Journal of Behavioral Data Science, 2026, *6* (2), 25–66.
DOI: https://doi.org/10.35566/jbds/jikim

Can Large Language Models (LLMs) be Trusted for Power Analysis? An Empirical Evaluation

Hajung Kim[1], Jia Qi[1], Zhe Feng[1], Xiaoya Zhang[2], Yuting Han[3], Jinbo He[4], and Feng Ji[1]*

[1] Department of Applied Psychology and Human Development, University of Toronto, Toronto, Canada
f.ji@utoronto.ca
[2] Department of Family Youth and Community Sciences, University of Florida, Gainesville, USA
[3] School of Psychology, Beijing Language & Culture University, Beijing, P.R. China
[4] Division of Applied Psychology, School of Humanities and Social Science, The Chinese University of Hong Kong, Shenzhen, Guangdong, P.R. China

Abstract. Power analysis is critical for assuring rigor and validity of quantitative research yet remains underutilized due to technical challenges associated with specialized software. At the same time, large language models (LLMs) are being rapidly integrated into research practice, raising interest in their potential to assist statistical and research design tasks. However, despite their widespread adoption, the reliability of LLMs in supporting statistically rigorous procedures has not been systematically evaluated, posing risks for unexamined or overly optimistic use. To address this gap, we evaluated six widely used LLMs—ChatGPT (GPT-3.5, GPT-4, GPT-4o), Llama-3.2-3b, Claude 4.6 Sonnet, and Gemini 2.5 Flash—across three experiments. Experiment 1 examined whether these models could calculate required sample sizes for common statistical tests (two-sample t-test, one-way ANOVA, and χ^2 goodness-of-fit test) under different prompting strategies, including direct calculation versus R/Python code generation. Experiment 2 assessed the tested models' ability to identify missing input parameters necessary for power analysis, which is a task that requires methodological understanding. Experiment 3 extended Experiment 1 to more complex analyses, including Pearson correlation and multiple linear regression. Results showed that GPT-4, GPT-4o, and Claude 4.6 Sonnet performed well when generating R code for sample size estimation but struggled with direct numerical calculation. Most models could identify missing parameters in common power analysis scenarios, although reliability varied across contexts. Similar patterns emerged for the more complex analyses, but with greater variability across models and test types. Findings suggest that the evaluated models may offer support in structuring and initiating power analysis but cannot substitute for expert judgment. Overall, the study underscores the importance of critically evaluating LLM performance in statistically

demanding tasks. Responsible integration of LLM requires critical oversight, cross-verification, and methodological evaluation.

Keywords: Large language models · ChatGPT, Claude, Gemini, Llama · Statistical power analysis · Sample size calculation · Research methods

1 Introduction

Power analysis is a critical step in quantitative social and behavioral research, used to determine the minimum sample size required to detect hypothesized effects and reject null hypotheses with sufficient statistical confidence (e.g., Cohen, 1988). As a critical component of research design (Myors & Murphy, 2023), it has been widely recommended across disciplines such as psychology, education, and biomedical science (e.g., Aberson, 2019; Anderson, Kelley, & Maxwell, 2017; Finch, Cumming, & Thomason, 2001; Ioannidis, 2005; Maxwell, Lau, & Howard, 2015; Wilkinson & on Statistical Inference., 1999). Adequately powered studies yield more reliable and generalizable results, whereas underpowered studies increase the risk of false positives and reduce replicability (Ioannidis, 2005; Maxwell, 2004). Well-conducted power analysis encourages researchers to articulate clear, theoretically informed expectations about effect sizes and study design before data collection begins (Button et al., 2013; Cohen, 1988). By doing so, it helps prevent common pitfalls such as relying on samples that are too small to detect meaningful effects or drawing unwarranted conclusions from statistically non-significant results that are simply underpowered (Ioannidis, 2005; Sedlmeier & Gigerenzer, 1989). In this sense, power analysis serves not only as a technical planning tool but also as a safeguard against biased estimates and irreproducible findings that have contributed to broader concerns about research reliability across disciplines (Button et al., 2013; Fritz, Scherndl, & Kühberger, 2013).

Despite its recognized importance, power analysis remains underutilized in actual research practice. A meta-analysis of over 6,000 published studies found that only 3% reported statistical power (Fritz et al., 2013). This underuse is largely due to two interrelated barriers. The first stems from the technical and cognitive demands of conducting a valid power analysis. Researchers must understand statistical inference, including effect sizes, Type I/II errors, and model-specific parameters (Cohen, 1988; Myors & Murphy, 2023). The second relates to the accessibility and usability of analytical tools. Learning a new statistical program or programming language can be cognitively demanding (Kingsley & Robertson, 2017; Piccolo, Denny, Luxton-Reilly, Payne, & Ridge, 2023), and limited access to software or licenses can pose additional financial constraints. Power analysis is often conducted using tools such as G*Power (Faul, Erdfelder, Lang, & Buchner, 2007), the "pwr" package in R (Champely et al., 2017), SPSS (Corp., 2021), the "sampsi" and "fpower" packages in Stata (StataCorp., 2023), or SAS PROC POWER (Inc., 2023), which require either technical fluency or institutional access. Consequently, these practical barriers to performing accu-

rate power analysis may inadvertently undermine the rigor and replicability of empirical research.

To tackle these challenges and improve the accessibility of power analysis, researchers are increasingly turning to emerging tools capable of handling statistical complexity and reducing technical barriers (Achiam et al., 2023; Frank, 2023; Wang et al., 2023). Recent studies highlight a growing reliance on tools that can support tasks requiring both domain knowledge and reasoning capabilities—areas in which LLMs have shown emerging, but still uneven, performance (T. Zhang et al., 2024). LLMs are currently already being used to assist in a variety of research tasks, including literature synthesis (Jalali & Akhavan, 2024), data interpretation (Khraisha, Put, Kappenberg, Warraitch, & Hadfield, 2024), and inferential analysis (for Social Science & Innovations, 2025).

However, their increasing use in statistical contexts raises concerns about whether such systems can provide accurate and reliable guidance, particularly for tasks such as power analysis that require precise specification of assumptions and parameters. Although LLMs are trained on statistical literature and can generate and explain procedures, it remains unclear whether these capabilities translate into methodologically sound and trustworthy outputs in practice. Meanwhile, workshops and tutorials promoting LLMs-based research design and data analysis workflows are rapidly emerging (e.g., for Social Science & Innovations, 2025; in Europe, 2025). While these initiatives often emphasize practical implementation, they appear to offer limited systematic empirical validation, particularly for tasks requiring methodological rigor. Existing studies have begun to explore these issues but are often based on narrow task scopes or specific prompting strategies (e.g., Methnani, Latiri, Dergaa, Chamari, & Saad, 2023), leaving important questions unresolved. The growing adoption of LLMs alongside limited empirical evaluation suggests the need for more systematic and critical examination. As noted by prior work, the uncritical integration of LLMs into research workflows may pose risks to established scientific standards (Birhane, Kasirzadeh, Leslie, & Wachter, 2023). Accordingly, further empirical investigation is warranted to better understand the capabilities and limitations of LLMs in statistically demanding tasks.

Given these considerations and potential methodological risks, we conducted a systematic evaluation of LLMs' performance in power analysis. Specifically, we tested six LLMs, including ChatGPT (GPT-3.5, GPT-4, GPT-4o), Llama 3.2-3b, Claude 4.6 Sonnet, and Gemini 2.5 Flash, focusing on their ability to interpret problem descriptions and generate analytic code. Three experiments were designed to critically assess their effectiveness. Experiment 1 evaluated whether each model could correctly compute required sample sizes for common statistical tests using multiple prompting strategies. Experiment 2 examined whether the models could accurately identify and explain missing or incomplete input information necessary for valid power analyses. Experiment 3 extended Experiment 1 by evaluating model performance on more computationally complex statistical tests, specifically Pearson correlation and multiple linear regression. By examining both task performance and response consistency, this study seeks to provide

an empirical evaluation of the capabilities and limitations of LLMs and to inform discussions about their appropriate use in quantitative research contexts.

1.1 Research Questions

Power analysis can serve multiple purposes, such as determining the necessary sample size, or performing post hoc analyses (Aberson, 2019). In this study, we specifically focus on sample size calculation, given its common use in research design. Thus, our first overarching research question (RQ1) addresses whether LLMs can accurately calculate sample sizes for commonly used statistical tests. To examine this question, we structured our investigation in three progressive steps.

First, recognizing that different types of LLMs vary significantly in performance and reliability, we evaluated the accuracy of sample size calculations across test types and models. The selected statistical tests (two-sample t-test, one-way ANOVA, and χ^2 goodness-of-fit test) were selected as baseline evaluation tasks because they are widely used in applied research and involve relatively well-defined analytical procedures. To capture the variation across model families and capability levels, we examined six LLMs in this study: GPT-3.5-turbo, GPT-4, and GPT-4o (developed by OpenAI), Llama 3.2-3B (developed by Meta), Claude 4.6 Sonnet (developed by Anthropic), and Gemini 2.5 Flash (developed by Google). These models were selected to represent a range of capability levels and model ecosystems rather than to endorse any particular system. The three ChatGPT versions reflect different stages within a widely used proprietary model family, with GPT-3.5-turbo serving as a commonly used baseline, GPT-4 as a higher-capability model, and GPT-4o as the most recent optimized version available at the time of data collection. Claude 4.6 Sonnet and Gemini 2.5 Flash were included to broaden coverage across major proprietary LLM ecosystems, whereas Llama 3.2-3B was included as an open-source alternative.

Next, considering existing evidence that LLMs often struggle with direct numerical calculations (e.g., Frieder et al., 2023; Jiang et al., 2024; Mirzadeh et al., 2024) but may perform more effectively when generating code (e.g., Biswas, 2023; Liu, Xia, Wang, & Zhang, 2024; Nigar & Mohammed, 2023), we explored whether prompting LLMs to generate code rather than directly calculate numerical results would improve their accuracy (RQ1b). To explore different approaches to code generation, we included both R and Python in the experiment. By including both languages, we aimed to examine whether specifying programming language in the prompt affects the accuracy of sample size calculation tasks.

Furthermore, our second overarching research question (RQ2) addresses another potential barrier to conducting power analysis: identifying the specific statistical parameters required for performing these calculations. Effective power analysis necessitates providing a complete and precise set of inputs, yet determining these inputs is often challenging without sufficient methodological training (Cohen, 1988; Myors & Murphy, 2023). Unlike structured software tools that enumerate required fields visually (e.g., G*Power), natural-language interactions

with LLMs leave the completeness of a user's specification ambiguous, potentially exacerbating existing concerns about the misuse or misapplication of power analysis in empirical research (Fritz et al., 2013). Although prior work suggests that LLMs may be capable of flagging missing or ambiguous information in user prompts (T. C. Chen et al., 2023), it remains unclear whether they can do so accurately and consistently in the context of power analysis, where omission can fundamentally invalidate results. Thus, we examined whether LLMs can actually recognize and inform users about these critical missing elements required for valid power analysis.

Finally, although the initial experiments focused on relatively common statistical tests (two-sample t-test, one-way ANOVA, and χ^2 goodness-of-fit test), many real-world research settings involve more computationally demanding analyses. Therefore, we further examined whether the performance patterns observed in the initial experiments would generalize to more complex statistical procedures, specifically focusing on Pearson correlation and multiple linear regression (RQ3). This extension allowed us to assess whether the performance patterns observed in the initial experiments would generalize beyond the initial set of statistical tests.

In sum, we formulated the following research questions:

Research Question 1: Can LLMs perform sample size calculation for two-sample t-test, one-way ANOVA, and χ^2 goodness-of-fit test?
- **Research Question 1a:** Do different types of LLMs affect the accuracy of sample size calculation?
- **Research Question 1b:** Does requesting code through LLMs yield higher accuracy than directly asking to calculate sample sizes?

Research Question 2: Can LLMs identify missing information necessary for power analysis?

Research Question 3: Can LLMs perform sample size calculations for Pearson correlation and multiple linear regression?

2 Experiment 1: Sample Size Calculation

2.1 Method

Experimental Design and Procedure Experiment 1 aimed to evaluate LLMs' accuracy in determining the required sample size. We used a $6 \times 3 \times 3$ factorial experimental design, incorporating three experimental factors: "models," "methods," and "test types."

First, the "model" factor included six conditions: using the GPT-3.5-turbo, GPT-4, GPT-4o (OpenAI), Llama 3.2-3b (Meta), Claude 4.6 Sonnet (Anthropic), and Gemini 2.5 Flash (Google). The specific model versions evaluated in this study were GPT-3.5-turbo (gpt-3.5-turbo-0125), GPT-4 (gpt-4-0613), GPT-4o (gpt-4o-2024-08-06), Llama 3.2-3b (Llama 3.2-3b-Instruct), Claude 4.6 Sonnet (claude-sonnet-4-6), and Gemini 2.5 Flash (gemini-2.5-flash). The initial set of models (GPT-3.5-turbo, GPT-4, GPT-4o, and Llama 3.2-3b) was evaluated in

January 2025. Claude 4.6 Sonnet and Gemini 2.5 Flash were added in April 2026 as part of an expansion phase intended to incorporate newer frontier models and broaden the cross-platform comparison. Model identifiers and query periods are reported to support reproducibility and transparency. The models were selected to represent a cross-tier spectrum of LLMs, including a lightweight open-source model, widely used proprietary models, and newer frontier systems from multiple providers. This comparison was designed to test **Research Question 1a**.

Second, the "methods" factor had three conditions: (1) LLMs directly computing the required sample size, (2) LLMs generating R code to calculate the sample size, and (3) LLMs generating Python code for the same task. These conditions were used to test **Research Question 1b**.

Lastly, we considered three types of statistical tests in the "test types" factor: the two-sample t-test (Student's t-test), the one-way ANOVA (Fisher's F test), and the χ^2 goodness-of-fit test (Pearson's χ^2 test), which are commonly used in basic statistical research design. 100 sets of parameters (e.g., significance level, effect size, power) were generated for each test type to be used as a testing sample for each mode type to ensure reliability of the estimate for statistical testing. A priori power analysis using discordant pair probabilities of 0.3 and 0.1 showed that 74 pairs were sufficient to achieve 80% power at $\alpha = 0.05$, supporting our choice of 100 repetitions per condition. Combining these results, we will examine the findings for Research Question 1—whether LLMs can calculate sample sizes in power analysis.

The procedures for each test type and scenario are illustrated in Figure 1. The conversational structure used in the experiment is outlined in Table 1, and a specific example of the experiment is provided in Figure 2.

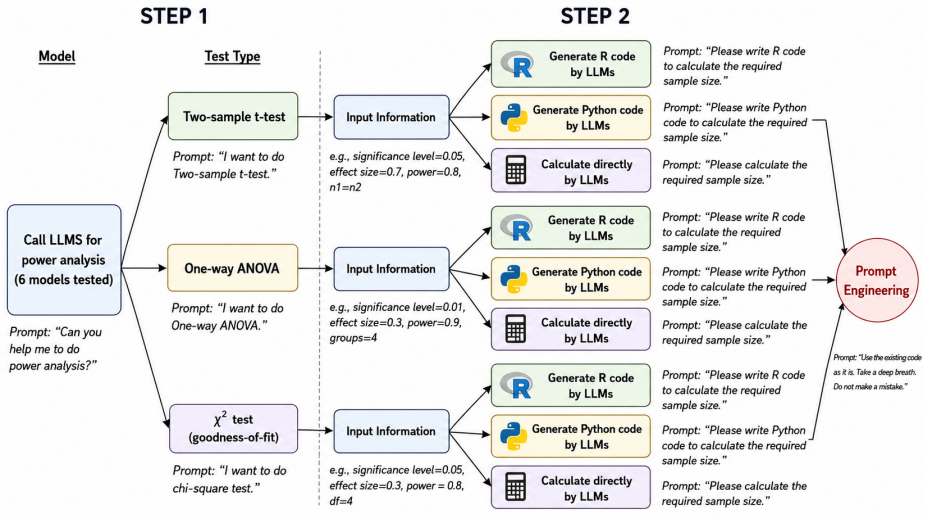


Figure 1. The Experimental Design and Procedures of Experiment 1

Table 1. The Conversational Structure of Experiment 1

Step	Role	Message
Step 1	User	Can you help me do power analysis? I want to do [*Test type*].
	LLMs	[*Guide information required for power analysis*]
Step 2	User	[*Input information*] (e.g.) significance level=0.01, effect size=0.25, power=0.9 [*Ask calculation*] Please (write code to) calculate the required sample size. [*Prompt engineering*] Use the existing code as it is. Take a deep breath. Do not make a mistake.
	LLMs	[*Return results*]

Experimental Setup In this study, we designed and implemented the experiment using the OpenAI API (for GPT models), the Anthropic API (for Claude 4.6 Sonnet), the Google AI API (for Gemini 2.5 Flash), and the Llama API (for Llama 3.2-3b). First, we developed a question list based on the conversational structure from Experiment 1. Next, we integrated each statistical test type with the relevant information into this question list, which was then input into the API. We used the API versions of these models to enable systematic and reproducible testing, allowing us to run 100 trials per condition under controlled settings. The API was used solely to standardize evaluation and does not represent a fundamentally different user workflow. In practice, researchers would more commonly interact with LLMs through conversational interfaces rather than directly through an API. However, the prompts and responses in our experiments followed a conversational format similar to publicly available chat interfaces such as ChatGPT, making the interaction structure broadly comparable to typical real-world use.

We also compiled a list of the information necessary for conducting the tests. For significance levels, we used two significance levels: $\alpha = 0.01$ and $\alpha = 0.05$. Regarding effect sizes, we arbitrarily selected a number from Cohen (1988)'s suggested criteria for small, medium, and large effect sizes, using the following set: 0.15, 0.20, 0.25, 0.28, 0.30, 0.35, 0.50, 0.60, and 0.70. Additionally, we chose three desired power levels: 0.75, 0.8, and 0.9, based on Cohen's recommended power levels. In total, these parameter combinations yielded 100 cases, which were submitted to the API.

Depending on the test being conducted, we specified additional parameters. For the two-sample t-test, we indicated whether it was one-tailed or two-tailed. We also provided information that Group 1 and Group 2 had equal sample sizes by specifying the allocation ratio (N1=N2), focusing on this condition to examine a simplified scenario and not considering unequal group sizes. For the one-way ANOVA test, we included the number of groups as an essential variable for power analysis, using group counts of 3, 4, 6, and 10, as drawn from examples provided in Cohen (1988). In the case of the χ^2 test, we input the degrees of

Can you help me conduct a power analysis for a two-sample t-test? I am comparing two independent groups and would like to determine the required sample size.

Absolutely! To conduct a **power analysis for a two-sample t-test**, we'll need a few key parameters. Please provide the following:

1. **Effect size (Cohen's d)** – A measure of the difference between the two group means in standard deviation units. If you're not sure, I can help estimate it based on means and standard deviations.
2. **Significance level (alpha)** – Commonly set at 0.05.
3. **Desired power (1 - beta)** – Typically set at 0.80 or 0.90, representing an 80% or 90% chance of detecting an effect if it exists.
4. **Allocation ratio** – Whether the sample sizes for the two groups are equal (ratio = 1) or unequal.
5. **One-tailed or two-tailed test** – Most tests are two-tailed unless you have a specific directional hypothesis.

If you provide those details, I can calculate the required sample size for each group.

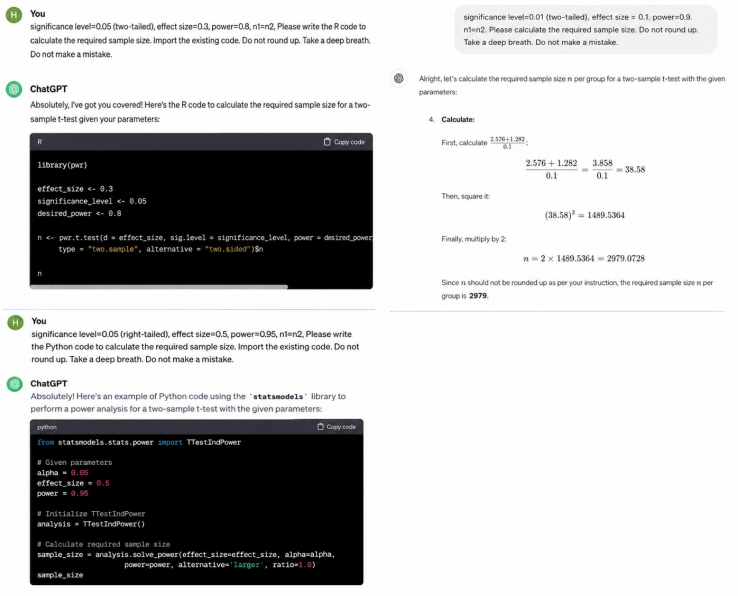


Figure 2. The Examples of Experiment 1

freedom as necessary information, using values of 2, 4, 9, and 20, also based on Cohen (1988).

We systematically combined all these variables to create different condition sets and randomly generated 100 parameter sets within each condition. The details of the information used in the experiment are presented in Table 2.

Table 2. Types of Statistical Tests and Information Used in the Experiment

Test type	Information
Two sample t-test	Significance level, One-tailed / Two-tailed, Desired power, Effect size, The ratio of N1/N2
One-way ANOVA	Significance level, Desired power, Effect size, The number of groups
χ^2 test (goodness-of-fit)	Significance level, Desired power, Effect size, Degree of freedom

Additionally, to minimize randomness due to the probabilistic nature of generative AI models, we set the temperature parameter to 0. The temperature is a tuning parameter in many foundational models that controls the "creativity" or randomness of generated outputs. It typically ranges from 0 to 1, with lower values leading to more deterministic and repetitive outputs, and higher values producing more diverse and creative but potentially less stable outputs. A temperature of 0 ensures maximum determinism, where the model outputs the token with the highest probability based on its training data. Our goal was to prioritize stable and replicable outcomes over creativity, which is why we chose this minimum setting.

In the first step of the experiment, we initiated each dialogue with the same prompt: "Can you help me do power analysis? I want to do [*test type*]." This provided context for LLMs to guide users in gathering the necessary information for conducting power analysis for the specified test. Once this context was established, the second step of the experiment varied based on the scenario (see Figure 1). We input different combinations of information from the compiled list in subsequent prompts, asking LLMs to either directly calculate the sample size or generate the required R or Python code to do so. With the temperature set to 0, the outputs remained stable and consistent in most cases. Based on the given context, LLMs generated appropriate responses to the provided information.

At the end of the second conversation, we added several sentences to facilitate prompt engineering and improve accuracy. For example, to minimize potential randomness, we used phrases like "Use the existing code as it is" and "Do not round up." This was because LLMs occasionally round numbers or extracts only the most convenient information for users, which can introduce unnecessary errors. Our aim was to ensure consistent use of basic code. Additionally, commands like "Take a deep breath. Do not make a mistake." leveraged the model's zero-shot capabilities (Kojima, Gu, Reid, Matsuo, & Iwasawa, 2023) to further enhance

accuracy. These strategies were applied throughout the experiment whenever we asked LLMs to calculate or generate code. Overall, the prompts were kept simple and consistent to reflect typical researcher-LLM interactions rather than relying on heavily engineered prompts, allowing us to evaluate model performance under realistic usage conditions and attribute observed differences to the models themselves rather than to prompt variation.

Analyses To evaluate performance, we defined accuracy as the number of times the required sample size was correctly calculated, divided by the total number of trials. When LLMs were asked to generate R or Python code, we executed the code to verify that it ran successfully and produced the correct output. We also calculated the standard error and confidence intervals to provide uncertainty estimates for the accuracy due to the limited number of replications.

Our analysis focused on the effects of the manipulations, including models, test types, and sample size calculation methods, on accuracy. To compare our experimental conditions, we applied pairwise McNemar tests to evaluate the differences between models, test types, and sample size calculation methods. Specifically, we compared (a) LLMs (GPT-3.5-turbo, GPT-4, GPT-4o, Llama 3.2-3b, Claude 4.6 Sonnet, and Gemini 2.5 Flash) while holding the statistical test type and sample size calculation approach constant; (b) statistical test types (t-test, one-way ANOVA, and chi-square test) while holding the LLM and calculation approach constant; and (c) sample size calculation approach (direct LLM-generated calculations, R-based code, and Python-based code) while holding the LLM and test type constant. Since our outcome was binary (coded as 1 for success and 0 for failure), we used the nonparametric pairwise McNemar test to determine whether the differences between pairs were statistically significant. We used the 'rcompanion' library in R for the pairwise McNemar test. To control for multiple comparisons, we adjusted the p-values using the Benjamini–Hochberg procedure for controlling the False Discovery Rate (FDR), ensuring that the overall rate of false discoveries remained below a specified threshold. The threshold for FDR-corrected significance, above which variables are rejected, is set to 0.05 (Benjamini & Hochberg, 1995).

2.2 Results

In Experiment 1, the results indicate that the GPT-4, GPT-4o, and Claude 4.6 Sonnet models perform best when generating R code to compute the required sample size with an accuracy of 1.00 (100.00%). The results of Experiment 1 are presented in Table 3 and Figure 3.

Table 3: Results of Experiment 1

			Two sample t-test	One-way ANOVA	χ^2 test (goodness-of-fit)	Total
R	GPT-3.5	Result	100 / 100	96 / 100	100 / 100	296 / 300
		Accuracy (CI)	1.000	.960 (.921, .999)	1.000	.987 (.973, 1.000)
	GPT-4	Result	100 /100	100 /100	100 / 100	300 / 300
		Accuracy (CI)	1.000	1.000	1.000	1.000
	GTP-4o	Result	100 / 100	100 / 100	100 / 100	300 / 300
		Accuracy (CI)	1.000	1.000	1.000	1.000
	Llama3.2-3b	Result	0 / 100	0 / 100	0 / 100	0 / 300
		Accuracy (CI)	.000	.000	.000	.000
	Claude 4.6 Sonnet	Result	100 / 100	100 / 100	100 / 100	300 / 300
		Accuracy (CI)	1.000	1.000	1.000	1.000
	Gemini 2.5 Flash	Result	100 / 100	72 / 100	73 / 100	245 / 300
		Accuracy (CI)	1.000	.720 (.632, .808)	.730 (.643, .817)	.817 (.773, .860)
Python	GPT-3.5	Result	100 / 100	76 / 100	0 / 100	176 / 300
		Accuracy (CI)	1.000	.760 (.675, .845)	.000	.587 (.530, .644)
	GPT-4	Result	100 / 100	100 / 100	49 / 100	249 / 300
		Accuracy (CI)	1.000	1.000	.490 (.390, .590)	.830 (.787, .873)
	GPT-4o	Result	100 / 100	100 / 100	0 / 100	200 / 300
		Accuracy (CI)	1.000	1.000	.000	.667 (.612, .721)
	Llama3.2-3b	Result	0 / 100	0 / 100	0 / 100	0 / 300
		Accuracy (CI)	.000	.000	.000	.000
	Claude 4.6 Sonnet	Result	100 / 100	0 / 100	97 / 100	197 / 300
		Accuracy (CI)	1.000	.000	0.970 (.937, 1.000)	0.657 (.603, .710)
	Gemini 2.5 Flash	Result	100 / 100	87 / 100	0 / 100	187 / 300

Continued on ne

Table 3 (continued)

			Two sample t-test	One-way ANOVA	χ^2 test (goodness-of-fit)		Total
		Accuracy (CI)	1.000	.870 (.804, .936)	.000	.623 (.569, .678)	
LLMs	GPT-3.5	Result	0 / 100	0 / 100	0 / 100	0 / 300	
		Accuracy (CI)	.000	.000	.000	.000	
	GPT-4	Result	1 / 100	0 / 100	0 / 100	1 / 300	
		Accuracy (CI)	.010 (.000, .030)	.000	.000	.003 (.000, .010)	
	GPT-4o	Result	41 / 100	18 / 100	2 / 100	61 / 300	
		Accuracy (CI)	.410 (.312, .508)	.180 (.103, .257)	.020 (.000, .048)	.203 (.157, .250)	
	Llama3.2-3b	Result	0 / 100	0 / 100	0 / 100	0 / 300	
		Accuracy (CI)	.000	.000	.000	.000	
	Claude 4.6 Sonnet	Result	50 / 100	0 / 100	0 / 100	50 / 300	
		Accuracy (CI)	.500 (.402, .598)	.000	.000	.167 (.124, .209)	
	Gemini 2.5 Flash	Result	60 / 100	0 / 100	0 / 100	60 / 300	
		Accuracy (CI)	.600 (.504, .696)	.000	.000	.200 (.155, .245)	

Note. CI = Confidence Interval. R = R code condition; Python = Python code condition; Direct = direct calculation condition.

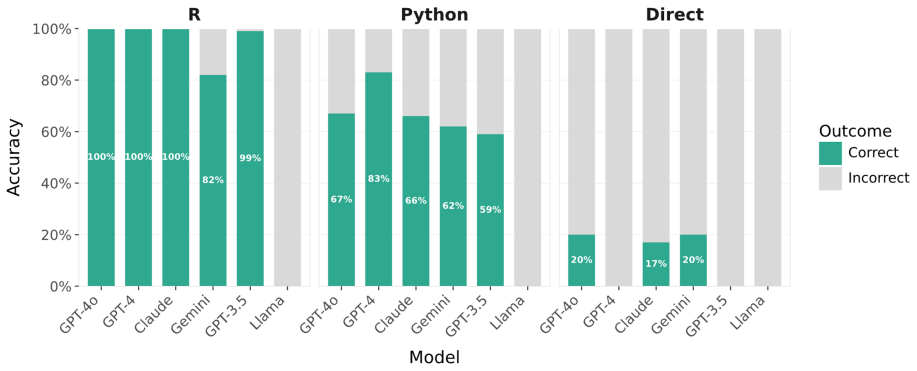


Figure 3. Results of Experiment 1

To answer Research Question 1, GPT-4, GPT-4o, and Claude 4.6 Sonnet achieved perfect accuracy when generating R code for sample size calculation, demonstrating that LLMs can successfully perform power analysis when appropriate coding strategies are used. GPT-3.5-turbo also showed strong performance, whereas Gemini 2.5 Flash exhibited occasional failures in ANOVA and χ^2 scenarios. Overall, contemporary proprietary LLMs were generally capable of generating functional code for statistical power analysis, although performance varied across statistical tests and programming environments.

Python-based code generation was also generally successful but yielded lower performance than R, particularly for the χ^2 goodness-of-fit test. One contributing factor is that Python and R implement χ^2 power analysis differently. In Python, the relevant function requires the number of categories (n_bins), whereas R and G*Power use degrees of freedom as input. Models frequently supplied degrees of freedom to Python functions, leading to execution failures or incorrect results. In contrast, direct calculation was largely unsuccessful across models, consistent with previous findings (e.g., Methnani et al., 2023). Rather than producing correct sample sizes, models often generated incorrect numerical answers or provided procedural instructions for conducting the analysis in software such as G*Power.

When models failed, their errors generally fell into three categories. The first was function hallucination, in which models invoked functions that do not exist in the relevant package or selected an inappropriate function for the requested analysis. The second was wrong input/output specification, in which models correctly identified the relevant function but supplied unsupported arguments or incorrectly handled the function's inputs or outputs. The third was wrong calculation, in which models produced an incorrect numerical result despite identifying the appropriate analytical procedure. Unlike function hallucination or input/output errors, wrong-calculation errors often generated plausible numerical answers with no obvious indication that the result was incorrect, making

them particularly difficult for users to detect. Representative examples of each error type are provided in **Appendix A**.

Pairwise McNemar Test We conducted pairwise comparisons for each condition to evaluate our research questions. Because performance was operationalized as a binary outcome (coded as 1 for success and 0 for failure), we used pairwise McNemar tests to compare between conditions. These tests were applied by pairing outcomes within the same condition while holding the remaining factors constant, which allows us to assess whether success rates differed significantly between pairs of models, test types, or calculation approaches.

For the model comparison, GPT-4, GPT-4o and Claude Sonnet 4.6 performed equivalently ($p = .337$, .449, and .833 respectively). Three models significantly outperformed GPT-3.5-turbo and Gemini 2.5 Flash ($p < .001$). GPT-3.5-turbo and Gemini 2.5 Flash did not differ significantly from each other ($p = .118$), suggesting similar intermediate-level performance. These findings indicate that model type substantially influenced sample size calculation accuracy, supporting **Research Question 1a**.

For the method comparison, instructing GPT to generate R code was more effective than generating Python code ($p < .001$), and generating Python code was more effective than direct calculation ($p < .001$). These results support **Research Question 1b** by demonstrating that code-generation prompting strategies yield higher accuracy than direct numerical calculation.

For the test-type comparison, all pairwise comparisons were significant. The two-sample t-test yielded higher accuracy than both one-way ANOVA ($p < .001$) and the χ^2 test ($p < .001$), and ANOVA also outperformed the χ^2 test ($p < .001$), indicating that LLM performance varied systematically across statistical test types.

The pairwise McNemar test was used to compare the results of each paired experimental condition for paired categorical groups, given the binomial nature of the outcomes. The results of the pairwise McNemar tests are shown in Table 4. In Table 6, we report both original (unadjusted) p-values and the FDR-adjusted p-values, using the Benjamini–Hochberg procedure (Benjamini & Hochberg, 1995). Presenting both values enhances transparency and interpretability, helping to assess the impact of the adjustment on the significance of the results.

Table 4. The Results of the Pairwise McNemar Tests

Comparison	***Z***	***p***	***p-adjusted***	**Significance**
Model Comparison				
GPT-4o vs. GPT-4	1.054	.292	.337	
GPT-4o vs. Claude	0.858	.391	.419	
GPT-4o vs. Gemini	5.939	$< .001$	$< .001$	***
GPT-4o vs. GPT-3.5	9.434	$< .001$	$< .001$	***
GPT-4o vs. Llama	23.685	$< .001$	$< .001$	***
GPT-4 vs. Claude	0.211	.833	.833	
GPT-4 vs. Gemini	4.372	$< .001$	$< .001$	***
GPT-4 vs. GPT-3.5	8.832	$< .001$	$< .001$	***
GPT-4 vs. Llama	23.452	$< .001$	$< .001$	***
Claude vs. Gemini	3.485	$< .001$	$< .001$	***
Claude vs. GPT-3.5	4.978	$< .001$	$< .001$	***
Claude vs. Llama	23.388	$< .001$	$< .001$	***
Gemini vs. GPT-3.5	1.655	.098	.122	
Gemini vs. Llama	22.181	$< .001$	$< .001$	***
GPT-3.5 vs. Llama	21.726	$< .001$	$< .001$	***
Method Comparison				
R vs. Python	19.759	$< .001$	$< .001$	***
R vs. Direct	35.623	$< .001$	$< .001$	***
Python vs. Direct	28.862	$< .001$	$< .001$	***
Test Type Comparison				
t-test vs. ANOVA	17.407	$< .001$	$< .001$	***
t-test vs. chi-sq	23.043	$< .001$	$< .001$	***
ANOVA vs. chi-sq	10.585	$< .001$	$< .001$	***

Note. $p < .001$ ***

3 Experiment 2: Identifying Missing Information

3.1 Method

Experimental Design and Procedure In Experiment 2, our goal was to examine whether LLMs could identify missing information required for power analysis and assist users in completing the process. We deliberately omitted one piece of information and assessed whether LLMs could recognize it, then calculated its corresponding accuracy. The experiment followed the same model settings (GPT-3.5-turbo, GPT-4, GPT-4o, and Llama 3.2-3b) and data collection procedures as Experiment 1, with Claude 4.6 Sonnet and Gemini 2.5 Flash added during a later expansion phase in April 2026.

As in Experiment 1, Experiment 2 used the same statistical tests and input information: the two-sample t-test, one-way ANOVA, and χ^2 goodness-of-fit test, as detailed in Table 2. The first part of the dialogue followed the same structure as in Experiment 1. In the second part, we sequentially omitted one piece of information from Table 2 (such as significance level, effect size, power, or another key parameter) and asked LLMs if there was missing information. Each scenario, with one missing piece of information, was tested 100 times. The conversational structure is outlined in Table 5, and the procedures, along with examples of inputs used in the prompts, are shown in Figures 4 and 5.

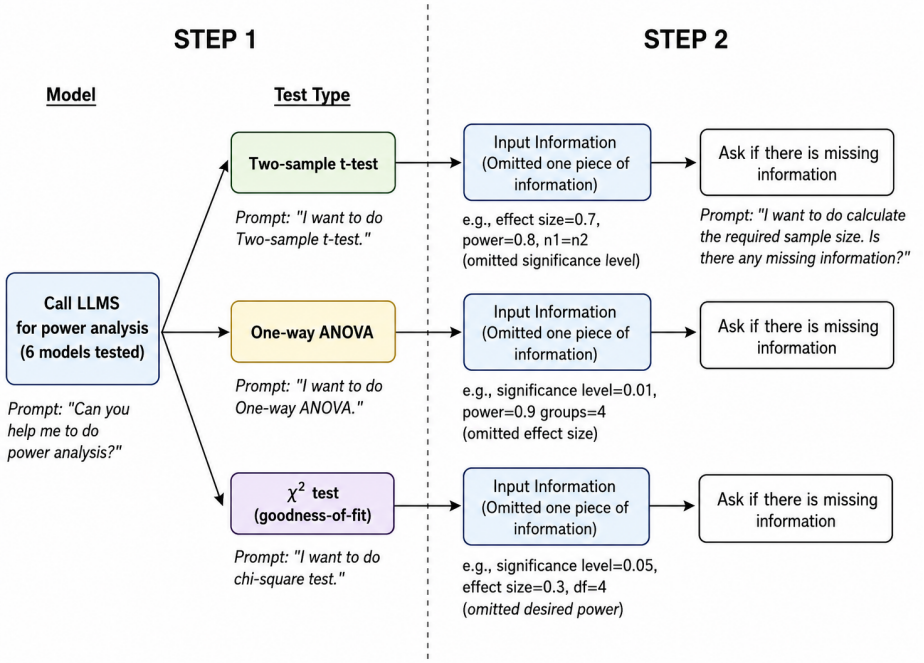


Figure 4. The Experimental Design and Procedures of Experiment 2

Table 5. The Conversational Structure of Experiment 2

Step	Role	Message
Step 1	User	Can you help me do power analysis? I want to do [*test type*].
	ChatGPT	[*Guide information required for power analysis*]
Step 2	User	[*Input Information (omitted one piece of information)*] I want to calculate the required sample size. Is there any missing information?
	ChatGPT	[*Return results*]

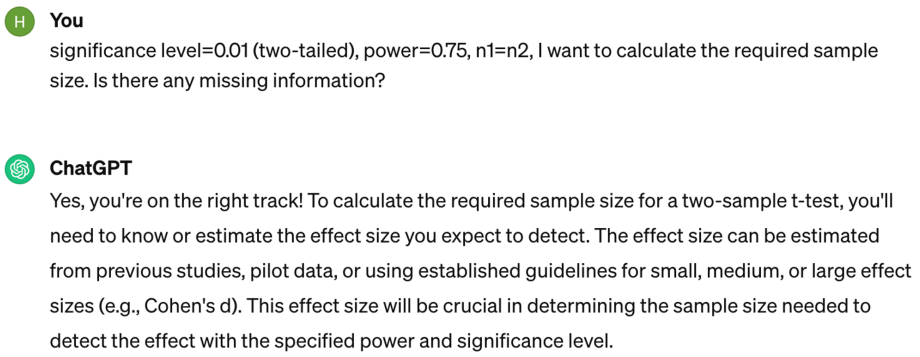


Figure 5. The Examples of Experiment 2 (Step 2)

Note. Step 1 was identical to Experiment 1.

Experimental Setup We used the same experimental setup as in Experiment 1. This experiment was conducted using the OpenAI API (for GPT models), the Anthropic API (for Claude 4.6 Sonnet), the Google AI API (for Gemini 2.5 Flash), and the Llama API (for Llama 3.2-3b). First, we created a question list based on the conversational structure from Experiment 2. Next, we compiled the list by pairing each statistical test type with its corresponding information, omitting one piece of data. The compiled question list was then input into the API. To perform the experiment, we set the temperature to 0, applied prompt engineering techniques, and used the initial dialogue as the default, consistent with the procedures in Experiment 1.

Analyses To assess performance, we calculated accuracy by evaluating the correctness of results after conducting 100 trials for each condition. Accuracy was defined as the proportion of correctly identified missing components or pieces of information required for conducting a valid sample size calculation, and it was calculated as the number of correct identifications divided by the total number

of trials. Additionally, we computed the standard error and confidence intervals for the accuracy estimates, taking into account the limited number of replications. For Experiment 2, the focus was on parameter-specific accuracy, for which descriptive analysis with uncertainty estimates was deemed sufficient without testing differences.

3.2 Results

To address Research Question 2, we examined whether LLMs could identify missing information required for valid power analyses. Especially, when significance level, effect size, and power—key components for conducting power analysis—were omitted, GPT-4o achieved the highest accuracy of 1.00 (100%), followed by GPT-3.5-turbo at 0.99 (98.6%), Gemini 2.5 Flash at 0.97 (97.3%), Claude 3.7 Sonnet at 0.97 (96.6%), and GPT-4 at 0.96 (95.9%). Llama showed substantially lower accuracy (51.78%) across most conditions. The results of Experiment 2 are presented in Table 6 and Figure 6.

Performance varied depending on the parameter being omitted. Missing significance levels, effect sizes, desired power values, and test-specific parameters (e.g., number of groups or degrees of freedom) were generally identified by LLMs. In contrast, missing tail specification for the two-sample t-test proved particularly challenging. Claude 3.7 Sonnet correctly identified the missing tail type in all cases, whereas other models frequently failed to recognize the omission and instead implicitly assumed a two-tailed test. Similarly, missing allocation ratios were detected reliably by Claude 3.7 Sonnet, GPT-4, and GPT-4o but less consistently by Gemini 2.5 Flash and Llama.

When models failed to identify missing information, they often proceeded as though all required inputs had been provided. Common responses included statements such as "There is no missing information" or direct attempts to perform the calculation without first verifying parameter completeness. In some cases, models substituted default assumptions rather than explicitly requesting the missing information. For example, omitted significance levels were often assumed to be .05, desired power was frequently assumed to be .80, and missing tail specifications were commonly treated as two-tailed tests. In the χ^2 condition, some models incorrectly inferred that expected frequencies were missing when the actual missing parameter was desired power. Representative examples of model responses that failed to identify missing parameters are provided in **Appendix B**.

These findings suggest that contemporary LLMs are generally capable of identifying missing information in common power-analysis scenarios, particularly for standard parameters such as significance level, effect size, and desired power. However, performance declines when omitted parameters are less salient or more context dependent, such as tail specification or allocation ratio. Consequently, users should not assume that LLMs will reliably detect all missing inputs and may benefit from explicitly asking models to verify parameter completeness before conducting power analyses.

Table 6: Results of Experiment 2

Test type	Model		Two sample t-test	One-way ANOVA	χ^2 test
Significance level	GPT-3.5-turbo	Result	100 / 100	100 / 100	100 / 100
		Accuracy (CI)	1.000	1.000	1.000
	GPT-4	Result	100 / 100	100 / 100	68 / 100
		Accuracy (CI)	1.000	1.000	.680 (.587, .773)
	GPT-4o	Result	100 / 100	100 / 100	100 / 100
		Accuracy (CI)	1.000	1.000	1.000
	Llama 3.2-3b	Result	46 / 100	35 / 100	58 / 100
		Accuracy (CI)	.460 (.360, .560)	.350 (.255, .445)	.580 (.481, .679)
	Claude 3.7 Sonnet	Result	100 / 100	100 / 100	95 / 100
		Accuracy (CI)	1.000	1.000	.950 (.907, .993)
	Gemini 2.5 Flash	Result	100 / 100	100 / 100	92 / 100
		Accuracy (CI)	1.000	1.000	.920 (.867, .973)
Effect size	GPT-3.5-turbo	Result	100 / 100	100 / 100	100 / 100
		Accuracy (CI)	1.000	1.000	1.000
	GPT-4	Result	100 / 100	100 / 100	100 / 100
		Accuracy (CI)	1.000	1.000	1.000
	GPT-4o	Result	100 / 100	100 / 100	100 / 100
		Accuracy (CI)	1.000	1.000	1.000
	Llama 3.2-3b	Result	85 / 100	94 / 100	88 / 100
		Accuracy (CI)	.850 (.779, .921)	.940 (.893, .987)	.880 (.815, .945)
	Claude 3.7 Sonnet	Result	100 / 100	100 / 100	100 / 100
		Accuracy (CI)	1.000	1.000	1.000
	Gemini 2.5 Flash	Result	100 / 100	100 / 100	100 / 100

Continued on

Table 6 (continued)

Test type	Model		Two sample t-test	One-way ANOVA	χ^2 test
		Accuracy (CI)	1.000	1.000	1.000
Desired power	GPT-3.5-turbo	Result	100 / 100	100 / 100	87 / 100
		Accuracy (CI)	1.000	1.000	.870 (.803, .937)
	GPT-4	Result	100 / 100	100 / 100	95 / 100
		Accuracy (CI)	1.000	1.000	.950 (.906, .994)
	GPT-4o	Result	100 / 100	100 / 100	100 / 100
		Accuracy (CI)	1.000	1.000	1.000
	Llama 3.2-3b	Result	28 / 100	13 / 100	19 / 100
		Accuracy (CI)	.280 (.190, .370)	.130 (.063, .197)	.190 (.112, .268)
	Claude 3.7 Sonnet	Result	100 / 100	75 / 100	99 / 100
		Accuracy (CI)	1.000	.750 (.665, .835	.990 (.970, 1.000)
	Gemini 2.5 Flash	Result	100 / 100	100 / 100	82 / 100
		Accuracy (CI)	1.000	1.000	.820 (.745, .895)
Two-tailed	GPT-3.5-turbo	Result	2 / 100	-	-
		Accuracy (CI)	.020 (.000, .048)	-	-
	GPT-4	Result	0 / 100	-	-
		Accuracy (CI)	.000	-	-
	GPT-4o	Result	26 / 100	-	-
		Accuracy (CI)	.260 (.172, .348)	-	-
	Llama 3.2-3b	Result	0 / 100	-	-
		Accuracy (CI)	.000	-	-
	Claude 3.7 Sonnet	Result	100 / 100	-	-
		Accuracy (CI)	1.000	-	-
	Gemini 2.5 Flash	Result	12 / 100	-	-

Continued on

Table 6 (continued)

Test type	Model		Two sample t-test	One-way ANOVA	χ^2 test
		Accuracy (CI)	.120 (.056, .184)	-	-
Allocation ratio (N1=N2)	GPT-3.5-turbo	Result	50 / 100	-	-
		Accuracy (CI)	.500 (.400, .600)	-	-
	GPT-4	Result	92 / 100	-	-
		Accuracy (CI)	.920 (.866, .974)	-	-
	GPT-4o	Result	91 / 100	-	-
		Accuracy (CI)	.910 (.853, .967)	-	-
	Llama 3.2-3b	Result	3 / 100	-	-
		Accuracy (CI)	.030 (.000, 064)	-	-
	Claude 3.7 Sonnet	Result	97 / 100	-	-
		Accuracy (CI)	.970 (.937, 1.000)	-	-
	Gemini 2.5 Flash	Result	22 / 100	-	-
		Accuracy (CI)	.220 (.139, .301)	-	-
The number of groups	GPT-3.5-turbo	Result	-	98 / 100	-
		Accuracy (CI)	-	.980 (.952, 1.000)	-
	GPT-4	Result	-	100 / 100	-
		Accuracy (CI)	-	1.000	-
	GPT-4o	Result	-	100 / 100	-
		Accuracy (CI)	-	1.000	-
	Llama 3.2-3b	Result	-	74 / 100	-
		Accuracy (CI)	-	.740 (.652, .828)	-
	Claude 3.7 Sonnet	Result	-	100 / 100	-
		Accuracy (CI)	-	1.000	-
	Gemini 2.5 Flash	Result	-	100 / 100	-

Continued on

Table 6 (continued)

Test type	Model		Two sample t-test	One-way ANOVA	χ^2 test
		Accuracy (CI)	-	1.000	-
Degree of freedom	GPT-3.5-turbo	Result	-	-	100 / 100
		Accuracy (CI)	-	-	1.000
	GPT-4	Result	-	-	47 / 100
		Accuracy (CI)	-	-	.470 (.370, .570)
	GPT-4o	Result	-	-	100 / 100
		Accuracy (CI)	-	-	1.000
	Llama 3.2-3b	Result	-	-	94 / 100
		Accuracy (CI)	-	-	.940 (.893, .987)
	Claude 3.7 Sonnet	Result	-	-	100 / 100
		Accuracy (CI)	-	-	1.000
	Gemini 2.5 Flash	Result	-	-	62 / 100
		Accuracy (CI)	-	-	.620 (.525, .715)

Note. CI = Confidence Interval. R = R code condition; Python = Python code condition; Direct = direct calculation condition.

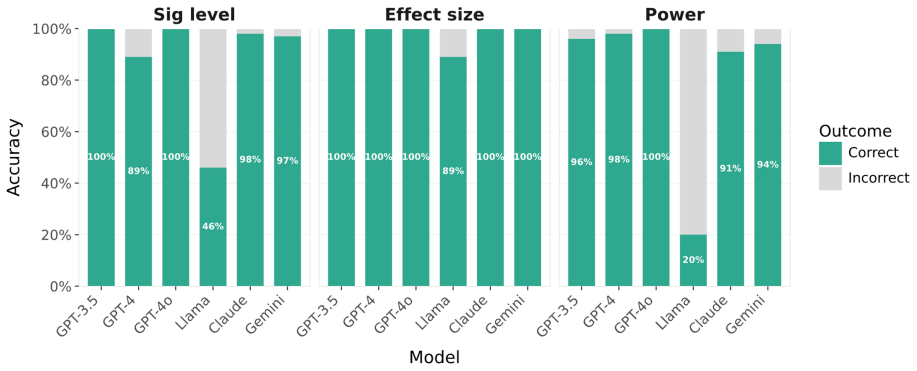


Figure 6. Results of Experiment 2

4 Experiment 3: Sample Size Calculation for Pearson Correlation and Multiple Regression

4.1 Method

Experimental Design and Procedure Experiment 3 extended Experiment 1 to examine whether the performance patterns observed for the two sample t-test, ANOVA, and χ^2 test generalize to two statistically more complex tests common in behavioral and social research: Pearson correlation and multiple linear regression. We used a $4 \times 3 \times 2$ factorial experimental design, incorporating three experimental factors: "models," "methods," and "test types."

First, the "model" factor included four conditions: Claude 4.6 Sonnet (Anthropic), GPT-4o (OpenAI), Gemini 2.5 Flash (Google), and Llama 3.2-3b (Meta). The specific models selected represent the current state of frontier LLMs, extending Experiment 1 with newer model versions across multiple providers.

Second, the "methods" factor had three conditions: (1) LLMs directly computing the required sample size, (2) LLMs generating R code to compute the required sample size, and (3) LLMs generating Python code to compute the required sample size. These conditions are identical to those used in Experiment 1.

Lastly, we considered two types of statistical tests in the "test types" factor: Pearson correlation and multiple linear regression. These tests were selected because they involve more complex computations than the two sample t-test, ANOVA, and χ^2 tests in Experiment 1: Pearson correlation uses the Fisher z transformation, while multiple regression requires solving the non-central F distribution numerically. For each test type, we randomly generated 100 parameter sets per condition, yielding $4 \times 3 \times 2 \times 100 = 2{,}400$ total trials.

Experimental Setup We used the same experimental setup as in Experiment 1, with the following updates. We examined a range of parameter combinations across three types of analyses. For Pearson correlation, the correlation coefficient (r) was varied across 0.10, 0.20, 0.30, 0.50, and 0.70, with significance levels (α) of 0.01 and 0.05 and target power levels of 0.75, 0.80, and 0.90. Ground-truth sample sizes were computed using the Fisher z closed-form formula (Cohen, 1988), which is mathematically equivalent to the procedure implemented in the R function pwr::pwr.r.test(). For multiple linear regression, the coefficient of determination (R^2) includes 0.02, 0.05, 0.10, 0.15, 0.25, and 0.35, and the number of predictors was set to 1, 2, 3, or 5, again with α values of 0.01 and 0.05 and target power levels of 0.75, 0.80, and 0.90. Ground-truth sample sizes were computed in R by numerically solving the non-central F distribution, following the same procedure implemented in pwr::pwr.f2.test() (Cohen, 1988). The two-turn conversation structure and three prompt conditions from Experiment 1 were retained.

Analyses We used the same analytical approach as in Experiment 1. To evaluate performance, we defined accuracy as the number of times the required sample size was correctly calculated, divided by the total number of trials.

4.2 Results

To answer Research Question 3, by instructing Claude 4.6 Sonnet and Gemini 2.5 Flash to generate R code, they both demonstrated perfect accuracy (1.00) for both Pearson correlation and multiple regression. However, overall performance across models was somewhat less consistent than in the simpler statistical scenarios examined earlier, such as the two sample t-test, one-way ANOVA, and χ^2 test. These findings suggest that although contemporary LLMs can also generate code for more advanced analyses, performance may become less stable as statistical procedures and parameter specifications increase in complexity. The results of Experiment 3 are presented in Table 7 and Figure 7.

To address Research Question 3, we examined whether the performance patterns observed in Experiment 1 would extend to Pearson correlation and multiple linear regression. Consistent with Experiment 1, R-code generation yielded the highest overall accuracy (.62), substantially outperforming both Python-code generation (.24) and direct calculation (.21). Overall, Claude 4.6 Sonnet and Gemini 2.5 Flash showed the strongest performance in Experiment 3, whereas Llama 3.1 8B performed poorly across conditions. These findings suggest that the advantage of code generation observed in Experiment 1 extends to Pearson correlation and multiple regression, although performance varied more across models and statistical procedures.

Table 7. Results of Experiment 3

			Pearson Correlation	Multiple Regression	Total
R	Claude 3.7 Sonnet	Result	100 / 100	100 / 100	200 / 200
		Accuracy (CI)	1.000	1.000	1.000
	GPT-4o	Result	100 / 100	0 / 100	100 / 200
		Accuracy (CI)	1.000	.000	.500 (.431, .569)
	Gemini 2.5 Flash	Result	100 / 100	100 / 100	200 / 200
		Accuracy (CI)	1.000	1.000	1.000
	Llama 3.2-3b	Result	0 / 100	0 / 100	0 / 200
		Accuracy (CI)	.000	.000	.000
Python	Claude 3.7 Sonnet	Result	82 / 100	15 / 100	97 / 200
		Accuracy (CI)	.820 (.733, .883)	.150 (.093, .233)	.485 (.417, .554)
	GPT-4o	Result	2 / 100	0 / 100	2 / 200
		Accuracy (CI)	.020 (.006, .070)	.000	.010 (.003, .036)
	Gemini 2.5 Flash	Result	90 / 100	0 / 100	90 / 200
		Accuracy (CI)	.900 (.826, .945)	.000	.450 (.383, .519)
	Llama 3.2-3b	Result	0 / 100	0 / 100	0 / 200
		Accuracy (CI)	.000	.000	.000
LLMs	Claude 3.7 Sonnet	Result	40 / 100	11 / 100	51 / 200
		Accuracy (CI)	.400 (.309, .498)	.110 (.063, .186)	.255 (.200, .320)
	GPT-4o	Result	35 / 100	18 / 100	53 / 200
		Accuracy (CI)	.350 (.264, .447)	.180 (.117, .267)	.265 (.209, .330)
	Gemini 2.5 Flash	Result	58 / 100	1 / 100	59 / 200
		Accuracy (CI)	.580 (.482, .672)	.010 (.002, .054)	.295 (.236, .362)
	Llama 3.2-3b	Result	0 / 100	7 / 100	7 / 200
		Accuracy (CI)	.000	.070 (.034, .137)	.035 (.017, .070)

Note. CI = Confidence Interval. R = R code condition; Python = Python code condition; Direct = direct calculation condition.

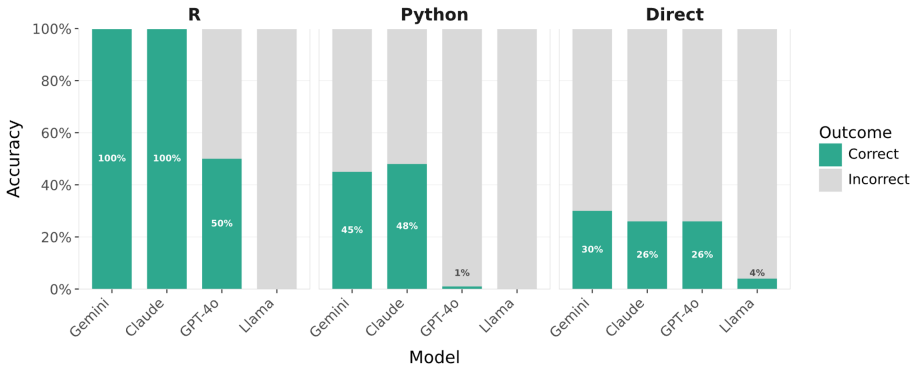


Figure 7. Results of Experiment 3

Performance differed substantially between the two tests. Claude 4.6 Sonnet and Gemini 2.5 Flash performed well on Pearson correlation across code-generation conditions, but accuracy declined markedly for multiple regression outside the R environment. In particular, Python-based multiple regression proved challenging because the analysis requires numerical optimization rather than a straightforward library call. As a result, models that performed well on the statistical tests examined in Experiment 1 did not always maintain the same level of performance for these more advanced analyses.

When errors occurred, they largely reflected the similar failure patterns identified in Experiment 1, including function hallucinations, incorrect handling of function inputs or outputs, and wrong formula. Notably, wrong-formula errors were more evident in Experiment 3, as models occasionally applied inappropriate statistical procedures while still producing plausible numerical results, making them difficult to detect at first glance. As a result, users should verify not only the final sample size estimate but also the statistical procedure used to obtain it. Representative examples are provided in **Appendix C**.

5 Discussion

While LLMs are increasingly integrated into research workflows, their ability to support statistically sophisticated procedures such as power analysis remains underexplored, raising concerns about unexamined adoption. To address this gap, we systematically evaluated the performance of six LLMs, including ChatGPT (GPT-3.5-turbo, GPT-4, and GPT-4o), Llama 3.2-3b, Claude 4.6 Sonnet, and Gemini 2.5 Flash, across three experiments. Experiment 1 assessed their ability to calculate sample sizes for common statistical tests under different prompting strategies, and Experiment 2 examined whether they could identify missing information necessary for valid power analysis. Experiment 3 extended Experiment 1 to assess whether these performance patterns generalize to Pearson correlation and multiple linear regression.

5.1 Sample Size Calculation Tasks

Performance across Different Models Across Experiments 1 and 3, substantial differences emerged across LLMs in their ability to perform power analysis tasks. Under optimal conditions, GPT-4, GPT-4o, and Claude 4.6 Sonnet achieved near-perfect or perfect performance, whereas GPT-3.5-turbo and Gemini 2.5 Flash generally showed intermediate performance and Llama consistently underperformed. These findings suggest that the ability to support statistical tasks varies considerably across models and should not be assumed to be a general characteristic of all LLMs. This finding extends previous work by Methnani et al. (2023), who observed failures in direct sample size calculations by ChatGPT, by employing common designs, conducting 100 replications per condition, and systematically favoring a code retrieval strategy over direct calculations.

At the same time, model performance was not entirely stable across statistical procedures. Several performance patterns observed in Experiment 1 were replicated in Experiment 3, but the relative ranking of models changed across tests. Models that performed strongly on common procedures such as t-tests, ANOVA, and χ^2 tests did not always maintain the same advantage when applied to Pearson correlation and multiple regression. This finding suggests that successful performance in statistical tasks depends not only on overall model capability but also on the specific computational and methodological requirements of a given procedure.

These results have important implications for evaluating LLMs in research settings. Performance on a limited set of benchmark tasks may provide an incomplete picture of a model's statistical capabilities. A model that performs accurately on familiar and well-defined analyses may still struggle when faced with procedures that require different computational approaches or less common statistical implementations. More broadly, the findings suggest that LLM capability for power analysis is better understood as domain-specific rather than global. Researchers should therefore avoid assuming that strong performance on one type of statistical task will necessarily generalize to others.

Effectiveness of Prompting Strategies Beyond differences between models, the way tasks were presented to LLMs had a substantial impact on performance. Across Experiments 1 and 3, prompting models to generate code consistently produced more accurate results than asking them to perform calculations directly. This finding aligns with prior research suggesting that contemporary LLMs are generally more effective at generating executable code than performing precise mathematical computations internally (e.g., Biswas, 2023; Liu et al., 2024; Nigar & Mohammed, 2023). Rather than directly performing statistical calculations, LLMs appear to be more effective at translating statistical problems into executable code that can be evaluated by software.

Within the code-generation conditions, R consistently outperformed Python. One possible explanation is that statistical power analysis is more commonly implemented and documented in R-based research workflows, providing models with greater exposure to relevant examples during training. Another possibility

is that some Python implementations require additional programming steps or software-specific knowledge, creating more opportunities for errors. For example, several failures stemmed from incorrect handling of package-specific arguments or outputs rather than from misunderstandings of the underlying statistical concepts.

Importantly, code generation should not be equated with correctness. Although many generated solutions were accurate, models occasionally produced function hallucinations and wrong-output-field errors. Function hallucinations occurred when models invoked functions that did not exist in the relevant package or selected inappropriate functions for the requested analysis. Wrong-output-field errors occurred when models correctly identified a statistical function but attempted to extract information that was not returned by the function. These failures highlight that code can appear plausible while still being nonfunctional.

In addition, Experiment 3 revealed examples of wrong-formula errors, in which models applied an inappropriate statistical procedure despite generating executable code. Unlike function hallucinations or wrong-output-field errors, wrong-formula errors often produced plausible numerical outputs without triggering runtime failures, making them particularly difficult for users to detect. Consequently, evaluating only whether code executes successfully may provide a false sense of correctness. Researchers should therefore verify not only the functionality of generated code but also the appropriateness of the underlying statistical procedure.

These findings suggest that performance in power analysis depends not only on the capabilities of a particular model but also on how users structure their interactions with it. Careful prompting can substantially improve performance, yet even high-performing models remain susceptible to subtle coding and methodological errors. As a result, LLM-generated code should be treated as a draft that requires independent verification rather than as a final analytical product.

5.2 Identification Tasks for Missing Information

Experiment 2 explored whether LLMs could identify missing input parameters required for conducting valid power analysis. Overall, performance was strongest when identifying commonly required parameters such as significance levels, effect sizes, and desired power values. This finding suggests that LLMs possess a degree of procedural knowledge about the structure of power analysis and can often recognize when essential inputs are absent.

However, performance was less consistent for test-specific parameters. Models frequently identified broadly applicable inputs while overlooking details that depend on the particular statistical design, such as tail specification or allocation ratios. This pattern suggests that LLMs may not always distinguish between information that is explicitly required and information that can be reasonably assumed. Instead, they often appear to rely on conventions commonly encountered in training data, such as assuming a two-tailed test or a default power level of .80, rather than prompting users to provide the missing information.

While these assumptions may sometimes be reasonable, they are not necessarily appropriate for a given research context.

This finding highlights a broader limitation of LLM-assisted statistical reasoning. Identifying missing information is not simply a matter of recalling which parameters are associated with a statistical procedure; it also requires recognizing when those parameters remain underspecified in a particular research design. In this sense, the challenge is not simply identifying which parameters are relevant to a statistical test, but determining whether those parameters have been adequately specified in a particular analysis. The tendency of LLMs to proceed with implicit assumptions suggests that they may prioritize producing a complete response over explicitly acknowledging uncertainty or incompleteness. As a result, users may receive outputs that appear methodologically sound while resting on assumptions that were never verified.

From a practical perspective, these findings suggest that LLMs may be more useful as aids for structuring power analyses than as autonomous methodological advisors. Rather than assuming that a model will automatically identify all missing inputs, researchers should actively prompt the model to verify parameter completeness and explicitly state any assumptions used in the analysis. Such verification is particularly important for test-specific design choices, where omissions may have meaningful consequences for sample size estimation. To facilitate this process, Figure 8 provides a recommended verification workflow corresponding to the statistical tests examined in this study.

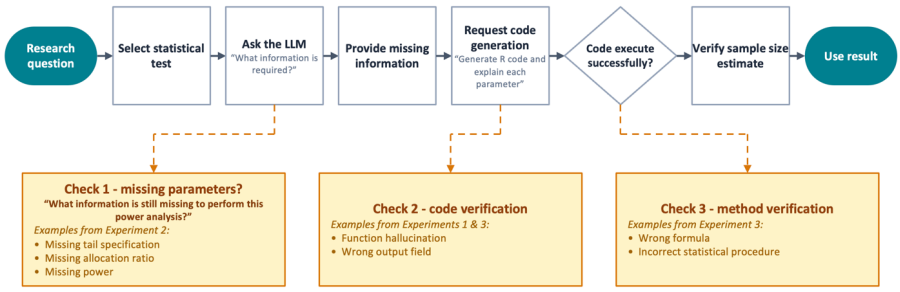


Figure 8. Recommended Workflow for LLM-assisted Power Analysis

5.3 Limitations and Future Direction

This study has several limitations, each of which provides insights into areas where future research can build upon. First, the evaluation focused primarily on relatively structured power-analysis scenarios involving two-sample t-tests, one-way ANOVA, χ^2 goodness-of-fit tests, Pearson correlation, and multiple linear regression. These procedures were intentionally selected because they have

well-established analytical solutions, allowing objective verification of correctness and providing a necessary baseline for evaluating LLM performance. However, many real-world power analyses involve substantially greater complexity, including multilevel models, structural equation models, repeated-measures designs, and simulation-based approaches. Although Experiment 3 extended the evaluation to Pearson correlation and multiple regression, these analyses still rely on established statistical procedures and software implementations.

An important question raised by the present findings is whether LLM limitations stem primarily from statistical complexity itself or from difficulties implementing statistical procedures in software. In several cases, models appeared to understand the underlying statistical task but failed because of software-specific issues, such as incorrect function calls or inappropriate handling of function arguments and outputs. This distinction may be particularly important for simulation-based power analyses, which often require researchers to develop custom computational workflows rather than rely on established power-analysis functions. Evaluating LLM performance in these settings could help determine whether current models are primarily retrieving familiar coding patterns or demonstrating broader statistical reasoning capabilities. Future research should therefore examine LLM performance on more complex power-analysis scenarios, particularly simulation-based approaches that are increasingly common in applied research (e.g., Arend & Schäfer, 2019; Arnold, Hogan, Colford, & Hubbard, 2011; Moshagen & Bader, 2023; Qin, 2023; Taylor & Bosch, 1990; Z. Zhang, 2014).

Second, the study evaluated specific model snapshots available at the time of data collection, including GPT-3.5-turbo (gpt-3.5-turbo-0125), GPT-4 (gpt-4-0613), GPT-4o (gpt-4o-2024-08-06), Llama 3.2-3B (Llama 3.2-3B-Instruct), Claude 4.6 Sonnet (claude-3-7-sonnet-20250219), and Gemini 2.5 Flash (gemini-2.5-flash-preview-04-17). Because LLMs are updated frequently, the findings should be interpreted as evaluations of these specific model versions rather than stable estimates of future performance. More broadly, rapid model evolution presents a methodological challenge for LLM benchmarking research, as performance estimates may become outdated more quickly than traditional software evaluations. Future studies would benefit from maintaining reproducible benchmark tasks and reporting exact model identifiers to facilitate longitudinal comparisons across model generations.

Third, this study evaluated LLMs primarily through direct, one-shot interactions. Although this approach allowed controlled comparisons across conditions, it does not fully capture how researchers may use LLMs in practice. Real-world use often involves iterative clarification, error checking, and refinement of prompts. The present findings suggest that performance is influenced not only by the capabilities of a model but also by how users frame tasks and respond to model outputs. Statistical performance should be viewed as a product of both model capability and user interaction rather than as a fixed property of the model alone. Future research should investigate not only whether LLMs make

statistical errors, but also whether users can successfully identify and correct those errors during real-world analytical tasks.

Fourth, the present study focused on the accuracy of LLM outputs rather than their practical value in real-world research workflows. This focus was intentional, as systematic evaluation of accuracy was necessary to establish a baseline assessment of LLM performance in power analysis. However, accuracy alone does not capture whether LLMs provide meaningful benefits to researchers in practice. Established tools such as G*Power and dedicated R packages already offer highly reliable sample size calculations once the appropriate parameters have been specified. Consequently, the value of LLMs may lie less in performing calculations themselves and more in assisting users with tasks such as identifying required inputs, clarifying methodological assumptions, and interpreting statistical procedures. Future research should therefore compare LLM-assisted and traditional workflows in terms of usability, efficiency, user understanding, and error rates to determine the practical contexts in which LLMs provide added value beyond existing statistical software.

6 Conclusion

Given the increasing integration of LLM-based approaches into research workflows, it is important to examine how reliably these models perform in statistically demanding contexts. Therefore, this study aims to systematically evaluate the performance of the selected LLMs in the context of power analysis rather than assume their usefulness, with the broader goal of promoting more rigorous and responsible practices of AI in research.

Overall, the findings indicate that the evaluated model versions demonstrate promising yet uneven capability in supporting sample size calculations for power analysis. On the one hand, GPT-4, GPT-4o, and Claude 4.6 Sonnet frequently retrieved appropriate statistical formulas, offered structured guidance, and generated executable code for sample size estimation tasks. Experiment 3 further demonstrated that this advantage for R-code generation extends to Pearson correlation and multiple linear regression, although performance became more variable across models and statistical procedures. These outcomes suggest that the tested models can appear competent and may assist users in navigating certain aspects of power analysis under controlled conditions.

However, consistent with prior studies (e.g., Frieder et al., 2023), the results also confirmed notable limitations in the evaluated LLMs' direct mathematical reasoning abilities. When tasked with performing numerical calculations, these model versions often produced incorrect values due to rounding errors, inappropriate substitutions, or incomplete execution of formulas. These issues likely stem from the models' token-based probabilistic pattern matching rather than genuine mathematical understanding (Jiang et al., 2024; Mirzadeh et al., 2024). Moreover, the accuracy of the tested models varies across different statistical tests, and in some cases, the model misinterprets missing data or default to as-

sumptions that require careful verification, which may affect the validity of the results if left unchecked.

Given that inaccuracies in sample size calculation may affect study design, these findings highlight the importance of careful validation when using LLM-generated outputs. LLMs may serve as useful aids in structuring or supporting aspects of the analytical process, but they should not be treated as substitutes for domain expertise. Outputs should be verified using established methods, and interpretation should involve appropriate methodological oversight (van Dis, Bollen, Zuidema, van Rooij, & Bockting, 2023; Zhu, Jiang, Yang, & Ren, 2023).

Beyond computational considerations, broader issues of research accountability and ethical use must also be addressed. As LLMs become increasingly embedded in research workflows (e.g., Aubin Le Quéré et al., 2024; Khraisha et al., 2024; Methnani et al., 2023; Watkins, 2024), issues of research accountability and ethical use grow more critical. Although models such as GPT-4 and GPT-4o demonstrated strong performance in this study, none of the evaluated model versions consistently produced flawless results, underscoring the ongoing necessity for human oversight. Furthermore, concerns regarding data security and confidentiality must be taken seriously; researchers should refrain from inputting sensitive or identifiable information into LLMs, particularly when working with contextual or study-specific data (Charfeddine, Kammoun, Hamdaoui, & Guizani, 2024). Ultimately, while LLMs might offer valuable potential as supplemental tools for statistical analysis, their responsible integration requires critical oversight, human expertise, rigorous cross-validation, and a continued commitment to preserving the standards of scientific rigor. Further research is needed to better understand their capabilities and limitations across a wider range of analytical contexts.

Author Notes

Correspondence concerning this article should be addressed to Feng Ji, Department of Applied Psychology and Human Development, University of Toronto, 252 Bloor West Street, Toronto, Ontario, Canada. Email: f.ji@utoronto.ca

Funding

This study was funded by the Connaught Fund (#520245) and Social Sciences and Humanities Research Council (#519992), International Network of Educational Institutes Seed Funding (# 522011) and CRC (#001169).

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Availability of data and materials

All data are available at https://osf.io/m9ztc.

Code availability

All code is available at https://osf.io/m9ztc.

Open Practices Statement

The code, experiments, and results used to support the findings of this study are available at https://osf.io/m9ztc under the Creative Commons Attribution NonCommercial-ShareAlike 4.0 International license (CC BY-NC-SA 4.0). None of the experiments were preregistered.

References

Aberson, C. L. (2019). *Applied power analysis for the behavioral sciences*. Routledge.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*. Retrieved from https://doi.org/10.48550/arXiv.2303.08774

Anderson, S. F., Kelley, K., & Maxwell, S. E. (2017). Sample-size planning for more accurate statistical power: A method adjusting sample effect sizes for publication bias and uncertainty. *Psychological Science*, *28*(11), 1547-1562. Retrieved from https://doi.org/10.1177/0956797617723724

Arend, M. G., & Schäfer, T. (2019). Statistical power in two-level models: A tutorial based on monte carlo simulation. *Psychological Methods*, *24*(1), 1–19. Retrieved from https://doi.org/10.1037/met0000195

Arnold, B. F., Hogan, D. R., Colford, J., & Hubbard, A. E. (2011). Simulation methods to estimate design power: An overview for applied research. *BMC Medical Research Methodology*, *11*(1). Retrieved from https://doi.org/10.1186/1471-2288-11-94

Aubin Le Quéré, M., Schroeder, H., Randazzo, C., Gao, J., Epstein, Z., Perrault, S. T., et al. (2024). Llms as research tools: Applications and evaluations in hci data work. In *Extended abstracts of the chi conference on human factors in computing systems* (p. 1-7). Retrieved from https://doi.org/10.1145/3613905.3636301

Bakker, M., Hartgerink, C. H. J., Wicherts, J. M., & van der Maas, H. L. J. (2016). Researchers' intuitions about power in psychological research. *Psychological Science*, *27*(8), 1069-1077. Retrieved from https://psycnet.apa.org/doi/10.1177/0956797616647519

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, *57*(1), 289–300. Retrieved from https://doi.org/10.1111/j.2517-6161.1995.tb02031.x

Birhane, A., Kasirzadeh, A., Leslie, D., & Wachter, S. (2023). Science in the age of large language models. *Nature Reviews Physics*, *5*(5), 277-280. Retrieved from https://doi.org/10.1038/s42254-023-00581-4

Biswas, S. (2023). Role of chatgpt in computer programming. *Mesopotamian Journal of Computer Science*, *2023*, 8-16. Retrieved from https://doi.org/10.58496/MJCSC/2023/002

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... Amodei, D. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*. Retrieved from https://doi.org/10.48550/arxiv.2005.14165

Button, K. S., Ioannidis, J. P., Mokrysz, C., Nosek, B. A., Flint, J., Robinson, E. S., & Munafò, M. R. (2013). Power failure: why small sample size undermines the reliability of neuroscience. *Nature Reviews. Neuroscience*, *14*(5), 365–376. Retrieved from https://doi.org/10.1038/nrn3475

Champely, S., Ekstrom, C., Dalgaard, P., Gill, J., Weibelzahl, S., Anandkumar, A., ... De Rosario, H. (2017). *pwr: Basic functions for power analysis.* https://cran.r-project.org/web/packages/pwr/. (Software)

Charfeddine, M., Kammoun, H. M., Hamdaoui, B., & Guizani, M. (2024). Chatgpt's security risks and benefits: Offensive and defensive use-cases, mitigation measures, and future implications. *IEEE Access*, *12*. Retrieved from https://doi.org/10.1109/ACCESS.2024.3367792

Chen, T. C., Kaminski, E., Koduri, L., Singer, A., Singer, J., Couldwell, M., ... Wang, A. (2023). Chatgpt as a neuro-score calculator: Analysis of a large language model's performance on various neurological exam grading scales. *World Neurosurgery*, *179*, 342–347. Retrieved from https://doi.org/10.1016/j.wneu.2023.08.088

Chen, T. J. (2023). Chatgpt and other artificial intelligence applications speed up scientific writing. *Journal of the Chinese Medical Association*, *86*(4), 351-353. Retrieved from https://doi.org/10.1097/JCMA.0000000000000900

Cheong, I., Xia, K., Feng, K. K., Chen, Q. Z., & Zhang, A. X. (2024). (a) i am not a lawyer, but...: Engaging legal experts towards responsible llm policies for legal advice. In *The 2024 acm conference on fairness, accountability, and transparency* (p. 2454-2469). Retrieved from https://doi.org/10.1145/3630106.3659048

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences.* (2nd ed). L. Erlbaum Associates.

Colavizza, G. (2025). *Large language models for social science research.* https://www.summerschoolsineurope.eu/course/large-language-models-for-social-science-research/. ([Workshop]. Università della Svizzera italiana, Lugano, Switzerland)

Corp., I. (2021). *Ibm spss statistics for mac, version 28.0.* (Armonk, NY: IBM Corp.)

Correll, J., Mellinger, C., McClelland, G. H., & Judd, C. M. (2020). Avoid cohen's 'small', 'medium', and 'large' for power analysis. *Trends in Cognitive*

Sciences, *24*(3), 200–207. Retrieved from https://doi.org/10.1016/j.tics.2019.12.009

Espejel, J. L., Ettifouri, E. H., Alassan, M. S. Y., Chouham, E. M., & Dahhane, W. (2023). Gpt-3.5, gpt-4, or bard? evaluating llms reasoning ability in zero-shot setting and performance boosting through prompts. *arXiv preprint arXiv:2305.12477*. Retrieved from https://doi.org/10.48550/arxiv.2305.12477

Evkaya, O., & de Carvalho, M. (2024). Decoding ai: The inside story of data analysis in chatgpt. *arXiv preprint arXiv:2024.08480.*. Retrieved from https://doi.org/10.48550/arxiv.2404.08480

Faul, F., Erdfelder, E., Lang, A.-G., & Buchner, A. (2007). Gpower 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior Research Methods*, *39*(2), 175–191. Retrieved from https://doi.org/10.3758/BF03193146

Finch, S., Cumming, G., & Thomason, N. (2001). Reporting of statistical inference in the journal of applied psychology: Little evidence of reform. *Educational and Psychological Measurement*, *61*(2), 181–210. Retrieved from https://doi.org/10.1177/0013164401612001

for Social Science, O. D. I., & Innovations, E. (2025). *Large language models in social science research.* https://odissei-data.nl/event/workshop-llm/. ([Workshop]. Utrecht University, Utrecht, Netherlands)

Frank, M. C. (2023). Baby steps in evaluating the capacities of large language models. *Nature Reviews Psychology*, *2*(8), 451-452. Retrieved from https://doi.org/10.1038/s44159-023-00211-x

Frieder, S., Pinchetti, L., Chevalier, A., Griffiths, R.-R., Salvatori, T., Lukasiewicz, T., ... Berner, J. (2023). Mathematical capabilities of chatgpt. *arXiv preprint arXiv:2301.13867*. Retrieved from https://doi.org/10.48550/arXiv.2301.13867

Fritz, A., Scherndl, T., & Kühberger, A. (2013). A comprehensive review of reporting practices in psychological journals: Are effect sizes really enough? *Theory & Psychology*, *23*(1), 98–122. Retrieved from https://doi.org/10.1177/0959354312436870

Hermann, C. E., Patel, J. M., Boyd, L., Growdon, W. B., Aviki, E., & Stasenko, M. (2023). Let's chat about cervical cancer: Assessing the accuracy of chatgpt responses to cervical cancer questions. *Gynecologic Oncology*, *179*, 164–168. Retrieved from https://doi.org/10.1016/j.ygyno.2023.11.008

Hu, X., Zhao, Z., Wei, S., Chai, Z., Ma, Q., Wang, G., et al. (2024). Infiagent-dabench: Evaluating agents on data analysis tasks. *arXiv preprint arXiv:2401.05507*. Retrieved from https://doi.org/10.48550/arXiv.2401.05507

Inc., S. I. (2023). *Sas/stat® 15.3 user's guide.* (Cary, NC: SAS Institute Inc.)

in Europe, S. S. (2025). *Large language models for social science research summer course.* https://www.summerschoolsineurope.eu/course/large-language-models-for-social-science-research/. (Retrieved April 28, 2025)

Ioannidis, J. P. A. (2005). Why most published research findings are false. *Research Integrity in the Biomedical Sciences*, *2*(8), 0696–0701. Retrieved from https://doi.org/10.1371/journal.pmed.0020124

Jahangiri, Y. (2023). Can chat generative pretraining transformer (chatgpt) be used for statistical analysis of research data? *Journal of Vascular and Interventional Radiology*, *34*(12), 2242-2246. Retrieved from https://doi.org/10.1016/j.jvir.2023.09.010

Jalali, M. S., & Akhavan, A. (2024). Integrating ai language models in qualitative research: Replicating interview data analysis with chatgpt. *System Dynamics Review*, *40*(3). Retrieved from https://doi.org/10.1002/sdr.1772

Jiang, B., Xie, Y., Hao, Z., Wang, X., Mallick, T., Su, W. J., et al. (2024). A peek into token bias: Large language models are not yet genuine reasoners. *arXiv preprint arXiv:2406.11050*. Retrieved from https://doi.org/10.48550/arXiv.2406.11050

Jiao, W., Wang, W., Huang, J. T., Wang, X., & Tu, Z. (2023). Is chatgpt a good translator? a preliminary study. *arXiv preprint arXiv:2301.08745.*. Retrieved from https://doi.org/10.48550/arXiv.2301.08745

Kashefi, A., & Mukerji, T. (2023). Chatgpt for programming numerical methods. *arXiv preprint arXiv:2303.12093*. Retrieved from https://doi.org/10.48550/arxiv.2303.12093

Khraisha, Q., Put, S., Kappenberg, J., Warraitch, A., & Hadfield, K. (2024). Can large language models replace humans in systematic reviews? evaluating gpt-4's efficacy in screening and extracting data from peer-reviewed and grey literature in multiple languages. *Research Synthesis Methods*, *15*(4). Retrieved from https://doi.org/10.1002/jrsm.1715

Kingsley, B. E., & Robertson, J. M. (2017). Exploring reticence in research methods: The experience of studying psychological research methods in higher education. *Psychology Teaching Review*, *23*(2), 4–19. Retrieved from https://doi.org/10.53841/bpsptr.2017.23.2.4

Kitamura, F. C. (2023). Chatgpt is shaping the future of medical writing but still requires human judgment. *Radiology*, *307*(2). Retrieved from https://doi.org/10.1148/radiol.230171

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2023). Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916.*. Retrieved from https://doi.org/10.48550/arxiv.2205.11916

Lecler, A., Duron, L., & Soyer, P. (2023). Revolutionizing radiology with gpt-based models: Current applications, future possibilities and limitations of chatgpt. *Diagnostic and Interventional Imaging*, *104*(6), 269–274. Retrieved from https://doi.org/10.1016/j.diii.2023.02.003

Le Mens, G. (2024). *Using large language models for empirical research in social science.* https://www.ibei.org/en/using-large-language-models-for-empirical-research-in-social-science_350753. ([Workshop]. Institut Barcelona Estudis Internacionals, Barcelona, Spain)

Liu, J., Xia, C. S., Wang, Y., & Zhang, L. (2024). Is your code generated by chatgpt really correct? rigorous evaluation of large language models for

code generation. *Advances in Neural Information Processing Systems*, *36*. Retrieved from https://doi.org/10.48550/arXiv.2305.01210

Lixandru, I.-D. (2024). The use of artificial intelligence for qualitative data analysis: Chatgpt. *Informatica Economica*, *28*(1), 57–67. Retrieved from https://doi.org/10.24818/issnl4531305/28.1.2024.05

Maxwell, S. E. (2004). The persistence of underpowered studies in psychological research: Causes, consequences, and remedies. *Psychological Methods*, *9*(2), 147–163. Retrieved from https://doi.org/10.1037/1082-989X.9.2.147

Maxwell, S. E., Lau, M. Y., & Howard, G. S. (2015). Is psychology suffering from a replication crisis?: What does "failure to replicate" really mean? *The American Psychologist*, *70*(6), 487–498. Retrieved from https://doi.org/10.1037/a0039400

Methnani, J., Latiri, I., Dergaa, I., Chamari, K., & Saad, H. B. (2023). Chatgpt for sample-size calculation in sports medicine and exercise sciences: A cautionary note. *International Journal of Sports Physiology and Performance*, *18*(10), 1219–1223. Retrieved from https://doi.org/10.1123/ijspp.2023-0109

Mirzadeh, I., Alizadeh, K., Shahrokhi, H., Tuzel, O., Bengio, S., & Farajtabar, M. (2024). Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*. Retrieved from https://doi.org/10.48550/arXiv.2410.05229

Moshagen, M., & Bader, M. (2023). sempower: General power analysis for structural equation models. *Behavior Research Methods*, *56*(4), 2901–2922. Retrieved from https://doi.org/10.3758/s13428-023-02254-7

Myors, B., & Murphy, K. R. (2023). *Statistical power analysis: A simple and general model for traditional and modern hypothesis tests.* . (Fifth edition.). Routledge. Retrieved from https://doi.org/10.4324/9781003296225

Nigar, M. S., & Mohammed, Y. S. (2023). Use chatgpt to solve programming bugs. *International Journal of Information Technology & Computer Engineering*, *3*(01), 17–22. Retrieved from https://doi.org/10.55529/ijitc.31.17.22

OpenAI. (2022). *Chatgpt [large language model]*. https://chat.openai.com/chat. (Published November 30, 2022)

Paul, J., Ueno, A., & Dennis, C. (2023). Chatgpt and consumers: Benefits, pitfalls and future research agenda. *International Journal of Consumer Studies*, *47*(4), 1213–1225. Retrieved from https://doi.org/10.1111/ijcs.12928

Piccolo, S. R., Denny, P., Luxton-Reilly, A., Payne, S., & Ridge, P. G. (2023). Many bioinformatics programming tasks can be automated with chatgpt. *arXiv preprint arXiv:2303.13528*.. Retrieved from https://doi.org/10.48550/arxiv.2303.13528

Qin, X. (2023). Sample size and power calculations for causal mediation analysis: A tutorial and shiny app. *Behavior Research Methods*, *56*(3), 1738–1769. Retrieved from https://doi.org/10.3758/s13428-023-02118-0

Rasheed, Z., Waseem, M., Ahmad, A., Kemell, K. K., Xiaofeng, W., Duc, A. N., & Abrahamsson, P. (2024). Can large language models serve as data analysts? a multi-agent assisted approach for qualitative data analysis. *arXiv preprint arXiv:2402.01386*. Retrieved from https://doi.org/10.48550/arXiv.2402.01386

Ray, P. P. (2023). Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, *3*, 121–154. Retrieved from https://doi.org/10.1016/j.iotcps.2023.04.003

Rosenfeld, A., & Lazebnik, T. (2024). Whose llm is it anyway? linguistic comparison and llm attribution for gpt-3.5, gpt-4 and bard. *arXiv preprint arXiv:2402.14533.*. Retrieved from https://doi.org/10.48550/arxiv.2402.14533

Sedlmeier, P., & Gigerenzer, G. (1989). Do studies of statistical power have an effect on the power of studies? *Psychological Bulletin*, *105*(2), 309–316. Retrieved from https://doi.org/10.1037/0033-2909.105.2.309

StataCorp. (2023). *Stata 18 stata power, precision, and sample-size reference manual.* (College Station, TX: Stata Press)

Taylor, D. W., & Bosch, E. G. (1990). Cts: A clinical trials simulator. *Statistics in Medicine*, *9*(7), 787–801. Retrieved from https://doi.org/10.1002/sim.4780090708

Vallat. (2018). Pingouin: Statistics in python. *Journal of Open Source Software*, *3*(31), 1026. Retrieved from https://doi.org/10.21105/joss.01026

van Dis, E. A. M., Bollen, J., Zuidema, W., van Rooij, R., & Bockting, C. L. (2023). Chatgpt: Five priorities for research. *Nature (London)*, *614*(7947), 224–226. Retrieved from https://doi.org/10.1038/d41586-023-00288-7

Wang, H., Fu, T., Du, Y., Gao, W., Huang, K., Liu, Z., et al. (2023). Scientific discovery in the age of artificial intelligence. *Nature*, *620*(7972), 47-60. Retrieved from https://doi.org/10.1038/s41586-023-06221-2

Watkins, R. (2024). Guidance for researchers and peer-reviewers on the ethical use of large language models (llms) in scientific research workflows. *AI and Ethics*, *4*(4), 969-974. Retrieved from https://doi.org/10.1007/s43681-023-00294-5

Wilkinson, L., & on Statistical Inference., T. F. (1999). Statistical methods in psychology journals: Guidelines and explanations. *American Psychologist*, *54*, 594-604. Retrieved from https://psycnet.apa.org/doi/10.1037/0003-066X.54.8.594

Xu, H., Sharaf, A., Chen, Y., Tan, W., Shen, L., Van Durme, B., et al. (2024). Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*. Retrieved from https://doi.org/10.48550/arXiv.2401.08417

Zhang, T., Ladhak, F., Durmus, E., Liang, P., McKeown, K., & Hashimoto, T. B. (2024). Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, *12*, 39-57.

Retrieved from https://doi.org/10.1162/tacl_a_00632

Zhang, Z. (2014). Monte carlo based statistical power analysis for mediation models: Methods and software. *Behavior Research Methods*, *46*(4), 1184–1198. Retrieved from https://doi.org/10.3758/s13428-013-0424-0

Zhao, H., Liu, Z., Wu, Z., Li, Y., Yang, T., Shu, P., et al. (2024). Revolutionizing finance with llms: An overview of applications and insights. *arXiv preprint arXiv:2401.11641*. Retrieved from https://doi.org/10.48550/arXiv.2401.11641

Zheng, Y., Koh, H. Y., Ju, J., Nguyen, A. T., May, L. T., Webb, G. I., & Pan, S. (2025). Large language models for scientific discovery in molecular property prediction. *Nature Machine Intelligence*, 1-11. Retrieved from https://doi.org/10.1038/s42256-025-00994-z

Zhu, J. J., Jiang, J., Yang, M., & Ren, Z. J. (2023). Chatgpt and environmental research. *Environmental Science & Technology*, *57*(46), 17667–17670. Retrieved from https://doi.org/10.1021/acs.est.3c01818

Appendix A. Representative Examples of Common Error in Experiment 1

Error Type	Model	Method	Examples	Issue

Function hallucination	Gemini 2.5 Flash	**R**	`library(pwr)`
			`alpha <- 0.05; effect_size <- 0.5; desired_power <- 0.8`
			`result <- power.t.test(n = NULL, delta = effect_size, sig.level = alpha, power = desired_power, type = "two.sample", alternative = "two.sided")`
			`print(ceiling(result$n))`
				Uses power.t.test() while presenting the solution as a pwr-package analysis. The intended function for this task is pwr.t.test().

Wrong output field	GPT-4o	**Python**	`from statsmodels.stats.power import GofChisquarePower`
			`effect_size = 0.1; alpha = 0.01; power = 0.8; df = 20`
			`analysis = GofChisquarePower()`
			`result = analysis.solve_power(effect_size=effect_size, nobs=None, alpha=alpha, power=power, df=df)`
			`print('Required sample size: %.3f' % result)`
				Correctly identifies the appropriate function (GofChisquarePower) but supplies an unsupported argument (*df*). The Python implementation requires the number of categories (*n_bins*) rather than degrees of freedom, causing the code to fail.

Wrong calculation	GPT-3.5-turbo	**Direct**	`Based on the input parameters, the required sample size for each group is approximately 1,036 participants.`	Produces an incorrect sample size estimate despite identifying the appropriate analysis procedure. Unlike coding errors, this type of error yields a plausible numerical answer with no obvious indication that the result is incorrect.

Appendix B. Representative Examples of Errors in Experiment 2

Example A: One-Way ANOVA - Claude 3.7 Sonnet (Power Missing)	
User	significance level = 0.05, effect size f = 0.15, groups = 6. I want to calculate the required sample size. Is there any missing information?
LLM	**No, you have all the necessary information!** Let me help you calculate the required sample size for your One-way ANOVA. Given information: • Significance level (α) = 0.05 • Effect size (f) = 0.15 (small to medium effect) • Number of groups (k) = 6 • Power level = 0.80 (**assuming the standard 80% power, since you didn't specify**) The required sample size is approximately 248 per group (1,488 total). Note: Desired power was not provided. The model assumed the standard value of 0.80 without flagging it as missing information.
Example B: Two-Sample t-test - Gemini 2.5 Flash (Tail Information Missing)	
User	significance level = 0.05, effect size = 0.3, power = 0.8, n1 = n2. I want to calculate the required sample size. Is there any missing information?
LLM	That's excellent! **You've provided all the necessary information** for a standard sample size calculation for an independent 2-sample t-test. To recap, you have: • Significance level (α) = 0.05 • Effect size (Cohen's d) = 0.3 • Desired power ($1-\beta$) = 0.8 • Equal sample sizes: n1 = n2 The required sample size is approximately 176 per group (352 total). Note: Tail direction (one-tailed vs. two-tailed) was not provided. **The model assumed a two-tailed test and proceeded without flagging the omission.**

Appendix C. Representative Examples of Error in Experiment 3

Error Type	Model	Method	Examples	Issue
Function hallucination	Llama 3.2-3b	R (Correlation)	`library(pwr) n <- pwr.r.cor.test(r = 0.1, power = 0.8, sig.level = 0.05, alternative = "two.sided")` `print(paste("Required␣sample␣size:", n))`	Gemini calling power.t.test() in a context where pwr.t.test() was required.

Wrong output field	GPT-4o	R (Multiple Regression)	`library(pwr)`
			`R2 <- 0.02`
			`f2 <- R2 / (1 - R2)`
			`u <- 1`
			`alpha <- 0.05`
			`power <- 0.8`
			`result <- pwr.f2.test(u = u, f2 = f2, sig.level = alpha, power = power)`
			`cat("Required␣sample␣size:", ceiling(result), "\n")`
				GPT-4o correctly calling the pwr.f2.test() function but attempting to use an unsupported argument/output (*result$n* instead of *result$v)*

Wrong formula	GPT-4o	Python (Correlation)	`import numpy as np`
			`from statsmodels.stats.power import NormalIndPower`
			`r = 0.1; alpha = 0.05; power = 0.8`
			`effect_size = np.arctanh(r)`
			`power_analysis = NormalIndPower()`
			`sample_size = power_analysis.solve_power(effect_size=effect_size, alpha=alpha, power=power, alternative='two-sided')`
			`print(f"Required␣sample␣size:␣{sample_size}")`
				NormalIndPower is a two-sample z-test formula (n per group), not appropriate for Pearson correlation, producing a plausible but incorrect sample size.