention for the deeper tasks of interpretation and theory building. By treating reproducibility as a machine-actionable property, we can ensure that the next era of quantitative social science, though faster and more automated, remains firmly grounded in verifiable evidence.

References

Acemoglu, D., Johnson, S., and Robinson, J. A. (2001). The colonial origins of comparative development: An empirical investigation. *American Economic Review*, 91(5):1369–1401.

Aher, G. V., Arriaga, R. I., and Kalai, A. T. (2023). Using large language models to simulate multiple humans and replicate human subject studies. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 337–371. PMLR.

Albouy, D. Y. (2012). The colonial origins of comparative development: An empirical investigation: Comment. *American Economic Review*, 102(6):3059–3076.

American Economic Association (2024). Aea data and code repository. https://www.openicpsr.org/openicpsr/aea.

Argyle, L. P., Busby, E. C., Fulda, N., Gubler, J. R., Rytting, C., and Wingate, D. (2023). Out of one, many: Using language models to simulate human samples. *Political Analysis*, 31(3):337–351.

Baggerly, K. A. and Coombes, K. R. (2009). Deriving chemosensitivity from cell lines: Forensic bioinformatics and reproducible research in high-throughput biology. *The Annals of Applied Statistics*, 3(4):1309–1334.

Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454.

Begley, C. G. and Ellis, L. M. (2012). Raise standards for preclinical cancer research. *Nature*, 483(7391):531–533.

Boelaert, J., Coavoux, S., Ollion, E., Petev, I. D., and Präg, P. (2025). Machine Bias. How Do Generative Language Models Answer Opinion Polls? *Sociological Methods & Research*, 54(3):1156–1196.

Bran, A. M. et al. (2024). Chemcrow: Augmenting large-language models with chemistry tools. *Nature Machine Intelligence*.

Brand, J., Israeli, A., and Ngwe, D. (2023). Using gpt for market research. *arXiv preprint arXiv:2305.03763*.

Brodeur, A., Carrell, S., Figlio, D., and Lusher, L. (2024a). Mass reproducibility and replicability: A new hope. *American Economic Review*, 114(6).

Brodeur, A., Esterling, K., Ankel-Peters, J., Bueno, N. S., Desposato, S., Dreber, A., Genovese, F., Green, D. P., Hepplewhite, M., Hoces de la Guardia, F., Johannesson, M., Kotsadam, A., Miguel, E., Velez, Y. R., and Young, L. (2024b). Promoting reproducibility and replicability in political science. *Research and Politics*, 11(1):1–8.

Brodeur, A., Valenta, D., Marcoci, A., Aparicio, J. P., Mikola, D., Barbarioli, B., Alexander, R., Deer, L., Stafford, T., et al. (2025). Comparing human-only, ai-assisted, and ai-led teams on assessing research reproducibility in quantitative social science. IZA Discussion Paper 17645, IZA Institute of Labor Economics.

Brown, N. J. and Heathers, J. A. (2017). The grim test: A simple technique detects numerous anomalies in the reporting of results in psychology. *Social Psychological and Personality Science*, 8(4):363–369.

Bryson, B. (1996). “anything but heavy metal”: Symbolic exclusion and musical dislikes. *American Sociological Review*, 61(5):884–899.

Camerer, C. F., Dreber, A., Forsell, E., Ho, T.-H., Huber, J., Johannesson, M., Kirchler, M., Almenberg, J., Altmejd, A., Chan, T., et al. (2016). Evaluating replicability of laboratory experiments in economics. *Science*, 351(6280):1433–1436.

Camerer, C. F., Dreber, A., Holzmeister, F., Ho, T.-H., Huber, J., Johannesson, M., Kirchler, M., Nave, G., Nosek, B. A., Pfeiffer, T., et al. (2018). Evaluating the replicability of social science experiments in nature and science between 2010 and 2015. *Nature Human Behaviour*, 2(9):637–644.

Campos-Varela, I. and Ruano-Raviña, A. (2019a). Scientific fraud. part i: Definition, general concepts, historical cases. *Archivos de Bronconeumología*, 55(10):533–538.

Campos-Varela, I. and Ruano-Raviña, A. (2019b). Scientific fraud. part ii: From past to present, facts and analyses. *Archivos de Bronconeumología*, 55(11):597–602.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. d. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. (2021). Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Collberg, C. S. and Proebsting, T. A. (2016). Repeatability in computer systems research. *Communications of the ACM*, 59(3):62–69.

Conroy, G. (2023). Scientists use ai to write research papers. is that a problem? *Nature*, 620:27.

Data Colada (2021). [98] evidence of fraud in an influential field experiment about dishonesty. https://datacolada.org/98.

Delios, A., Clemente, E. G., Wu, T., Tan, H., Wang, Y., et al. (2022). Examining the generalizability of research findings from archival data. *Proceedings of the National Academy of Sciences*, 119(30):e2120377119.

Donohue III, J. J. and Levitt, S. D. (2001). The impact of legalized abortion on crime. *The Quarterly Journal of Economics*, 116(2):379–420.

Du, Y. et al. (2025). Llm4sr: A survey on large language models for scientific research. arXiv preprint. arXiv:2501.04306.

Errington, T. M., Mathur, M., Soderberg, C. K., Denis, A., Perfito, N., Iorns, E., and Nosek, B. A. (2021). Investigating the replicability of preclinical cancer biology. *eLife*, 10:e71601.

Fink, M. et al. (2025). Replication code availability over time and across fields: Evidence from the German Socio-Economic Panel. *Economic Inquiry*. Early view, November 2024.

Foote, C. L. and Goetz, C. F. (2008). The impact of legalized abortion on crime: Comment. *The Quarterly Journal of Economics*, 123(1):407–423.

Freedman, L. P., Cockburn, I. M., and Simcoe, T. S. (2015). The economics of reproducibility in preclinical research. *PLOS Biology*, 13(6):e1002165.

Freese, J. and Peterson, D. (2017). Replication in social science. *Annual Review of Sociology*, 43:147–165.

Gelman, A. and Loken, E. (2013). The garden of forking paths: Why multiple comparisons can be a problem, even when there is no fishing expedition" or p-hacking" and the research hypothesis was posited ahead of time. *Department of Statistics, Columbia University*.

Gherghina, S. and Katsanidou, A. (2013). How are we doing? data access and replication in political science. *PS: Political Science & Politics*.

Gilardi, F., Alizadeh, M., and Kubli, M. (2023). Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

Goodman, S. N., Fanelli, D., and Ioannidis, J. P. A. (2016). What does research reproducibility mean? *Science Translational Medicine*, 8(341):341ps12.

Hagger, M. S., Chatzisarantis, N. L., Alberts, H., Anggono, C. O., Batailler, C., Birt, A. R., Brand, R., Brandt, M. J., Brewer, G., Bruyneel, S., et al. (2016). A multilab preregistered replication of the ego-depletion effect. *Perspectives on Psychological Science*, 11(4):546–573.

Hardwicke, T. E., Mathur, M. B., MacDonald, K., Nilsonne, G., Banks, G. C., Kidwell, M. C., Hofelich Mohr, A., Clayton, E., Yoon, E. J., Tessler, M. H., Lenne, R., and Frank, M. C. (2018). Data availability, reusability, and analytic reproducibility: Evaluating the impact of a mandatory open data policy at the journal cognition. *Royal Society Open Science*, 5:180448.

Healy, K. (2024). *gssr: General Social Survey Data for Use in R.* R package version 0.2.0.

Heathers, J. A., Anaya, J., van der Zee, T., and Brown, N. J. (2018). Recovering data from summary statistics: Sample parameter reconstruction via iterative techniques (sprite). *PeerJ Preprints*, 6:e26968v1.

Herndon, T., Ash, M., and Pollin, R. (2014). Does high public debt consistently stifle economic growth? a critique of reinhart and rogoff. *Cambridge Journal of Economics*, 38(2):257–279.

Hosseini, M. et al. (2025). Large language models for automated scholarly paper review: A survey. *Information Fusion*. Available online.

I4R (2024). Institute for replication. https://i4replication.org/.

Imagetwin (2024). Imagetwin: Detecting image duplications in scientific publications. https://imagetwin.ai. Accessed: 2025-01-06.

Institute for Progress (2025). The replication engine. https://ifp.org/the-replication-engine/.

Ioannidis, J. P. (2005). Why most published research findings are false. *PLoS Medicine*, 2(8):e124.

Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., et al. (2024). Swe-bench: Can language models resolve real-world github issues? In *International Conference on Learning Representations*.

John, L. K., Loewenstein, G., and Prelec, D. (2012). Measuring the prevalence of questionable research practices with incentives for truth telling. *Psychological Science*, 23(5):524–532.

Klein, R. A., Ratliff, K. A., Vianello, M., Adams Jr, R. B., Bahník, Š., Bernstein, M. J., Bocian, K., Brandt, M. J., Brooks, B., Brumbaugh, C. C., et al. (2014). Investigating variation in replicability: A “many labs” replication project. *Social Psychology*, 45(3):142–152.

Klein, R. A., Vianello, M., Hasselman, F., Adams, B. G., Adams Jr, R. B., Alper, S., Aveyard, M., Axt, J. R., Babalola, M. T., Bahník, Š., et al. (2018). Many labs 2: Investigating variation in replicability across samples and settings. *Advances in Methods and Practices in Psychological Science*, 1(4):443–490.

Lizardo, O. and Skiles, S. (2016). The End of Symbolic Exclusion? The Rise of "Categorical Tolerance" in the Musical Tastes of Americans: 1993 – 2012. *Sociological Science*, 3:85–108.

Lu, C., Lu, C., Lange, R. T., Foerster, J., Clune, J., and Ha, D. (2024). The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*.

National Academies of Sciences, Engineering, and Medicine (2019). *Reproducibility and Replicability in Science*. National Academies Press, Washington, DC.

Nature Human Behaviour Editorial (2021). Supporting computational reproducibility through code review. *Nature Human Behaviour*, 5:965–966.

Naudet, F., Siebert, M., Robinson, P., Baron, G., et al. (2018). Data sharing and reanalysis of randomized controlled trials in leading biomedical journals with a full data sharing policy: survey of studies published in the bmj and plos medicine. *BMJ*, 360:k400.

Open Science Collaboration (2015). Estimating the reproducibility of psychological science. *Science*, 349(6251):aac4716.

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*.

Peng, R. D. (2011). Reproducible research in computational science. *Science*, 334(6060):1226–1227.

Plesser, H. E. (2018). Reproducibility vs. replicability: A brief history of a confused terminology. *Frontiers in Neuroinformatics*, 11:76.

Proofig Ltd. (2024). Proofig: AI-powered image integrity solution for scientific publications. https://www.proofig.com. Accessed: 2025-01-06.

Reinhart, C. M. and Rogoff, K. S. (2010). Growth in a time of debt. *American Economic Review*, 100(2):573–578.

Rougier, N. P., Hinsen, K., Alexandre, F., Arildsen, T., Barba, L. A., Benureau, F. C. Y., Brown, C. T., De Buyl, P., Caglayan, O., Davison, A. P., et al. (2017). Sustainable computational science: The rescience initiative. *PeerJ Computer Science*, 3:e142.

Saito, K. et al. (2024). Ai-driven review systems: Evaluating llms in scalable and bias-aware peer review. arXiv preprint. arXiv:2408.10365.

Sandve, G. K., Nekrutenko, A., Taylor, J., and Hovig, E. (2013). Ten simple rules for reproducible computational research. *PLOS Computational Biology*, 9(10):e1003285.

Simonsohn, U., Nelson, L. D., and Simmons, J. P. (2014). P-curve: A key to the file-drawer. *Journal of Experimental Psychology: General*, 143(2):534–547.

Toner-Rodgers, A. (2024). The mit ai productivity research controversy. Working paper on AI and worker productivity, subject to data integrity scrutiny.

Trisovic, A., Lau, M. K., Pasquetto, I. V., Crosas, M., et al. (2022). A large-scale study on research code quality and execution. *Scientific Data*.

Vanpaemel, W., Vermorgen, M., Deriemaecker, L., and Storms, G. (2015). Are we wasting a good crisis? the availability of psychological research data after the storm. *Collabra: Psychology*, 1(1).

Wang, X. et al. (2024). Agentic scientific discovery. *arXiv preprint*.

Wang, Y. et al. (2025). A review of prominent paradigms for llm-based agents: Tool use, planning, and feedback learning. In *Proceedings of COLING 2025*.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. (2023). React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*.

Zhang, Y. et al. (2024). Is llm a reliable reviewer? a comprehensive evaluation of llm on automatic paper reviewing. In *Proceedings of LREC-COLING*.

Zhou, Y. et al. (2024). Swe-bench+: Enhanced coding benchmark for llms. arXiv preprint. arXiv:2410.06992.

A. Additional Tables and Discussions

Table 3: Original vs. LLM-Generated Table 2 from Bryson (1996)

	Dislike of Rap, Reggae, Blues/R&B, Jazz, Gospel, Latin		Dislike of the 12 Remaining Genres	
Independent Variable	Original	LLM	Original	LLM
Racism score	.130**	.139*	.080	−.005
Education	−.175***	−.261***	−.242***	−.224***
Household income per capita	−.037	−.034	−.065	−.095
Occupational prestige	−.020	.030	.005	−.012
Female	−.057	−.026	−.070	−.091
Age	.163***	.191***	.126**	.091
Black	−.132***	−.127*	.042	.112
Hispanic	−.058	– (omitted)	−.029	– (omitted)
Other race	−.017	.004	.047	.132*
Conservative Protestant	.063	– (omitted)	.048	– (omitted)
No religion	.057	.079	.024	.080
Southern	.024	.022	.069	.142**
R^2	.145	.190	.147	.166
Adj. R^2	.129	.164	.130	.138
Number of cases	644	327	605	308

Notes: Original values are from Bryson (1996), LLM values are standardized coefficients from the automated replication output. $^{*}p < .05$, $^{**}p < .01$, $^{***}p < .001$ (two-tailed tests).

Table 2 in Bryson (1996) proved the most challenging target, achieving a maximum score of only 45/100 at attempt 6, with 37 of 100 attempts ending in runtime errors. The discrepancy reports identified three persistent issues: sample sizes approximately half the published values (N=327 vs. 644; N=308 vs. 605), Hispanic and No Religion variables dropped due to zero variance, and sign flips in occupational prestige and the racism score for Model 2. These failures likely stem from underspecified coding rules for the racism scale and dependent variable construction. Nevertheless, education, age, and Black coefficients aligned reasonably well in direction and magnitude.

Table 4: LLM-Generated Replication of Table 3 from Bryson (1996)

Attitude		Music Genre					
		Latin	Jazz	Blues/R&B	Show Tunes	Oldies	Classical
(1)	Like very much	85	254	221	235	405	281
(2)	Like it	325	540	669	562	688	478
(3)	Mixed feelings	416	393	367	369	213	371
(4)	Dislike it	403	297	220	281	172	263
(5)	Dislike very much	144	69	61	68	77	136
(M)	Don't know much about it	222	42	57	80	40	66
(M)	No answer	11	11	11	11	11	11
	Mean	3.14	2.61	2.50	2.59	2.25	2.67
		Reggae	Swing	New Age	Opera	Bluegrass	Folk
(1)	Like very much	84	269	48	73	145	130
(2)	Like it	362	588	186	257	562	553
(3)	Mixed feelings	340	290	269	359	411	472
(4)	Dislike it	297	230	429	515	255	274
(5)	Dislike very much	217	53	368	306	59	87
(M)	Don't know much about it	295	165	295	85	163	79
(M)	No answer	11	11	11	11	11	11
	Mean	3.15	2.45	3.68	3.48	2.67	2.76
		Easy Listen.	Pop/Rock	Rap	Heavy Metal	Country	Gospel
(1)	Like very much	251	206	44	48	385	356
(2)	Like it	698	645	159	123	592	571
(3)	Mixed feelings	323	296	284	189	364	364
(4)	Dislike it	200	245	433	400	167	197
(5)	Dislike very much	49	152	614	766	66	71
(M)	Don't know much about it	74	51	61	69	21	36
(M)	No answer	11	11	11	11	11	11
	Mean	2.41	2.67	3.92	4.12	2.32	2.39

Notes: Values are frequency counts from LLM-generated output. Mean is the average rating on the 1–5 scale where 1 = Like very much and 5 = Dislike very much.

Figure 3: Original Table 3 from Bryson (1996)

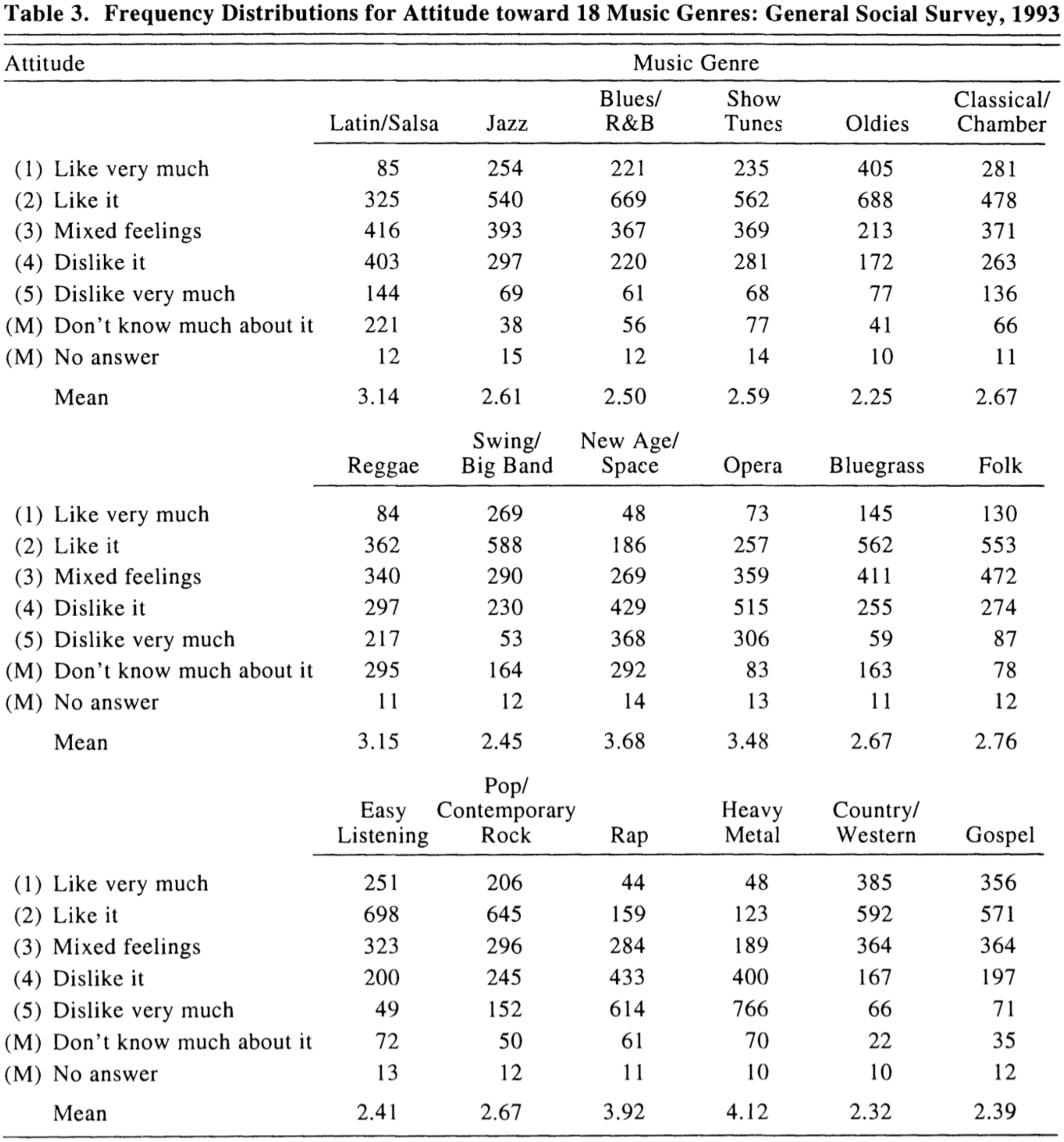


Table 3. Frequency Distributions for Attitude toward 18 Music Genres: General Social Survey, 1993

Attitude	Music Genre					
	Latin/Salsa	Jazz	Blues/ R&B	Show Tunes	Oldies	Classical/ Chamber
(1) Like very much	85	254	221	235	405	281
(2) Like it	325	540	669	562	688	478
(3) Mixed feelings	416	393	367	369	213	371
(4) Dislike it	403	297	220	281	172	263
(5) Dislike very much	144	69	61	68	77	136
(M) Don't know much about it	221	38	56	77	41	66
(M) No answer	12	15	12	14	10	11
Mean	3.14	2.61	2.50	2.59	2.25	2.67
	Reggae	Swing/ Big Band	New Age/ Space	Opera	Bluegrass	Folk
(1) Like very much	84	269	48	73	145	130
(2) Like it	362	588	186	257	562	553
(3) Mixed feelings	340	290	269	359	411	472
(4) Dislike it	297	230	429	515	255	274
(5) Dislike very much	217	53	368	306	59	87
(M) Don't know much about it	295	164	292	83	163	78
(M) No answer	11	12	14	13	11	12
Mean	3.15	2.45	3.68	3.48	2.67	2.76
	Easy Listening	Pop/ Contemporary Rock	Rap	Heavy Metal	Country/ Western	Gospel
(1) Like very much	251	206	44	48	385	356
(2) Like it	698	645	159	123	592	571
(3) Mixed feelings	323	296	284	189	364	364
(4) Dislike it	200	245	433	400	167	197
(5) Dislike very much	49	152	614	766	66	71
(M) Don't know much about it	72	50	61	70	22	35
(M) No answer	13	12	11	10	10	12
Mean	2.41	2.67	3.92	4.12	2.32	2.39

Table 3 in Bryson (1996) achieved the highest score (91/100) among all tables and succeeded on the first attempt for basic frequency counting—the five substantive response categories and means matched immediately. The difficulty arose from the “Don’t know much about it” and “No answer” categories, which caused 46 of 100 attempts to end in runtime errors and showed small count discrepancies in successful runs due to ambiguity in how GSS codes different types of missing data.