**The performance of widely used language models tasked with identifying the distribution responsible for generating simulated data**

Justus Eaglesmith1, Tim Johnson2, Robert W. Walker2

1. NARA Northwest, Portland, Oregon, USA 97201

2. Willamette University, Salem, Oregon, USA 97301

**Abstract.** Discussions of fully automated data analysis have grown more salient with advances in generative artificial intelligence (AI) systems, such as large language models (LLMs). Researchers have found that LLMs succeed at coding tasks, research design, and the discovery of sophisticated statistical models. Despite success in these advanced applications, LLM performance in basic analytic tasks has received little attention. We report a study that randomly generated data from a known distribution, presented those data to LLMs via stem-and-leaf plots and summary statistics, and tasked the LLMs with naming the distribution that generated the observed data. The study did not fine-tune LLMs, nor did it employ prompt engineering techniques that might influence performance. LLMs in the test exhibited highly variable performance and inconsistently detected the known true distribution responsible for data generation.

**Introduction**

The prospect of artificial intelligence (AI) systems automating data analysis no longer turns heads. The idea itself has a long history (Bie et al. 2022) and over a year has passed since researchers began musing that future data scientists will need to prepare for a career more akin to that of a product manager (i.e. orchestrating a production process) than that of a software engineer (i.e. coding and performing analysis) (Tu et al. 2023). These projections have resulted from the success of large language models (LLMs) performing tasks previously deemed within the domain of human intelligence (Bubeck et al. 2023). Research has shown that LLMs exhibit noteworthy capabilities in data wrangling (Jaimovitch-López et al. 2023), research design (Manning, Zhu, and Horton 2024), discovery of sophisticated statistical models (Li, Fox, and Goodman 2024), and the automation of machine learning itself (Tayebi Arasteh et al. 2024), not to mention in general coding tasks, albeit with debated performance (Nejjar et al. ; Wang and Chen 2023).

In this paper, we contribute to this literature via a more-modest test of the analytic capabilities of LLMs. We examine the performance of LLMs tasked with assessing the best distribution with which to model a random variable. This foundational task of parametric statistics has the advantage of offering an objective “ground truth” that we can use to assess model performance—that is, we randomly generate data from a known distribution and we present salient information about that data to an LLM that we also task with specifying the (known) distribution responsible for generating the data values.

Specifically, our tests involve generating data randomly before presenting them to LLMs via stem-and-leaf plots as well as summary statistics. We query LLMs without any prompt-engineering, fine-tuning, or retrieval-augmented generation, thus getting a sense of how LLMs perform “off-the-shelf” as a naïve user might deploy them to assess the distribution of a given variable’s values. In this context, we find that LLMs rarely output the name of the distribution that we used to generate the data, thus raising questions about the efficacy of current LLMs at performing this foundational data analysis task—namely, identifying a distribution well-suited to model a variable’s values. We interpret this finding, as discussed in our conclusion, not as evidence relating to the “intelligence” of LLMs versus humans, but as evidence pertaining to the usefulness of LLMs in a particular analytic task and the ways that features of an analytic task (e.g., modeling indeterminacy due to low sample size) can pose problems even to advanced AI systems.

**Table 1. Values of key distribution parameters in the study**

| Beta | Binomial | Chi-square | Gamma | Geometric | Lognormal | Normal | Poisson | Uniform |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1,2 | p=0.1; size=5, 10, 20, 40 | df=1 | 1,2 | p=0.1 | log-Mean=0; log-SD=1 | Mean=0; SD=1 | lambda=0.5 | min=0; max=1 |
| 2,2 | p=0.2; size=5, 10, 20, 40 | df=2 | 2,2 | p=0.2 | log-Mean=-5; log-SD=1 | Mean=-5; SD=1 | lambda=1 | min=0; max=2 |
| 3,2 | p=0.3; size=5, 10, 20, 40 | df=3 | 3,2 | p=0.3 | log-Mean=5; log-SD=1 | Mean=5; SD=1 | lambda=2 | min=0; max=3 |
| 4,2 | p=0.4; size=5, 10, 20, 40 | df=4 | 4,2 | p=0.4 | log-Mean=0; log-SD=5 | Mean=0; SD=5 | lambda=3 | min=0; max=4 |
| 5,2 | p=0.5; size=5, 10, 20, 40 | df=5 | 5,2 | p=0.5 | log-Mean=-5; log-SD=5 | Mean=-5; SD=5 | lambda=4 | min=0; max=5 |
|  | p=0.6; size=5, 10, 20, 40 | df=6 | 6,2 | p=0.6 | log-Mean=5; log-SD=5 | Mean=5; SD=5 | lambda=5 | min=0; max=6 |
|  | p=0.7; size=5, 10, 20, 40 | df=7 | 7,2 | p=0.7 |  |  | lambda=7.5 | min=0; max=7 |
|  | p=0.8; size=5, 10, 20, 40 | df=8 | 8,2 | p=0.8 |  |  | lambda=10 | min=0; max=8 |
|  | p=0.9; size=5, 10, 20, 40 | df=9 | 9,2 | p=0.9 |  |  |  | min=0; max=9 |
|  |  |  |  |  |  |  |  | min=-1; max=1 |

**Methods**

We tested whether LLMs could identify the distribution responsible for randomly generating a sample of data. We randomly drew samples from the beta, binomial, chi-square, gamma, geometric, lognormal, normal, Poisson, and uniform distributions, varying key parameters across the values reported in Table 1 and systematically varying sample size, *n*,to each value in the set, *N* = {10, 20, 35, 60, 100}. We drew 100 samples for each combination of data-generating distribution’s parameter settings and sample size, thus creating a data set consisting of 100 iterations x 98 parameter settings x 5 sample sizes equaling 49,000 test iterations (i.e. observations) for a given model. We then used these data as the “ground-truth” in tests that prompted three separate large language models, gpt-4-0613, gpt-4-turbo-2024-04-09, and gpt-4o-2024-08-06 to assess which distribution generated the ground-truth data.[[1]](#endnote-1)

Specifically, in each of 147,000 observations, we presented the models with a prompt that, first, introduced the LLM to the task by stating, “Below, I am going to provide you a stem-and-leaf-plot of a variable and the summary statistics for that variable. I would like you to tell me which distribution you think generated the data.” Next we presented the model with the stem-and-leaf plot from a sample generated by a given parameter-setting-sample-size combination and the summary statistics (minimum, 1st-quartile, median, mean, 3rd-quartile, and maximum) from that particular random sample. We also indicated whether the data were *discrete* or *not discrete*. We then repeated our description of the task that we wanted the LLM to perform and specified the format that we wanted the LLM to use when generating output. An example of these instructions, in their entirety, appears below:

*Below, I am going to provide you a stem-and-leaf-plot of a variable and the summary statistics for that variable. I would like you to tell me which distribution you think generated the data.*

*The decimal point is 1 digit(s) to the left of the |*

*0 | 392*

*2 | 0473*

*4 | 3*

*6 | 5*

*8 | 4*

*Here are the summary statistics: Min. = 0.03441; 1st Qu. = 0.141; Median = 0.2508; Mean = 0.3191; 3rd Qu. = 0.4031; Max. = 0.8438*

*The data type is: not discrete.*

*Please use the above information to predict which distribution generated the data. Please do not provide an explanation in your response. Please only state the name of the distribution in one word. Do not include the word 'distribution' following the name of the predicted distribution. Indeed, please read your response after writing it and ensure that you only provide the name of the distribution in your response.*

Each test iteration contained a prompt, such as the one above, with a stem-and-leaf plot and summary statistics from a unique random draw from the study’s generating distributions. The model responses contain considerable metadata that allows us to track input and output tokens and, most importantly, the model responses.

There is some variation in spelling and other aspects of these recorded outcomes. We obtained LLM output for the Lognormal distribution as “Log-normal”, “Lognormal”, and “Log-normal” and regarded all such values as specifying “Lognormal.” The “Negative Binomial” also appeared as the “NegativeBinomial,” which we treated as a specification of the Negative Binomial Distribution. Finally, we obtained peculiar results—if not errant ones—relating to the skewness of the distribution—specifically, responses listing “SkewNormal”, “Skew-normal”, “Skew-right”, “Skewed”, “Skewness”, and “Skewnormal.” We combined all of these findings under a global category, “Skewed.”

**Results**

Across the combinations of distributions, parameter settings, and sample sizes, the LLMs in our study rarely responded to our prompts with the correct name of the distribution that we used to generate the data displayed in the iteration’s stem-and-leaf plot and summary statistics (see Figure 1).

**Figure 1.**

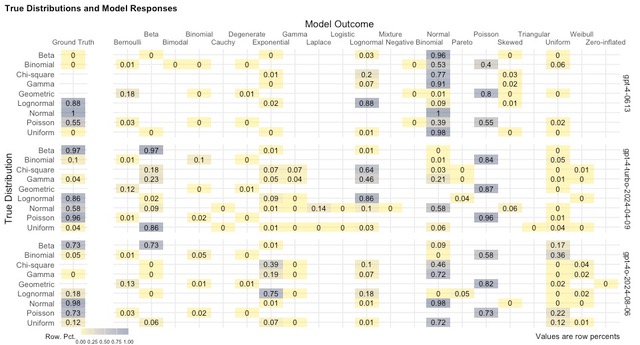


Figure 1 presents results obtained when testing all three models. The far left column displays Ground Truth; that is, what proportion of experiments did the output match the true distribution. First, the negative theme: no model ever recovers the chi-square or geometric distributions. gpt-4-0613 always recovers the normal (100%), is highly likely to recover the lognormal (88%), and recovers the Poisson (55%) slightly more than half of the time while basically never recovering the other true distributions. gpt-4-turbo-2024-04-09 consistently recovers the Beta (97%) and Poisson (96%) distributions, performs similarly with the lognormal (86%), is considerably worse with the normal distribution (58%), seldom recovers the binomial (10%), and rarely recovers the uniform (4%) or gamma (4%) distributions. gpt-4o-2024-08-06 is nearly perfect for true normal data (98%), performs well for the Beta (73%) and Poisson (73%) distributions, seldom recovers the lognormal (18%) or uniform (12%), and rarely recovers the binomial (5%) and Gamma (.1%) distributions.

**Discussion and Conclusion**

Our test showed that, only in a select number of cases, the LLMs in our study could identify the known and true distribution responsible for generating a random sample of data. Before discussing these findings in further depth, some caveats are n order. First, we did not attempt to use prompt-engineering techniques that might improve the LLM’s ability to identify the distribution that accurately generated the data. Second, we selected the values of key distribution parameters without an established rationale, thus our tests might have relied on a set of parameter values that resulted in data samples that happened to result in lower levels of accuracy in our studied LLMs. Third, we did not use the computer vision capabilities of the multimodal LLM in our study, but, instead, we passed stem-and-leaf plots to the LLM as text files; subsequent iterations of our study will present multimodal models with image data.

Recognizing these features of our study, our findings suggest that the LLMs in our investigation poorly identified the distributions responsible for generating particular subsets of randomly drawn data when supplied with a straightforward prompt describing the task. We cannot find anything anomalous about the prompts used in our study or the particular randomly generated data samples presented to the LLMs, thus we conclude from our study that the task we presented the LLMs rests outside the scope of queries to which it can respond successfully. This determination, we should note, does not indicate anything about LLMs abilities vis-à-vis humans, as humans themselves might not always have a sense of the right distribution to use to model particular data, but it does suggest that LLMs might not succeed at particular tasks in the data science workflow. These results should inform projections about the automation of data science activities.

**References**

Bie, T. D., Raedt, L. D., Hernández-Orallo, J., Hoos, H. H., Smyth, P., and Williams, C. K. I. (2022), "Automating Data Science," *Communications of the ACM*, 3 (3), 76-87.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M. T., and Zhang, Y. (2023), "Sparks of Artificial General Intelligence: Early experiments with GPT-4," *arXiv*. DOI: <https://doi.org/10.48550/arXiv.2303.12712>.

Jaimovitch-López, G., Ferri, C., Hernández-Orallo, J., Martínez-Plumed, F., and Ramírez-Quintana, M. J. (2023), "Can language models automate data wrangling?," *Machine Learning*, 112 (6), 2053-2082. DOI: 10.1007/s10994-022-06259-9.

Li, M. Y., Fox, E. B., and Goodman, N. D. (2024), "Automated Statistical Model Discovery with Language Models," *arXiv*, 2402.17879v1.

Manning, B. S., Zhu, K., and Horton, J. J. (2024), "Automated Social Science: Language Models as Scientist and Subjects," *National Bureau of Economic Research Working Paper Series*, 32381 (April). DOI: 10.3386/w32381.

Nejjar, M., Zacharias, L., Stiehle, F., and Weber, I. "LLMs for science: Usage for code generation and data analysis," *Journal of Software: Evolution and Process*, n/a (n/a), e2723. DOI: <https://doi.org/10.1002/smr.2723>.

Tayebi Arasteh, S., Han, T., Lotfinia, M., Kuhl, C., Kather, J. N., Truhn, D., and Nebelung, S. (2024), "Large language models streamline automated machine learning for clinical studies," *Nature Communications*, 15 (1), 1603. DOI: 10.1038/s41467-024-45879-8.

Tu, X., Zou, J., Su, W. J., and Zhang, L. (2023), "What Should Data Science Education Do with Large Language Models?," *arXiv*. DOI: <https://doi.org/10.48550/arXiv.2307.02792>.

Wang, J., and Chen, Y. 2023. A Review on Code Generation with LLMs: Application and Evaluation. In *2023 IEEE International Conference on Medical Artificial Intelligence (MedAI)*.

1. All computer code in the R language, input and output files in json format, and processing steps can be obtained from https://robertwwalker.github.io/LLM\_Distributions. [↑](#endnote-ref-1)