What Are the Odds?
Language Models Are Capable of Probabilistic Reasoning

Akshay Paruchuri*, **Jake Garrison, Shun Liao, John Hernandez,**
Jacob Sunshine, Tim Althoff, Xin Liu, Daniel McDuff
Google
akshay@cs.unc.edu, {althoff, xliucs, dmcduff}@google.com

Abstract

Language models (LM) are capable of remarkably complex linguistic tasks; however, numerical reasoning is an area in which they frequently struggle. An important but rarely evaluated form of reasoning is understanding probability distributions. In this paper, we focus on evaluating the probabilistic reasoning capabilities of LMs using idealized and real-world statistical distributions. We perform a systematic evaluation of state-of-the-art LMs on three tasks: estimating percentiles, drawing samples, and calculating probabilities. We evaluate three ways to provide context to LMs 1) anchoring examples from within a distribution or family of distributions, 2) real-world context, 3) summary statistics on which to base a Normal approximation. Models can make inferences about distributions, and can be further aided by the incorporation of real-world context, example shots and simplified assumptions, even if these assumptions are incorrect or misspecified. To conduct this work, we developed a comprehensive benchmark distribution dataset with associated question-answer pairs that we have released publicly.

1 Introduction

Language models (LMs) (Workshop et al., 2022; Touvron et al., 2023; Achiam et al., 2023) are versatile interfaces to knowledge, capable of remarkably complex linguistic tasks. Summarization of complex documents (Tang et al., 2023; Zhang et al., 2024b), reasoning over long passages of text (Shaham et al., 2022; Chen et al., 2023; Team et al., 2023) and zero-shot inference in specialist domains such as medicine (McDuff et al., 2023) are a few examples that demonstrate their abilities. While performance on primarily linguistic problems, can often be strong, effectiveness on operations that involve numerical reasoning is a domain that language models have struggled with (Kojima et al.,

Figure 1: **LMs & Probabilistic Reasoning.** Models can make inferences about distributions, but can be aided by the incorporation of real-world context, example shots and simplified assumptions, even if these assumptions are incorrect or misspecified.

*Work completed during an internship at Google.

2022). Difficulties handling numbers may be due to model pretraining formulations (e.g., using autoregressive next token prediction pretext tasks) or numerical token representations not necessarily being suited to mathematical reasoning (Bachmann and Nagarajan, 2024), or simply a limited representation of these types of tasks in the training corpora. Nevertheless, some work suggests prompting techniques can substantially improve LM performance on numerical reasoning tasks, indicating that relevant knowledge may already be encoded within these models (Imani et al., 2023).

A form of numerical reasoning that is important for interpreting many different forms of data is contextualizing an individual measurement or measurements (a sample or samples) within a population (a distribution). Drawing insights from data frequently requires comparing and contrasting a sample from other samples. This is because absolute values in isolation can be hard to interpret, without the context of how probable they are or how close they are to the maximum or minimum values observed across the population. Probabilistic reasoning is something that the human brain appears to do (Knill and Pouget, 2004) and that is an important component in cognition (Chater et al., 2006). Thinking probabilistically is efficient as one does not have to represent every detail of every sample that one observes, and instead can have the data summarized with a small number of parameters that describe the distribution (Lindskog et al., 2021). Research has shown that some probabilistic reasoning processes lead to superior performance; for example, people are more accurate at answering questions about statistical properties when they estimate the full distribution first (Goldstein and Rothschild, 2014). Yet, there are limited examples evaluating or improving on the probabilistic reasoning by designing LMs that reason over sets (Ozturkler et al., 2023).

Understanding the distributions is important in many contexts. In population level data it is important when gauging whether an individual behavior is normative (e.g., Is sleeping 8 hours normal for a college aged student?). In climatology, inferences about distributions of temperature or precipitation data on a given day of the year at a particular location are important when determining if observed events are typical or abnormal. Is a maximum temperature of 35°C likely to be observed in Seattle every year?

In this work we propose and define a set of probabilistic reasoning tasks and use them to evaluate the capabilities of LMs - *estimating percentiles*, *drawing samples*, and *calculating probabilities*. Next we evaluate the impact of additional real-world context and parametric assumptions (Normal distribution) using the task of *estimating percentiles* (task choice is motivated in Section 3).

To summarize our research questions:

1. Section 5.1: Provided with an idealized distribution, are language models able to accurately answer questions about them? Does this vary by the distribution family? Does providing prompt examples from different distributions in the same family or samples from the same distribution help? Do LMs simply repeat the nearest in-context example or is there evidence of more complex LM behavior?

2. Section 6.2: Can an LM answer questions about distributions in the world (e.g., income in the US population)? Are LMs able to retrieve statistics and answer questions about these distributions in a zero-shot manner?

3. Section 6.2: Using simple approximations such as assuming a Normal distribution, can we design prompts that lead to more accurate answers to probabilistic reasoning questions?

To answer these questions we develop a distribution dataset with associated question-answer pairs that we will release publicly. The dataset includes questions about 12 families of standard, idealized distributions (e.g., Normal or Power-law distributions) and distributions of real-world data from the domains of population health, climate, and finance. Code and additional results for our work can be found here: `https://github.com/yahskapar/LLMs-and-Probabilistic-Reasoning`.

2 Related Work

Language Models and Numerical Reasoning. Working with numbers is necessary for many everyday tasks. Yet, while large language models pretrained on vast numbers of documents exhibit impressive linguistic capabilities, they often struggle at tasks involving numerical reasoning (Saxton et al., 2019; Kojima et al., 2022) (for a survey see Lu et al. (2022)). Different approaches to numerical reasoning have been proposed, many focusing on logical reasoning of mathematical tasks (Geva et al., 2020; Imani et al., 2023; Yang

et al., 2022; Webb et al., 2023). In quantitative reasoning problems such as those in the domains of mathematics, science, and engineering, the process of fine-tuning models has been used to successfully remedy weaknesses (Lewkowycz et al., 2022). Automatic generation of data can be used as a way of obtaining training examples (Geva et al., 2020; Liu et al., 2022). The fact that specific prompting, such as providing examples, can improve the performance of LMs on numerical tasks, suggests that their training data may already include relevant information to perform these tasks (Imani et al., 2023; Yang et al., 2022). Benchmark datasets have helped the research community to develop these methods (e.g., (He-Yueya et al., 2023; Zhang et al., 2024a; Liu et al., 2024)) further and automatic evaluation of numerical reasoning problems has been proposed to help in cases where accuracy cannot be computed mathematically (Cobbe et al., 2021).

Numerical Reasoning Prompt Design. Prompts designed to handle automatically generated content (from an LM) were used to improve on numerical and scientific commonsense reasoning tasks (Liu et al., 2022). Algorithms and code are useful tools when working with numbers, chain-of-thought prompts have been designed to leverage these specifically to improve the performance of LMs on arithmetic problems (Imani et al., 2023; Merrill et al., 2024). Retrieval of correlated-examples (Yang et al., 2022), generating intermediate reasoning steps (Gao et al., 2023) and expressing reasoning as a program (Chen et al., 2022) are examples that can also help improve LM performance on mathematical and logic problems. Simple approaches such as zero-shot chain-of-thought (Kojima et al., 2022) exist and are capable of leveraging multi-step reasoning, but for probabilistic reasoning tasks lead to severely degraded performance due to generally poor numerical reasoning performance.

Probabilistic Reasoning and Cognition. Inspiration for AI systems is often drawn from our understanding of human cognition, cognitive science has revealed insights about how humans can think probabilistically (Cosmides and Tooby, 1996; Oaksford and Chater, 2001) and can build representations of relatively complex probability distributions (Lindskog et al., 2021), yet our perceptions of means and variances are subject to biases (Tversky and Kahneman, 1974). The thought processes people use when answering questions about distributions have an impact on their accuracy. Specifically, eliciting a full distribution before computing summary or sample statistics can make answers more accurate (Goldstein and Rothschild, 2014).

3 Defining Probabilistic Reasoning Tasks

Our probabilistic reasoning benchmark contains three distinct tasks that explore a language model's (LM's) context-free (idealized) understanding of basic, *idealized distributions*, we describe these tasks as follows:

Task 1: Estimating Percentiles. Given a distribution, the model is asked for the percentile a sample would appear in. A question is composed of a value (a sample) that is calculated given the target percentile. The answer is expected to be a numerical response from 0 to 100. The target percentiles utilized were $n^{st/th} = \{1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 99\}$. Language models responses to these questions are sampled 10 times with a random seed.

> *Return the percentile of the value {X} within a normal distribution with mean {Y} and standard deviation {Z}.*

Task 2: Drawing Samples. Given a distribution the model is asked to draw samples at random from it. A random seed is used for each sample and we repeat this 1000 times per distribution. The language model is explicitly instructed to avoid generating any code or using additional tools to perform the sampling. The answer is expected to be a numerical response.

> *Sample a number from the normal distribution with mean {X} and standard deviation {Y}.*

Task 3: Calculating Probabilities. Given a distribution the model is asked for the probability a sample from the distribution will fall between two given values. The target probabilities and the corresponding target ranges are computed based on a lower and upper quantile to form examples with different probabilities in the set $\mathcal{P} = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. The answer from the LM is expected to be a numerical response from 0 to 1.

Figure 2: **Distributions.** A visualization of the 12 idealized and 12 real-world distributions across the domains of health, finance, and climate involved in our evaluation.

> *Calculate the probability that a value falls between {W} and {X} in a normal distribution with mean {Y} and standard dev. {Z}.*

In order to evaluate the impact of real-world context on LM probabilistic reasoning, we explore distributions in the real-world that have additional real-world context (e.g., prior knowledge and expectations about US household incomes). We then leverage Task 1 of *estimating percentiles* to evaluate to what degree real-world context impacts performances. Task 1 is particularly well suited for this exploration, as we later demonstrate that this Task 1 elicits the highest variation in performance across distribution families and the number of in-context examples/shots (further detailed in Section 6). In Appendix A we provide full prompt templates for our three proposed benchmark tasks, as well as further details regarding template components and real-world distribution prompt templates.

4 Curating Data Distributions

We use two distinct datasets for the purpose of understanding the probabilistic reasoning capabilities of LMs - a dataset of *idealized distributions* and a dataset of *real-world distributions*. A visualization of both datasets is shown in Figure 2.

Idealized Distributions. We identify 12 families of distributions: Normal, Log-Normal, Skew-Normal, Exponential, Power Law, Uniform, Gamma, Gumbel, Poisson, Geometric, Binomial and Multinomial. These encompass sets of distributions identified (Frank, 2009) and tested in prior studies (Goldstein and Rothschild, 2014). When an LM is asked about an idealized distribution, a *distribution description* is provided with parameters that can range from simple parameters such as the mean and standard deviation (e.g., for a Normal distribution) to less easily interpretable parameters such as location and scale (e.g., for a Gumbel distribution). Note that the distribution description does not include any distrbution probability density function (PDF) or cumulative distribution function (CDF). All parameters are captured in a popular, public Python library for scientific computing - NumPy.

Real-World Distributions. We choose real-world distributions from the domains of health, finance, and climate for which there is presumed to be relevant information in the model's training set.

Health: We sample 100K Fitbit users from the U.S. population, aged 18-65, and with at least 10 days of data from the calendar year 2023. The dataset was gender and age balanced (see Appendix). We analyze four common wearable metrics **(A)** step count, **(B)** resting heart rate, **(C)** sleep duration,

and **(D)** exercise minutes. We aggregate this data to obtain averages for each user and ultimately distributions of daily metrics across 100K users.

> *In what percentile of the US population would someone with an average of 6000 steps per day be?*

Finance: We use public census data from the Census Bureau's American Community Survey (ACS) Public Use Microdata Sample (PUMS) (Ruggles et al., 2020) that contains measures of **(E)** annual income, **(F)** monthly gross rent, **(G)** annual electricity costs, and **(H)** annual water costs ($) for individuals and households in the US. We select data from the calendar year 2018.

> *In what percentile of US households would someone with a household income of $70,000 be?*

Climate: We use the public Global Historical Climatology Network daily (GHCNd) dataset (Menne et al., 2012) maintained by the National Oceanic and Atmospheric Administration (NOAA) that contains average daily measures for variables of **(I)** temperature, **(J)** precipitation, **(K)** wind speed, and **(L)** humidity level for weather stations across the continental United States. We consider data from the calendar year 2018 and filter erroneous measures indicated by measurement quality flags built into the GHCNd dataset.

> *In what percentile would an average temperature of 20 degrees Celsius be in the USA?*

5 Experimental Setup

5.1 Idealized Distributions

Zero-shot Performance. We evaluate the zero-shot performance of three LMs (*Gemini 1.0 Ultra*, *GPT4-Turbo*, and *GPT3.5-Turbo*) across our three tasks and 12 idealized distributions. LM prompts are generated as formulated in Section 3.

N-Shot Performance. We propose two types of shots that might be reasonably employed for the tasks - *within distribution family distribution shots* and *within distribution shots*. *Within distribution family distribution shots* are examples from a different distribution from the same family as the current distribution in question. The distribution parameters of the variant are randomly sampled from a specified range of reasonable parameter values. For example, if we are asking for a percentile of a value in a normal distribution with a mean of 100 and a standard deviation of 10, and we are providing three shots, the randomized shots may be generated from three variant normal distributions with means of 108, 118, and 112 and corresponding standard deviations of 13, 16, and 10. *Within distribution shots* entail shots from the same distribution that is being asked about in a question.

To help contextualize the performance in the N-shot experiments, for the task of *estimating percentiles* we compare both shot types to a baseline where the answer is picked based on the nearest corresponding target percentile value in the shot examples (i.e., the nearest neighbor). This baseline does not perform any interpolation between percentiles, which would be required for optimal performance. If the LM performance exceeds this baseline performance, it would suggest that the LMs does perform some form of interpolation, instead of simply reciting in-context examples.

We explicitly avoid using shots that involve an answer that could be an answer to one of our proposed questions. Specifically, we use $n^{th}_{shots} = \{5, 15, 25, 35, 45, 55, 65, 75, 85, 95\}$ and $\mathcal{P}_{shots} = \{0.05, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95\}$. For example, if we are asking for a percentile of a value in a Normal distribution with a mean of 100 and a standard deviation of 10, and we are providing three shots, the mapped shots will be generated from the same normal distribution and correspond to 35.0, 55.0, and 75.0. We sample LM responses 10 times per question with a random seed. We provide further details, including examples per shot type and shot count, in our Section 8.

5.2 Real-World Distributions

Zero-shot Performance. We evaluate the zero-shot performance of three LMs (*Gemini 1.0 Ultra*, *GPT4-Turbo*, and *GPT3.5-Turbo*) across the proposed task of *estimating percentiles* in order to evaluate an LM's understanding of probabilistic reasoning of the real-world distributions in the domains of health, finance, and climate mentioned in Section 4. We design prompts that contain information about the corresponding real-world data, such as where it was sourced from, the year in time for which the data is relevant, and any relevant filtering that was done, this acts as context that we

Figure 3: **Results on Idealized Distributions.** Model results (top) estimating percentiles, (middle) drawing samples, (bottom) estimating probabilities, for five common distributions (see Appendix B for results on all distributions).

refer to as "Real-World Context" in the remainder of the paper. The prompts are built using 11 unique target values that correspond to ground truth percentiles generated from the real-world data. We sample LM responses 10 times per question with a random seed.

Performance by Context. To investigate the effects of context, we compare two conditions: 1) questions about an idealized distribution with comparable shape and parameters of the real-world distribution but no other context, and 2) questions about the distribution with Real-World Context (see examples in Appendix 8). Both conditions involve a distribution description as mentioned in Section 3. The prompts are again built using 11 unique target values that correspond to ground truth percentiles generated from the real-world data. We sample *Gemini 1.0 Ultra* responses 10 times per question with a random seed.

Simplified Assumptions. Certain real-world distributions found in Section 4 are Non-Normal. For example, annual income follows a Power Law distribution. Despite not all distributions found in Section 4 being perfectly Normal distributions, we devise a prompting strategy that involves treating any distribution in the prompt as a normal distribution with a specified mean and standard deviation. This approach can be further justified by the fact that, despite characteristics such as skewedness being present in real-world distributions, many distributions are similar to a Normal distribution. We quantitatively reinforce this observation using the Kolmogorov–Smirnov test (Chakravarti et al., 1967) to show that even if the Normal equivalent is not the best fit for a given real-world distribution, it can be remarkably close as evidenced by the K-S statistic. Additionally, we compare our proposed Normal approximation approach to simply providing a question involving a real-world distribution with three *within distribution* shots.

6 Experimental Results

We organize our results based on the research questions posed in Section 1. Alongside a concise answer in bold, we provide discussion based on our analysis of the experimental results.

6.1 Idealized Distributions

Q1: Are LMs able to accurately answer questions about idealized distributions in a zero-shot setting? *Answer*: **It varies, performance on some idealized distributions is better than**

Model	Percentiles (%)	Sampling (K-S)	Probabilities (%)
GPT3.5-Turbo	25.7 ± 3.11	0.73 ± 0.07	32.7 ± 2.38
GPT4-Turbo	$\mathbf{14.9 \pm 2.39}$	$\mathbf{0.59 \pm 0.08}$	21.0 ± 2.11
Gemini 1.0 Ultra	16.5 ± 2.67	0.76 ± 0.09	$\mathbf{19.4 \pm 2.26}$

Table 1: **Aggregated zero-shot task performance across different LMs.** We evaluate zero-shot performance for tasks such as percentiles, sampling, and probabilities using Gemini 1.0 Ultra, GPT4-Turbo, and GPT3.5-Turbo. For the tasks of estimating percentiles and calculating probabilities, results are reported as Mean Absolute Error (MAE) $\pm$ Standard Error (σ_M). For the task of drawing samples, the Mean K-S statistic $\pm$ Standard Error (σ_M) is reported with all reported values having $p < 0.01$.

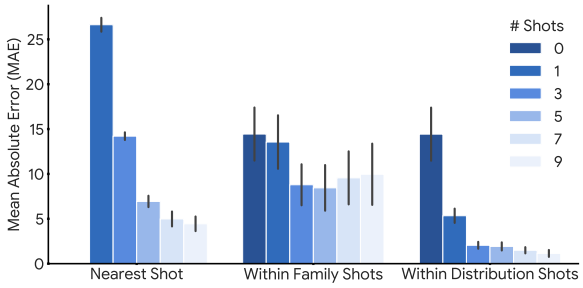

Figure 4: **Language models appear to interpolate between in-context examples.** Comparison of within family and within distribution shot types to a baseline where the answer is based on the nearest corresponding shot to the target percentile value (nearest neighbor), importantly the baseline does not perform any interpolation between percentiles.

others. *Discussion*: Language model performance varied considerably across families of distribution. Zero-shot performance on the percentile task was best for the uniform (MAE = 0.54%) and normal (MAE = 2.29%) distributions (see Figure 3 for examples and Figure 6 for more detailed results). Furthermore, the performance of LMs on answering questions about distributions varied by task. Estimating percentiles showed rather impressive zero-shot performance (MAE $\bar{\mu}$ = 16.52, min = 0.54, max = 28.36). Calculating probabilities was worse on average (MAE μ = 21.48, min = 9.03, max = 32.53) and sampling performance was generally poor in the zero-shot case. In all cases, providing shots (for different percentiles, samples or probabilities) within a distribution improved the performance substantially (Percentile $\Delta_{0 \to 1shot}$ = +59.14%, Sampling $\Delta_{0 \to 1shot}$ = +55.26% K-S stat., Probability $\Delta_{0 \to 1shot}$ = +70.13%).

Q2: Does providing prompt examples from different distributions in the same family or samples from the same distribution help? *Answer*: **Providing *within distribution* shots helps more than *within family* shots.** *Discussion*: As illustrated in Figure 4, providing prompt examples (shots) from different distributions from the same family has less impact on the performance than providing prompt examples from the same distribution in the question.

Q3: Do LMs simply repeat the nearest in-context example? *Answer*: **No, they appear to perform some interpolation that is superior to a nearest-in-context baseline.** *Discussion*: We observe that LM performance exceeds that of this baseline (Figure 4) which suggests that LMs perform some kind of interpolation, instead of simply reciting in-context examples.

6.2 Real-World Distributions

Q4: What is the zero-shot accuracy of an LM on distributions in the world (e.g., income in the US population)? *Answer*: **It varies, performance on some real-world distributions is better than others.** *Discussion*: As shown in Table 2, LMs are capable of varying degrees of zero-shot performance given different kinds of context. We consider the real-world context as the primary baseline in our investigation of real-world distributions, in contrast to idealized, context-free versions of the same real-world distributions and added context that simplifies assumptions (e.g., Normal approximation). On average with real-world context, *Gemini 1.0 Ultra* has superior zero-shot performance on distributions in the climate domain. In contrast, *GPT4-Turbo* has superior zero-shot performance in the health domain. Both *Gemini 1.0 Ultra* and *GPT4-Turbo* had comparable performance on the finance domain while being superior to *GPT3.5-Turbo*. We attribute these differences in zero-shot performance to underlying differences in the large amounts of training data used to train LMs, especially in the case *Gemini 1.0 Ultra* and *GPT4-Turbo*.

Q5: Does the provided real-world context help with probabilistic reasoning performance? *Answer*: **Yes.** *Discussion*: Figure 5 details both distribution-wise and domain-wise results using *Gemini 1.0 Ultra* with our four context categories - idealized, real-world context, real-world context with Normal approximation, and real-world context with 3 shots. On average, adding real world-context improves performance and adding real-world context with a Normal approximation improves performance still further. This trend is true on aggregate but not for all individual distributions (e.g., resting heart rate). This observation appears to be

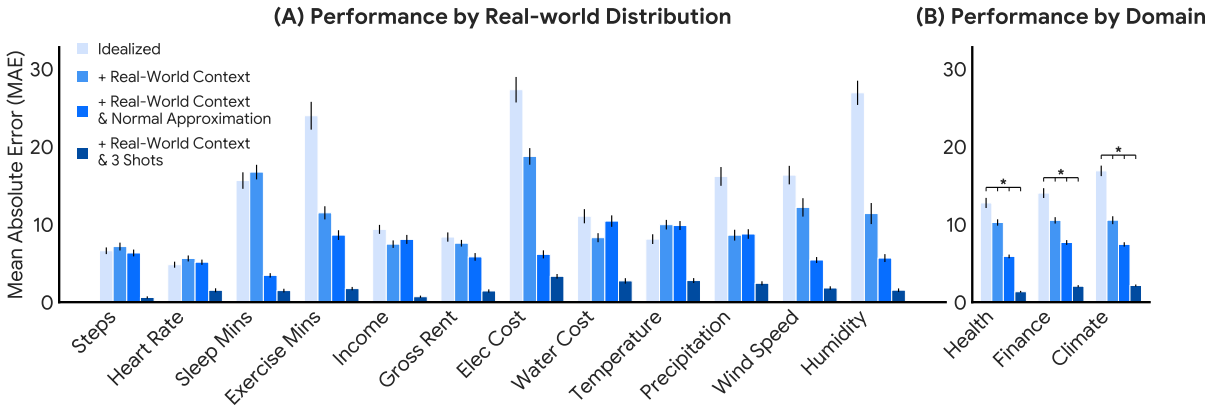

Figure 5: **Inferences can be aided by context and simplified assumptions.** Mean absolute error in calculating percentiles for real-world distributions with different prompts, including idealized distributions without real-world context, added real-world context, and a Normal approximation approach that simplifies parameter content. (*) designates $p < 0.05$ for all possible pairs using the Wilcoxon signed-rank test.

	Health			Finance			Climate		
Model	**Idealized**	**Real-world Con.**	**Norm. Approx.**	**Idealized**	**Real-world Con.**	**Norm. Approx.**	**Idealized**	**Real-world Con.**	**Norm. Approx.**
GPT3.5-Turbo	20.5 ± 9.62	20.3 ± 8.51	6.81 ± 0.68	17.7 ± 4.54	20.4 ± 2.88	7.55 ± 0.77	22.7 ± 6.88	25.7 ± 6.32	7.90 ± 0.22
GPT4-Turbo	$\mathbf{11.0 \pm 4.94}$	$\mathbf{4.92 \pm 3.18}$	$\mathbf{3.15 \pm 00.76}$	$\mathbf{8.99 \pm 1.18}$	10.7 ± 3.24	$\mathbf{5.50 \pm 0.48}$	18.5 ± 6.53	15.2 ± 5.13	$\mathbf{4.94 \pm 0.58}$
Gemini 1.0 Pro	25.30 ± 8.41	11.51 ± 1.06	10.42 ± 1.32	29.35 ± 3.72	11.77 ± 0.92	10.10 ± 1.01	26.20 ± 5.44	18.67 ± 2.01	16.53 ± 1.94
Gemini 1.0 Ultra	12.8 ± 4.43	10.3 ± 2.49	5.89 ± 1.09	14.0 ± 4.47	$\mathbf{10.5 \pm 2.75}$	7.62 ± 1.06	$\mathbf{16.9 \pm 3.86}$	$\mathbf{10.5 \pm 0.79}$	7.43 ± 1.11

Table 2: **Zero-shot performance by domain and context category across different LMs.** All results are reported as Mean Absolute Error (MAE) $\pm$ Standard Error (σ_M) with (%) units.

restricted to distributions that already have a reasonable baseline performance, we suspect that the saturated performance conflicts with the model's ability to leverage real-world context and simpler assumptions such as the Normal approximation. Additionally, certain distributions such as household income show a decrease in performance when Normal approximation is applied, likely because the household income distribution follows a Power Law distribution. It is unhelpful to apply a Normal approximation on distributions that differ greatly from a normal distribution. We empirically show this with the same set of 12 idealized distributions in Appendix D.3. Lastly, we note that real-world context with 3 shots has the best performance. This is unsurprising, and furthermore does not invalidate the impact of simplified assumptions such as Normal approximation, which can be more efficient due to not relying on 3 shots.

Q6: Do parametric assumptions such as a Normal approximation as a prompt design strategy improve performance? *Answer*: **Yes.** *Discussion*: On average across all domains, yes. The simple assumption of a Normal distribution performs well and when paired with real-world context, consistently improves performance on real-world distributions (see Figure 5). This seems reasonable given the aforementioned internal, potentially incorrect representations of real-world distributions that LMs can have, and subsequently how stats such as mean and standard deviation can help correct the LM's baseline knowledge. It is perhaps surprising that performance improves relatively consistently, despite the real-world distributions often differing from an idealized Normal distribution and therefore the LM is being conditioned on a misspecified, yet still helpful, model.

Q7: How do simpler assumptions such as a Normal approximation compare to providing three few-shot examples? *Answer*: **Providing three few-shot examples is generally better.** *Discussion*: Generally speaking, providing three few-shot examples is better and provides superior performance across our proposed domain-specific datasets of health, finance, and climate.

7 Conclusion

LMs are able to answer questions about idealized and real-world distributions, with real-world results suggesting there is some internal representation that enables modeling or interpolation from distribution parameters. The probabilistic reasoning performance of LMs varies, with certain distributions (e.g., uniform, normal) having much better performance in contrast to other distributions (e.g., log-normal, skew-normal). Within distribution shots and context can improve probabilistic

reasoning performance, as can a simplified Normal approximation.

8 Limitations

Numerical calculation and reasoning remains an area in which language models, even very large models, tend to perform poorly. Making approximations based on distributions is effective; however, it may also be a source of potential biases. Our experiments have not focused on a deep exploration of the ability of language models to represent and answer questions about extreme values, such as outliers in distributions. Our results do suggest that language model particularly struggle with accounting for extreme values in very skewed (long-tail) distributions. In computing percentiles the model would often overestimate the percentiles (and thus underestimate the presence of extreme values) - see the Power Law family results in Fig. 6 of our appendices. Our work shows that, despite some promising zero-shot performance and ways to improve that performance, language models require more improvements with Non-Uniform and Non-Normal distributions before they are capable of being relied on for probabilistic reasoning of real-world distributions that follow other distributions (e.g., Power Law). We hope our insights as a part of this work, as well as our proposed tasks and datasets critical to probabilistic reasoning and to be publicly released, prove valuable to the community at large and their efforts to make language models more useful, safer, and ultimately more reliable.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Gregor Bachmann and Vaishnavh Nagarajan. 2024. The pitfalls of next-token prediction. *arXiv preprint arXiv:2403.06963*.

Indra Mohan Chakravarti, Radha Govira Laha, and Jogabrata Roy. 1967. Handbook of methods of applied statistics. *Wiley Series in Probability and Mathematical Statistics (USA) eng*.

Nick Chater, Joshua B Tenenbaum, and Alan Yuille. 2006. Probabilistic models of cognition: Conceptual foundations. *Trends in cognitive sciences*, 10(7):287–291.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Leda Cosmides and John Tooby. 1996. Are humans good intuitive statisticians after all? rethinking some conclusions from the literature on judgment under uncertainty. *cognition*, 58(1):1–73.

Steven A Frank. 2009. The common patterns of nature. *Journal of evolutionary biology*, 22(8):1563–1585.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 946–958.

Daniel G Goldstein and David Rothschild. 2014. Lay understanding of probability distributions. *Judgment and Decision making*, 9(1):1–14.

Joy He-Yueya, Gabriel Poesia, Rose E Wang, and Noah D Goodman. 2023. Solving math word problems by combining language models with symbolic solvers. *arXiv preprint arXiv:2304.09102*.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models. *arXiv preprint arXiv:2303.05398*.

David C Knill and Alexandre Pouget. 2004. The bayesian brain: the role of uncertainty in neural coding and computation. *TRENDS in Neurosciences*, 27(12):712–719.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.

Marcus Lindskog, Pär Nyström, and Gustaf Gredebäck. 2021. Can the brain build probability distributions? *Frontiers in Psychology*, 12:596231.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated knowledge prompting for commonsense reasoning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3154–3169.

Xiao Liu, Zirui Wu, Xueqing Wu, Pan Lu, Kai-Wei Chang, and Yansong Feng. 2024. Are llms capable of data-based statistical and causal reasoning? benchmarking advanced quantitative reasoning with data. *arXiv preprint arXiv:2402.17644*.

Pan Lu, Liang Qiu, Wenhao Yu, Sean Welleck, and Kai-Wei Chang. 2022. A survey of deep learning for mathematical reasoning. *arXiv preprint arXiv:2212.10535*.

Daniel McDuff, Mike Schaekermann, Tao Tu, Anil Palepu, Amy Wang, Jake Garrison, Karan Singhal, Yash Sharma, Shekoofeh Azizi, Kavita Kulkarni, et al. 2023. Towards accurate differential diagnosis with large language models. *arXiv preprint arXiv:2312.00164*.

Matthew J Menne, Imke Durre, Russell S Vose, Byron E Gleason, and Tamara G Houston. 2012. An overview of the global historical climatology network-daily database. *Journal of atmospheric and oceanic technology*, 29(7):897–910.

Mike A. Merrill, Akshay Paruchuri, Naghmeh Rezaei, Geza Kovacs, Javier Perez, Yun Liu, Erik Schenck, Nova Hammerquist, Jake Sunshine, Shyam Tailor, Kumar Ayush, Hao-Wei Su, Qian He, Cory Y. McLean, Mark Malhotra, Shwetak Patel, Jiening Zhan, Tim Althoff, Daniel McDuff, and Xin Liu.

2024. Transforming wearable data into health insights using large language model agents. *Preprint*, arXiv:2406.06464.

Mike Oaksford and Nick Chater. 2001. The probabilistic approach to human reasoning. *Trends in cognitive sciences*, 5(8):349–357.

Batu Ozturkler, Nikolay Malkin, Zhen Wang, and Nebojsa Jojic. 2023. Thinksum: Probabilistic reasoning over sets using large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1216–1239.

Steven Ruggles, Sarah Flood, Ronald Goeken, Josiah Grover, Erin Meyer, Jose Pacas, and Matthew Sobek. 2020. Ipums usa: version 10.0 [dataset]. *Minneapolis, Mn: Ipums*, 10:D010.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. *arXiv preprint arXiv:1904.01557*.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, et al. 2022. Scrolls: Standardized comparison over long language sequences. *arXiv preprint arXiv:2201.03533*.

Liyan Tang, Zhaoyi Sun, Betina Idnay, Jordan G Nestor, Ali Soroush, Pierre A Elias, Ziyang Xu, Ying Ding, Greg Durrett, Justin F Rousseau, et al. 2023. Evaluating large language models on medical evidence summarization. *npj Digital Medicine*, 6(1):158.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Amos Tversky and Daniel Kahneman. 1974. Judgment under uncertainty: Heuristics and biases: Biases in judgments reveal some heuristics of thinking under uncertainty. *science*, 185(4157):1124–1131.

Taylor Webb, Keith J Holyoak, and Hongjing Lu. 2023. Emergent analogical reasoning in large language models. *Nature Human Behaviour*, 7(9):1526–1541.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Zhicheng Yang, Jinghui Qin, Jiaqi Chen, Liang Lin, and Xiaodan Liang. 2022. Logicsolver: Towards interpretable math word problem solving with logical prompt-enhanced learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1–13.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Dylan Slack, Qin Lyu, et al. 2024a. A careful examination of large language model performance on grade school arithmetic. *arXiv preprint arXiv:2405.00332*.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2024b. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57.

Overview of Appendices

The appendix is organized as follows:
Appendix A contains additional experimental details related to the usage of idealized distribution prompts, examples of idealized distribution prompts used in Section 5.1, per task, as well as distribution description examples, and examples of few-shots.
Appendix B contains 3x4 summary figures corresponding to idealized distribution results described in Section 6.1.
Appendix C contains examples of real-world distribution prompts used in Section 5.2.
Appendix D contains additional model-wise experimental results that extend Table 1 and Table 2, normal approximation results that show whether or not invoking the true distribution name makes a difference, and results for Chain-of-Thought (CoT) and code tool-use.
Appendix E contains our broader impacts statement.

A Idealized Distribution

A.1 Experimental Details

To systematically investigate performance on *idealized distributions* using the three proposed tasks of *estimating percentiles*, *drawing samples*, and *calculating probabilities*, our method involves sets of questions, or in the case of *drawing samples*, a command, that systematically tests the model's knowledge of a given distribution. In addition to investigating our proposed tasks in a zero-shot setting, we consider two different ways of providing in-context examples (shots) in the prompt - *within family shots* and *within distribution shots*.

Zero-shot Performance. We evaluate the zero-shot performance of three LMs (*Gemini 1.0 Ultra*, *GPT4-Turbo*, and *GPT3.5-Turbo*) across our proposed tasks of *estimating percentiles*, *drawing samples*, and *calculating probabilities* in order to evaluate an LM's understanding of probabilistic reasoning of the 12 idealized distributions described in Section 4. LM prompts are generated as formulated in Section 3.

Performance by Shot Type. We propose two shot types used across the aforementioned tasks - *within distribution family distribution shots* and *within distribution shots*. *Within distribution family distribution shots* entail randomized shots from a different variant of the distribution being asked about in a question. The distribution parameters of the variant are randomly sampled from a specified range of reasonable parameter values. For example, if we are asking for a percentile of a value in a normal distribution with a mean of 100 and a standard deviation of 10, and we are providing three shots, the randomized shots may be generated from three variant normal distributions with means of 108, 118, and 112 and corresponding standard deviations of 13, 16, and 10. *Within distribution shots* entail shots from the exact same distribution that is being asked about in a question. The shots are mapped per shot count to allow for a reasonable spread of shots throughout the distribution.

Additionally, for the task of *estimating percentiles*, we compare both shot types to a baseline where the LM is asked to pick from one or more shots' answer based on the nearest corresponding target percentile value. This baseline represents a nearest neighbor approach using the set of in-context examples. This baseline makes appropriate use of the information given to the model, but importantly does not perform any interpolation between percentiles, which would be required for strong performance. If LM performance exceeded baseline performance, this would suggest that LMs perform some kind of interpolation, instead of simply reciting in-context examples.

To avoid biasing our results, in the case of the *estimating percentiles* and *calculating probabilities* tasks, we explicitly avoid using shots that involve an answer (percentile or probability respectively) that could potentially be an answer to one of our proposed questions. Specifically, we use $n^{th}_{shots} = \{5, 15, 25, 35, 45, 55, 65, 75, 85, 95\}$ and $\mathcal{P}_{shots} = \{0.05, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95\}$. For example, if we are asking for a percentile of a value in a normal distribution with a mean of 100 and a standard deviation of 10, and we are providing three shots, the mapped shots will be generated from the same normal distribution and correspond to 35.0, 55.0, and 75.0. We sample *Gemini 1.0 Ultra* responses 10 times per question with a random seed. Then, we elect to average and effectively use all answers as a part of our final evaluation in order to help capture the variability in language model responses and provide a broader understanding of their reasoning capabilities. This effectively is a form of self-consistency. We provide further details, including examples per shot type and shot count, in our Section 8.

Key LM Parameters for Reproducibility. Language models typically have additional parameters,

such as temperature and sampling strategies, which have default settings that can vary from model to model. For *Gemini 1.0 Ultra*, *GPT4-Turbo*, and *GPT3.5-Turbo*, we utilize a default temperature of 0.7 for all of our experiments because we empirically discovered that a temperature of 0.7 to 0.9 with a random seed yielded similar, optimal performance on a hold-out dataset. A temperature of 0.7 also happens to be a default for many language models, such as various versions of the GPT and Gemini family models. Unless noted otherwise, results are obtained using a random seed.

Additional hyperparameters, such as frequency and brevity penalties, are not utilized in our experiments. We define a *frequency penalty* as a penalty that is applied proportional to how many times a token has appeared in the response and prompt. We define a *brevity penalty* as a penalty that targets responses that are very short, for example translations that contain only a few words. The nature of our proposed probabilistic reasoning questions, where the final answer from the instruction-tuned language model is always a single, numerical answer that is constrained by the prompt (for example, percentile answers of 25.3, 55.7, and 82.1) means that in both cases these penalties are not particularly relevant or effective at producing optimal answers. We utilized the default decoding approach for each model, with a default top-p (0.95), and where applicable, top-k (40).

A.2 Prompt Examples

Percentile Example

You are an expert on statistics. Your task is to estimate the percentile of a number within a specific distribution. Answer with just a numerical response from 0 to 100. Make sure your final answer is enclosed by xml tags <answer> and </answer>.

Here are some examples to help you understand the task:

{few_shot_examples}

Consider the following distribution:
{distribution_description}

Here is your question:
Question:
What is the percentile of the value **{target_number}** within the provided distribution?
Answer:

Sampling Example

You are an expert on statistics. Your task is to sample a number from a given distribution. Do not write any code or use any additional tools to perform the sampling. Answer with just a numerical response. Make sure your final answer is enclosed by xml tags <answer> and </answer>.

Here are some examples to help you understand the task:

{few_shot_examples}

Consider the following distribution:
{distribution_description}

Instruction: Sample a number from the given distribution and output only the numerical value.

Probability Example

You are an expert on statistics. Your task is to estimate the probability of being in a range of values within a given distribution. Answer with just a numerical response from 0 to 1, representing the probability. Make sure your final answer is enclosed by xml tags <answer> and </answer>.

Here are some examples to help you understand the task:

{few_shot_examples}

Consider the following distribution:
{distribution_description}

Here is your question:
Question:
Considering only values including and between the 1st percentile and the 99th percentile, what is the probability that a value from the provided distribution is between **{lower_target_number}** and **{upper_target_number}**?
Answer:

A.2.1 Distribution Description Examples

Normal

Distribution Type: Normal Distribution
Mean: **{mean}**
Standard Deviation: **{std}**

Log-Normal

Distribution Type: Log-Normal Distribution
Characteristics: This distribution models values that are the result of the multiplicative product of many independent random variables, such as income levels, stock prices, or city sizes Log Mean (mu): **{mean}**
Log Sigma (sigma): **{sigma}**
These parameters mean that the natural logarithm of the values follows a normal distribution with the specified mean and standard deviation.

Exponential

Distribution Type: Exponential Distribution
Characteristics: Models the time between events in a process where events occur continuously and independently at a constant average rate.
Rate: **{rate}** (The average number of events per unit time is **{1/rate:.2f}**.)

Power Law

Distribution Type: Power Law Distribution
Characteristics: Known for its heavy tails suitable for describing phenomena with a high incidence of extreme values.
Alpha: **{alpha}** (Controls the tail heaviness—the smaller the alpha, the fatter the tail.)
Xmin: **{xmin}** (Minimum value for which the power law behavior holds.)

Uniform

Distribution Type: Uniform Distribution
Characteristics: All values within the interval have equal probability of occurring.
Min: **{a}** (Minimum value of the distribution.)
Max: **{b}** (Maximum value of the distribution.)

Gamma

Distribution Type: Gamma Distribution
Characteristics: Used to model waiting times and life data among other things.
Shape: **{shape}** (Controls the skewness of the distribution.)
Scale: **{scale}** (Controls the spread of the distribution.)

Skew-Normal

Distribution Type: Skew-Normal Distribution
Characteristics: A generalization of the normal distribution to accommodate skewness.
Location: **{location}** (Shifts the distribution along the x-axis.)
Scale: **{scale}** (Controls the spread of the distribution.)
Skew: **{skew}** (Determines the direction and degree of skewness.)

Gumbel

Distribution Type: Gumbel Distribution
Characteristics: Often used to model the distribution of extreme values.
Location: **{loc}** (Centers the distribution.)
Scale: **{scale}** (Controls the spread of the distribution.)

Poisson

Distribution Type: Poisson Distribution
Characteristics: Suitable for modeling the number of events happening in a fixed interval of time or space.
Lambda: **{lam}** (Average rate of events per interval.)

Geometric

Distribution Type: Geometric Distribution
Characteristics: Models the number of trials until the first success.
Probability of Success: **{p}**

Binomial

Distribution Type: Binomial Distribution
Characteristics: Describes the number of successes in a fixed number of trials with a given probability of success.
Trials: **{n}** (Total number of trials.)
Probability of Success: **{p}** (Probability of success in each trial.)

Multinomial

Distribution Type: Multinomial Distribution
Characteristics: Generalizes the binomial distribution for scenarios where each trial can result in more than two outcomes.
Trials: **{n}** (Total number of trials.)
Probabilities: **{probs}**

A.2.2 Examples of Few-shots

Within Family Shots

Example 1:
Distribution:

Distribution Type: Normal Distribution
Mean: 80
Standard Deviation: 10

Question:
What is the percentile of 74.722 within the provided distribution?
Answer:
<answer>30.0</answer>

Example 2:
Distribution:

Distribution Type: Normal Distribution
Mean: 108
Standard Deviation: 13

Question:
What is the percentile of 107.903 within the provided distribution?
Answer:
<answer>50.0</answer>

Example 3:
Distribution:

Distribution Type: Normal Distribution
Mean: 82
Standard Deviation: 8

Question:
What is the percentile of 88.708 within the provided distribution?
Answer:
<answer>80.0</answer>

Within Distribution Shots

Example 1:
Distribution:

Distribution Type: Normal Distribution
Mean: 100
Standard Deviation: 10

Question:
What is the percentile of 96.183 within the provided distribution?
Answer:
<answer>35.0</answer>

Example 2:
Distribution

Distribution Type: Normal Distribution
Mean: 100
Standard Deviation: 10

Question:
What is the percentile of 101.298 within the provided distribution?
Answer:
<answer>55.0</answer>

Example 3:
Distribution:

Distribution Type: Normal Distribution
Mean: 100
Standard Deviation: 10

Question:
What is the percentile of 106.802 within the provided distribution?
Answer:
<answer>75.0</answer>

B Idealized Distributions Results Summaries

Figure 6: **Percentile Results.** Zero and three-shot (within distribution) results for *returning percentile* estimations in each of the 12 families of distributions.

Figure 7: **Sampling Results.** Zero and three-shot (within distribution) results for *drawing samples* (single repeated draws) in each of the 12 families of distributions.

Figure 8: **Probability Results.** Zero and three-shot (within distribution) results for *calculating probabilities* in each of the 12 families of distributions.

C Real-World Distribution Prompts

Health Example

You are an expert on population health and wearable fitness devices. Your task is to estimate the percentile of a given average step count value for a population that regularly uses Fitbit devices and is active on a daily basis. The data is filtered for individuals aged 18-65. The data is age-balanced and gender-balanced, and pertains to the U.S. population only. Do not use any additional tools such as code generation or search engines. Answer with just a numerical response from 0 to 100. Make sure your answer is enclosed by xml tags <answer> and </answer>.

Consider the following parameters that describe a normal distribution of this data:

Mean: 8366.971
Standard Deviation: 3291.940

Here is your question:
Question:
What is the percentile of the average step count value **{target_number}** steps for users of Fitbit devices? Do not use any additional tools such as code generation or search engines. Answer with just a numerical response from 0 to 100. Make sure your answer is enclosed by xml tags <answer> and </answer>.
Answer:

Finance Example

You are an expert on finance and statistics. Your task is to estimate the percentile of a given annual household income within the population using data from the year 2018 in the United States, sourced from the Census Bureau's American Community Survey (ACS) Public Use Microdata Sample (PUMS). Do not use any additional tools such as code generation or search engines. Answer with just a numerical response from 0 to 100. Make sure your answer is enclosed by xml tags <answer> and </answer>.

Consider the following parameters that describe a Gumbel distribution of this data:

Mean: 66028.713
Standard Deviation: 53616.018

Here is your question:
Question:
What is the percentile of an annual household income value of ${**target_number**}? Do not use any additional tools such as code generation or search engines. Answer with just a numerical response from 0 to 100. Make sure your answer is enclosed by xml tags <answer> and </answer>.
Answer:

Climate Example

You are an expert on climate science and statistics. Your task is to estimate the percentile of a given average temperature value using data from U.S. weather stations in the year 2018, sourced from the National Oceanic and Atmospheric Administration (NOAA) Global Historical Climatology Network Daily (GHCNd). Do not use any additional tools such as code generation or search engines. Answer with just a numerical response from 0 to 100. Make sure your answer is enclosed by xml tags <answer> and </answer>.

Consider the following parameters that describe a normal distribution of this data:

Mean: 10.643
Standard Deviation: 12.628

Here is your question:
Question:
What is the percentile of an average temperature of **{targe_number}** degrees Celsius? Do not use any additional tools such as code generation or search engines. Answer with just a numerical response from 0 to 100. Make sure your answer is enclosed by xml tags <answer> and </answer>.
Answer:

	Health			Finance			Climate		
Model	**Norm. Approx.**	**CoT**	**RWC + Code**	**Norm. Approx.**	**CoT**	**RWC + Code**	**Norm. Approx.**	**CoT**	**RWC + Code**
Gemini 1.0 Ultra	$\mathbf{5.89 \pm 1.09}$	6.45 ± 4.91	6.62 ± 1.03	$\mathbf{7.62 \pm 1.06}$	9.45 ± 1.26	8.46 ± 0.94	$\mathbf{7.43 \pm 1.11}$	10.48 ± 1.69	8.56 ± 0.92

Table 3: **Zero-shot performance by domain and context category across different LMs.** RWC = Real-world Context. All results are reported as Mean Absolute Error (MAE) $\pm$ Standard Error (σ_M) with (%) units.

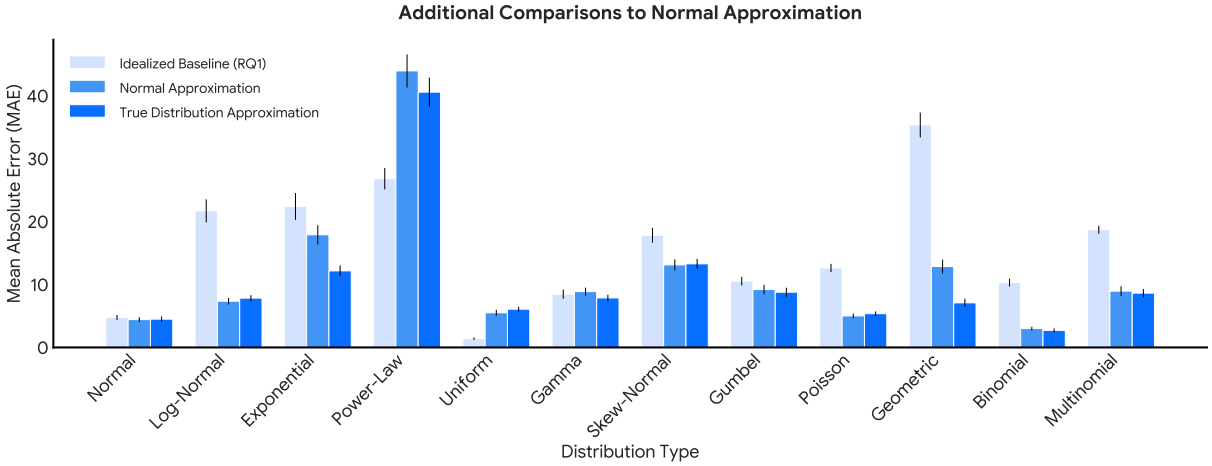

Figure 9: **Additional Normal Approximation Results.** Additional idealized distribution results comparing the normal approximation approach to the baseline corresponding to idealized or real-world distributions and the true distribution approach.

D Additional Experimental Results

D.1 Additional Model Results

Additional model results, extending Table 1 and Table 2, can be found as a part of our GitHub repo here: `https://github.com/yahskapar/LLMs-and-Probabilistic-Reasoning`.

D.2 CoT and Code Tool-use Results

We provide additional zero-shot Chain-of-Thought (CoT) (Kojima et al., 2022) and code tool-use results in Table 3. Though both CoT and code tool-use show benefits over assuming an idealized distribution or adding real-world context (Table 2), neither is convincingly better than the normal approximation approach. The fact that CoT results do not exceed normal approximation suggest that it is non-trivial to instruct a model to improve its modeling of non-normal distributions. Additional improvements to CoT-based approaches could be achieved with further investigation and development of techniques useful for numerical reasoning tasks. Furthermore, though we chose to use a cutoff data for corresponding datasets and did not employ a retrieval approach to retrieve and rank recent information that may be relevant to a proposed probabilistic reasoning question, we acknowledge that a retrieval tool can be useful to get more up-to-date information versus relying on parametric knowledge.

D.3 Additional Normal Approximation Results

In Figure 9, we additionally present results that involve a variant of normal approximation where the true distribution is assumed with a mean and standard deviation as a part of the prompt. In the case of idealized distributions, we utilize the same distribution name with provided mean and standard deviation. In the case of real-world distributions, we approximate the distribution based on the K-S statistic after a matching process across all 12 idealized distributions described in Section 4.

E Broader Impacts

Though we pose more constrained, systematic questions as a part of our investigation of language models and their ability to perform probabilistic reasoning, real-world questions such as "is taking 8000 steps a day normal for an average adult in the U.S. population using wearable devices?" and others that we pose as a part of section 4 are completely reasonable questions for an average user of LMs to ask. There is a significant practical impact in improving the probabilistic reasoning capabilities of language models on real-world distributions, especially when answers to the aforementioned reasonable questions can affect a user's perception of real-world distributions and ultimately their perspective on potentially critical matters such as those in the domains of health, finance, and climate.