Are LLMs Capable of Data-based Statistical and Causal Reasoning? Benchmarking Advanced Quantitative Reasoning with Data

Anonymous Authors
Affiliation
email@example.com

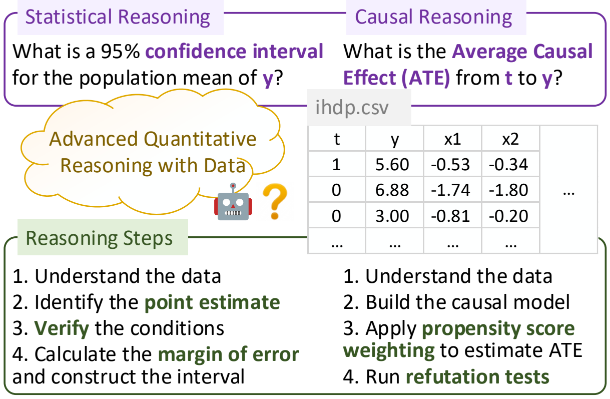
Abstract

Quantitative reasoning is a critical skill to analyze data, yet the assessment of such ability remains limited. To address this gap, we introduce the Quantitative Reasoning with Data (QRDATA) benchmark, aiming to evaluate Large Language Models' capability in statistical and causal reasoning with real-world data. The benchmark comprises a carefully constructed dataset of 411 questions accompanied by data sheets from textbooks, online learning materials, and academic papers. To compare models' quantitative reasoning abilities on data and text, we enrich the benchmark with an auxiliary set of 290 text-only questions, namely QRTEXT. We evaluate natural language reasoning, program-based reasoning, and agent reasoning methods including Chain-of-Thought, Program-of-Thoughts, ReAct, and code interpreter assistants on diverse models. The strongest model GPT-4 achieves an accuracy of 58%, which has much room for improvement. Among open-source models, Deepseek-coder-instruct, a code LLM pretrained on 2T tokens, gets the highest accuracy of 37%. Analysis reveals that models encounter difficulties in data analysis and causal reasoning, and struggle in using causal knowledge and provided data simultaneously.

1 Introduction

Data is everywhere in our daily life and work, from win-loss charts in sports competitions to insurance databases of national health. We rely on our *quantitative reasoning skills* to process the data, more specifically, to extract information from the data, draw inferences, and make predictions [Bueno de Mesquita and Fowler, 2021].

There is evidence that current Large Language Models (LLMs) can perform fundamental data analysis like visualization and summarization [Cheng *et al.*, 2023; Bai *et al.*, 2023]. However, little effort has been made to evaluate the more challenging aspects of quantitative reasoning. On the other hand, there are papers evaluating advanced quantitative reasoning abilities of LLMs [Chen *et al.*, 2023a; Jin *et al.*, 2023a], but they have not investigated whether LLMs can apply these skills to real data.


Statistical Reasoning
What is a 95% confidence interval for the population mean of y?
Causal Reasoning
What is the Average Causal Effect (ATE) from t to y?
Advanced Quantitative Reasoning with Data
ihdp.csv
Reasoning Steps
1. Understand the data
2. Identify the point estimate
3. Verify the conditions
4. Calculate the margin of error and construct the interval
1. Understand the data
2. Build the causal model
3. Apply propensity score weighting to estimate ATE
4. Run refutation tests

t	y	x1	x2	…
1	5.60	-0.53	-0.34	
0	6.88	-1.74	-1.80	
0	3.00	-0.81	-0.20	
…	…	…	…	

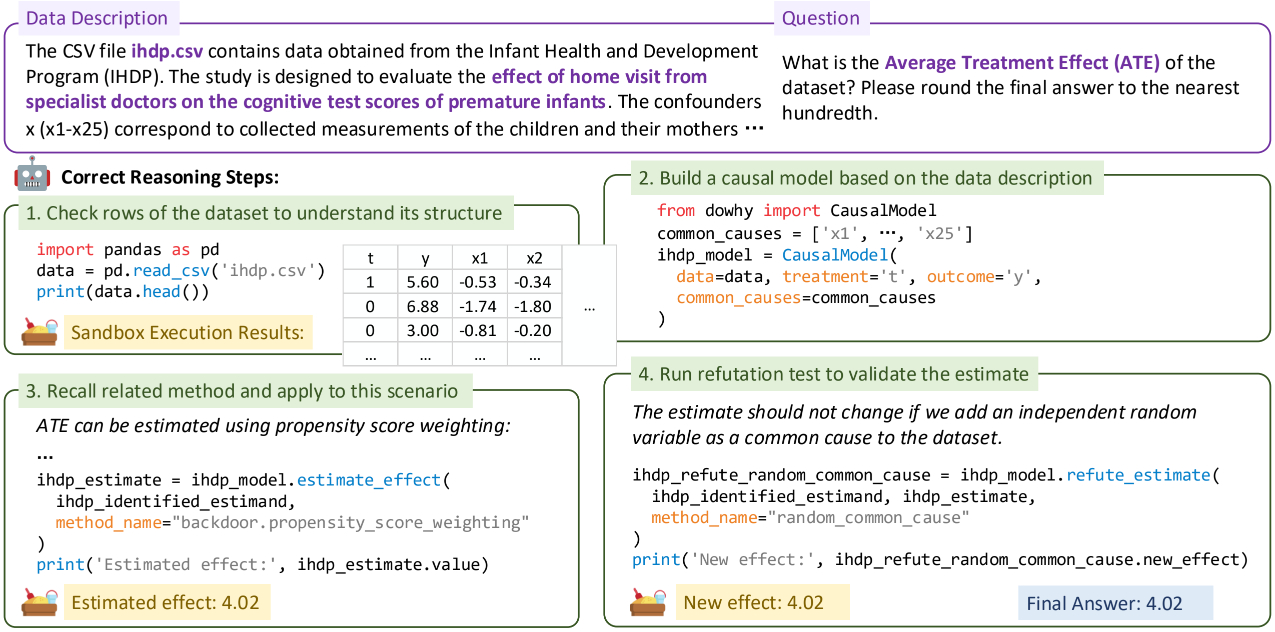
Figure 1: Examples of advanced quantitative reasoning questions and reasoning steps.

In this paper, we focus on two main areas of advanced quantitative reasoning: *statistical reasoning* and *causal reasoning*, with examples shown in Figure 1. Given a data sheet from sample surveys, statistical reasoning aims to infer the underlying probability distribution, addressing questions like *what is a 95% confidence interval for the population mean of* y; while causal reasoning aims to understand the causal relationship between variables, addressing questions like *what is the average treatment effect*[1] *from* t *to* y.

We propose the Quantitative Reasoning with Data (QRDATA) benchmark. It requires models to answer a quantitative reasoning question given data sheets as input. Our benchmark shares a similar form with table question answering [Pasupat and Liang, 2015] and data analysis code generation [Lai *et al.*, 2023] tasks, but has its unique challenges: 1) Our benchmark requires *advanced quantitative knowledge and skills*. Consider the example in Figure 2. To answer the question, models need to know what are common methods of estimating the average treatment effect and how to utilize them. To the best of our knowledge, few existing benchmarks touch upon concepts in statistical reasoning with data,

---

[1] Average treatment effect is a measure of causal effect, quantifying the difference in average outcomes between units assigned to the treatment and units assigned to the control.



Data Description

The CSV file **ihdp.csv** contains data obtained from the Infant Health and Development Program (IHDP). The study is designed to evaluate the **effect of home visit from specialist doctors on the cognitive test scores of premature infants**. The confounders x (x1-x25) correspond to collected measurements of the children and their mothers ⋯

Question

What is the **Average Treatment Effect (ATE)** of the dataset? Please round the final answer to the nearest hundredth.

Correct Reasoning Steps:

1. Check rows of the dataset to understand its structure

```
import pandas as pd
data = pd.read_csv('ihdp.csv')
print(data.head())
```

Sandbox Execution Results:

t	y	x1	x2	
1	5.60	-0.53	-0.34	
0	6.88	-1.74	-1.80	...
0	3.00	-0.81	-0.20	
...	...	...	...	

2. Build a causal model based on the data description

```
from dowhy import CausalModel
common_causes = ['x1', ⋯, 'x25']
ihdp_model = CausalModel(
  data=data, treatment='t', outcome='y',
  common_causes=common_causes
)
```

3. Recall related method and apply to this scenario

ATE can be estimated using propensity score weighting:

...

```
ihdp_estimate = ihdp_model.estimate_effect(
  ihdp_identified_estimand,
  method_name="backdoor.propensity_score_weighting"
)
print('Estimated effect:', ihdp_estimate.value)
```

Estimated effect: 4.02

4. Run refutation test to validate the estimate

The estimate should not change if we add an independent random variable as a common cause to the dataset.

```
ihdp_refute_random_common_cause = ihdp_model.refute_estimate(
  ihdp_identified_estimand, ihdp_estimate,
  method_name="random_common_cause"
)
print('New effect:', ihdp_refute_random_common_cause.new_effect)
```

New effect: 4.02

Final Answer: 4.02

Figure 2: An example question of QRData and correct reasoning steps.

and none of them involve causal reasoning with data. 2) Our benchmark requires *multi-turn* reasoning. After execution of model generated code, models need to read the execution results, and write subsequent code or interpret the results to produce the final answer. In the example, models should first check the data structure before safely indexing the variables in step 2. And after the refutation test in step 4, models need to compare the estimation before and after the refutation, to decide if the estimated effect is reliable, or if another estimation method should be used.

To obtain high-quality quantitative reasoning questions and corresponding data, we construct QRData from open-source textbooks, online learning resources, and academic papers that are accompanied by data. This ensures that all the questions are reasonable and match well with the data. Through careful manual construction, our dataset consists of 411 questions along with 195 data sheets. To analyze the difficulty of data analysis, we also collect an auxiliary dataset of Quantitative Reasoning with Text (QRText). QRText contains 290 questions from textbooks and academic papers, and the questions can be solved based on text descriptions only.

We evaluate a wide range of models and reasoning approaches on QRData. To find what is a good solution to the task, we implement natural language reasoning with Chain-of-Thought [Wei *et al.*, 2022], program-based reasoning with Program-of-Thoughts [Chen *et al.*, 2022], agent reasoning with ReAct [Yao *et al.*, 2022] and the code interpreter assistants of GPT models. We also compare general-purpose LLMs with LLMs finetuned for specific abilities like math, code, and agent reasoning, to find out whether these abilities help in tackling our benchmark. Experimental results show that the best closed-source LLM GPT-4 [OpenAI, 2023] achieves an accuracy of 58% with the code interpreter assistant, and the best open-source LLM Deepseek-coder-instruct [Guo *et al.*, 2024] achieves 37% with Program-of-Thoughts.

As there is much room for improvement in model performance, we further dive into what are the primary difficulties. We find that data analysis is challenging for LLMs, especially for smaller open-source LLMs. And all models including GPT-4 and Gemini [Team *et al.*, 2023] struggle with causal reasoning. They acquire causal knowledge from the pretraining corpus, but can hardly integrate the knowledge with the provided data in reasoning.

Our main contributions are as follows: 1) We propose QRData, the first benchmark for advanced quantitative reasoning with data, to assess models' abilities of data-based statistical and causal reasoning. 2) We evaluate various models and reasoning approaches on QRData, and provide insights into the effective approaches to tackle the benchmark. 3) We analyze the difficulties of current models in data analysis and causal reasoning.

2 Benchmark Construction

In this section, we introduce how the QRData benchmark is constructed. To ensure the quality of our benchmark, we first gather quantitative reasoning teaching and research resources, and then annotate questions based on the materials. To compare the performance of quantitative reasoning on data and text, we also introduce an auxiliary dataset QRText for analysis.

Resource Selection. We look for quantitative reasoning resources with the following criteria: 1) The resource is related to quantitative reasoning, especially statistical reasoning and causal reasoning. 2) It contains multiple quantitative reasoning questions with corresponding data and answers. 3) It is open-source and allows adaptation.

	Count
Data Files	195
Avg. Rows per File	15219
Avg. Columns per File	46
Avg. Words of Data Description	73
Questions	411
Multiple Choice / Numerical	248 / 163
Statistical / Causal	142 / 269
Avg. Words of Question	32

Table 1: Statistics of QRDATA.

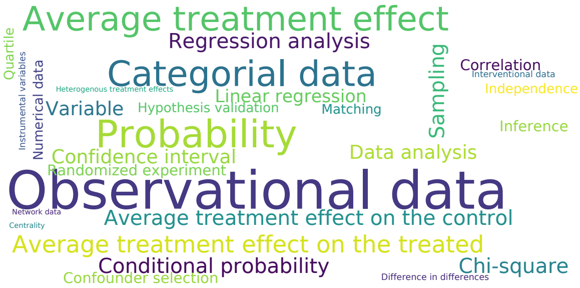


Figure 3: Key concepts in QRDATA.

Eight resources are selected based on these criteria, including two books: OpenIntro Statistics [Diez *et al.*, 2012] and Quantitative Social Science [Imai, 2018]; one online learning resource: Causal Inference for the Brave and True[2]; and five datasets from academic papers: Flow cytometry [Sachs *et al.*, 2005], IHDP [Hill, 2011], Jobs [Shalit *et al.*, 2017], Neuropathic pain diagnosis [Tu *et al.*, 2019], and Arctic sea ice [Huang *et al.*, 2021].

Question Collection. We collect multiple-choice questions and numerical questions from these resources, and ensure that the gold answer is unique. For textbooks and online learning resources, we extract the list of data sheets from their websites. If a data sheet is used in exercises, we use the exercise question and its solution as the question and answer; if it is used in the main content, we annotate questions and answers based on the content.

For existing causal discovery datasets (Flow cytometry, Neuropathic pain diagnosis, and Arctic sea ice), we collect multiple-choice questions like *Which cause-and-effect relationship is more likely? A. headache causes groin pain B. groin pain causes headache C. No causal relationship exists* following the style of Kıcıman *et al.* [2023]. For causal estimation datasets (IHDP and Jobs), we collect numerical questions of estimating *average treatment effect (ATE)*, *average treatment effect on the treated (ATT)*, and *average treatment effect on the control (ATC)* like the example in Figure 2. More details are in Appendix A.1.

We also collect data descriptions like the purpose and format of the data sheets from the resources, and provide them along with questions to models.

Statistics. We provide the dataset statistics in Table 1, and the number of questions collected from each source in Appendix Table 6. The questions we collect cover a wide range of concepts, and we illustrate some of them in Figure 3. We also summarize the topics and question examples in Appendix Table 8. Most questions in QRDATA are based on a single data sheet, while 4% of questions are based on multiple data sheets.

To check whether our dataset has already been memorized by models in training, we apply the pretraining data detection method Min-K% Prob [Shi *et al.*, 2023] on GPT models and Llama-2, and confirm that our dataset is unlikely to appear in their pretraining corpora.

QRTEXT. To separate the challenge of quantitative reasoning from data analysis and analyze whether models master the quantitative reasoning skills, we create an auxiliary benchmark for comparison called Quantitative Reasoning with Text.

Questions in QRTEXT can be answered without data, with an example shown in Appendix Figure 4. We collect questions from OpenIntro Statistics and two text-based causal reasoning datasets CLADDER [Jin *et al.*, 2023a] and CORR2CAUSE [Jin *et al.*, 2023b]. QRTEXT contains 290 questions, 100 for statistical reasoning and 190 for causal reasoning. For ease of model comparison, the ratio of statistical/causal questions of QRTEXT is similar to QRDATA. Details of QRTEXT question collection are in Appendix A.2.

Evaluation Metric. We report the accuracy by comparing the final answer generated by models with the gold answer. As models may encounter measurement error in numerical questions (e.g., different causal effect estimation methods will yield minor differences in results), we consider any result within ±3% of the gold answer to be correct.

3 Methods

We develop several zero-shot reasoning methods as our baselines. By comparing these methods, we aim to find a good solution to the task of quantitative reasoning with data.

Table Question Answering (Table QA). The task of table QA asks models to answer a question based on tables. To evaluate if our benchmark can be solved by existing table QA methods, we experiment with a competitive Table QA method [Zhang *et al.*, 2023]. It constructs a table instruction tuning dataset `TableInstruct`, and finetunes models on this dataset.

Chain-of-Thought (CoT). CoT prompting enables models to conduct complex natural language reasoning through intermediate reasoning steps. We feed the content of data sheets to models and prompt models to `respond to a quantitative question using the provided data ... Let's think step by step.` If the data sheet is too large, we truncate it to fit the length limit. To obtain the final answer from the model output, we employ ChatGPT for answer extraction following Chen *et al.* [2023a].[3]

---

[2] https://matheusfacure.github.io/python-causality-handbook/landing-page.html

[3] To assess the reliability of answer extraction with ChatGPT,

Program-of-Thoughts (PoT). PoT prompting asks models to generate a Python code program, and uses an external computer to execute the code. The output of the code is regarded as the answer. In contrast to TableLlama and CoT prompting, PoT processes the data sheet with code and gets rid of the length limit of data. It also enables models to call data analysis packages for computation.

ReAct-style Prompting. ReAct is a prevalent prompting paradigm for agent reasoning. It combines reasoning and acting in language models for task solving. To address our task, we restrict the action space to Python code execution, and ask models to generate *thoughts* and *actions*. Different from aforementioned methods that only allow one-turn generation, ReAct supports multi-turn reasoning. When a piece of code is generated, we execute it and feed the output back to models as an *observation*. Based on the observation, models will either generate new thoughts and actions, or decide to predict the final answer.

Code Interpreter Assistants. GPT models can be used as agents (called *assistants*), and a code interpreter tool is built-in, which could execute Python code in a sandbox to interpret the data. Similar to ReAct-style prompting, code interpreter assistants allow multi-turn code generation and execution, and end with a text-formed answer.

Details of prompt design and prompt examples are in Appendix A.3.

4 Experiments

4.1 Models

We experiment with the CoT, PoT, and ReAct-style prompting methods on the following models:

General-Purpose LLMs. We select Llama-2-chat, Gemini-Pro, and GPT-4 as representatives of general-purpose LLMs. They exhibit strong abilities to follow instructions [Touvron *et al.*, 2023; Team *et al.*, 2023; OpenAI, 2023].

Code LLMs. CodeLlama-instruct [Roziere *et al.*, 2023] is initialized on Llama-2, finetuned on a code-heavy dataset, and instruction tuned on a code-related instruction dataset. Deepseek-coder-instruct [Guo *et al.*, 2024] is pretrained on a project-level code corpus from scratch and instruction tuned to follow instructions. Both of them perform well among open-source code models on several code benchmarks.

LLM for Mathematical Reasoning. WizardMath [Luo *et al.*, 2023] is finetuned on Llama-2, enhancing the mathematical reasoning abilities with reinforcement learning. It outperforms many larger models on mathematical reasoning benchmarks.

LLM for Agent Reasoning. AgentLM [Zeng *et al.*, 2023] is instruction tuned based on Llama-2 to enhance the agent capabilities. It uses a combination of instructions from general domains and instructions of interaction trajectories.

For a fair comparison, we use the size of 7B for all the open-source models.[4] We use the two latest GPT models GPT-3.5 Turbo and GPT-4[5] for code interpreter assistants. More implementation details are in Appendix A.4.

We select TableLlama [Zhang *et al.*, 2023] as a representative table QA model. It is finetuned on Llama-2-7B [Touvron *et al.*, 2023] with `TableInstruct`, using LongLoRA [Chen *et al.*, 2023b] to extend the context length. It exhibits strong generalization ability across table QA datasets.

As a reference for assessment, we also report the random baseline and the human performance. The random baseline is calculated by randomly choosing from the multiple choices, and all the numerical questions are counted as incorrect.

The human performance is evaluated by asking three graduate students who have taken quantitative reasoning courses to answer a random sample of 50 questions from QRData. Each question is answered by at least two students, and the average accuracy is reported. The students are allowed to use a Python executor, and search for concepts rather than specific questions on the Internet.[6]

4.2 Main Results

To investigate which reasoning methods and models better solve our task, we summarize the performance of different models paired with different reasoning methods in Table 2. GPT-4 with the code interpreter assistant achieves the best performance, and Deepseek-coder-instruct with PoT prompting is the best among open-source models. The best model is 18% worse than human, showing that QRData is challenging for LLMs.

Natural Language Reasoning. CoT works best for several open-source models like Llama-2-chat and WizardMath, but the results are close to or even worse than random, as data-based quantitative reasoning requires heavy calculations that are beyond natural language reasoning. The table QA model TableLlama also underperforms in our benchmark.

Models can hardly answer numerical questions with CoT, with a maximum accuracy of less than 10%. A limitation of CoT and other natural language reasoning methods is that in many cases, we cannot feed the whole data sheets into the input, hindering models to perform accurate calculations. We further compare CoT and PoT prompting on a subset of QRData where the full datasheets can be fed into the base model, and find that CoT does not exceed PoT on this subset. The accuray of CoT prompting on numerical questions is below 20% across all models, suggesting these models struggle with performing precise calculations on data sheets using CoT.

Program-based Reasoning. PoT best suits Deepseek-coder-instruct, but to our surprise, another code LLM CodeLlama-

---

we randomly sample 50 CoT outputs from all models, and manually verify the extracted answers. ChatGPT achieves an accuracy of 98%, demonstrating its proficiency in answer extraction.

[4] The size of Deepseek-coder-instruct is 6.7B.

[5] The specific versions are `gpt-3.5-turbo-1106` and `gpt-4-1106-preview`.

[6] The majority of human errors are in causal discovery questions (71%), because state-of-the-art causal discovery algorithms cannot fully resolve this task [Vowels *et al.*, 2022].

Model	Size	Overall Acc.	Multiple Choice / Numerical	Statistical / Causal
Random		23.0	38.1 / 0.0	15.0 / 27.2
Table Question Answering				
TableLlama [Zhang *et al.*, 2023]	7B	9.5	14.9 / 1.2	3.5 / 12.6
Chain of Thought Prompting				
Llama-2-chat [Touvron *et al.*, 2023]	7B	22.1	35.1 / 2.5	20.4 / 23.0
AgentLM [Zeng *et al.*, 2023]	7B	24.3	40.0 / 0.0	19.0 / 27.1
WizardMath [Luo *et al.*, 2023]	7B	25.1	40.0 / 2.5	18.3 / 28.6
CodeLlama-instruct [Roziere *et al.*, 2023]	7B	19.2	31.0 / 1.2	14.1 / 21.9
Deepseek-coder-instruct [Guo *et al.*, 2024]	6.7B	18.7	30.2 / 1.2	15.5 / 20.4
Gemini-Pro [Team *et al.*, 2023]	-	31.9	46.8 / 9.2	25.4 / 35.3
GPT-4 [OpenAI, 2023]	-	36.5	57.3 / 4.9	24.6 / 42.8
Program of Thoughts Prompting				
Llama-2-chat	7B	3.7	6.1 / 0.0	7.8 / 1.5
AgentLM	7B	3.2	4.4 / 1.2	8.5 / 0.4
WizardMath	7B	16.8	16.5 / 17.2	31.7 / 8.9
CodeLlama-instruct	7B	17.3	24.2 / 6.7	19.7 / 16.0
Deepseek-coder-instruct	6.7B	37.0	40.7 / 31.3	45.8 / 32.3
Gemini-Pro	-	26.8	27.8 / 25.2	39.4 / 20.1
GPT-4	-	49.1	46.0 / 54.0	72.5 / 36.8
ReAct-style Prompting				
Llama-2-chat	7B	11.4	18.5 / 0.6	4.2 / 15.2
AgentLM	7B	10.7	16.9 / 1.2	5.6 / 13.4
WizardMath	7B	18.7	21.0 / 15.3	19.7 / 18.2
CodeLlama-instruct	7B	15.8	22.2 / 6.1	16.2 / 15.6
Deepseek-coder-instruct	6.7B	27.3	29.4 / 23.9	38.7 / 21.2
Gemini-Pro	-	38.9	47.6 / 25.8	41.5 / 37.5
GPT-4	-	57.4	**59.2** / 54.6	69.0 / **51.3**
Code Interpreter Assistants				
GPT-3.5 Turbo	-	41.4	42.3 / 39.9	60.6 / 31.2
GPT-4	-	**57.9**	56.9 / **59.5**	**78.9** / 46.8
Human (subset)		76.0	71.9 / 83.4	88.8 / 68.8

Table 2: Performance of models on QRDATA. Numbers are accuracies in percentages (%). The best results are in bold. For models evaluated with multiple reasoning methods, the model-level best results are underlined.

instruct does not perform well with PoT. The gap in coding abilities can also be seen from the execution success rate, which is 89% for Deepseek-coder-instruct and 73% for CodeLlama-instruct.

Although the overall performance of several non-code LLMs drops with PoT compared to CoT, the accuracy of numerical questions improves with the help of code execution. This exhibits the potential for performance improvement through the hybrid of CoT and PoT rationales [Yue *et al.*, 2023].

Agent Reasoning. Closed-source LLMs like GPT-4 and Gemini-Pro perform best with agent reasoning. GPT-4 performance with ReAct-style prompting is close to with the code interpreter assistant, exhibiting that our ReAct-style prompts successfully activate the model abilities.

Smaller open-source models can hardly follow the complex ReAct-style instructions, which limits their performance. Although AgentLM is finetuned for agent reasoning, it cannot generalize well to the quantitative reasoning task. On the other hand, open-source models except code LLMs perform better with ReAct-style prompting than PoT, because multi-turn code execution gives them chances to fix the generated code, rising the average execution success rate from 25.5% to 35.6%. We also manually review 50 randomly sampled ReAct outputs of these models, and find that in 10% of the cases, the initially generated code is non-executable, and is fixed in the following steps. This suggests that the iterative, multi-turn nature of agent reasoning can contribute to fixing the generated code.

LLMs with Enhanced Abilities. We compare three models finetuned on Llama-2 for enhanced abilities of code, math, and agent reasoning with the general-purpose LLM Llama-2-chat. AgentLM and WizardMath only gain marginal improvement, and CodeLlama-instruct even underperforms Llama-2-chat. This may be due to the limited capability of the base model.

On the other hand, Deepseek-coder-instruct, which masters in code generation, outperforms all other models with similar sizes by more than 10%. We further evaluate Deepseek-coder-instruct-33B to see if it could approach the performance of closed-source LLMs, but it gets an overall accuracy of 31.9% with PoT and 27.0% with ReAct-style prompting, even lower than the 6.7B model. Although coding ability is important in solving quantitative questions with data, the performance of code LLMs is still limited compared to larger general-purpose LLMs like GPT-4.

Statistical Reasoning vs. Causal Reasoning. Statistical reasoning focuses on identifying patterns in data, whereas causal reasoning goes a step further to find causal relationships be-

Model	Overall Acc.	Statistical / Causal
Random	23.7	20.8 / 25.3
Chain of Thought Prompting		
Llama-2-chat	28.6	33.0 / 26.3
AgentLM	26.6	28.0 / 25.8
WizardMath	36.6	46.0 / 31.6
CodeLlama-i	26.2	26.0 / 26.3
Deepseek-coder-i	23.1	18.0 / 25.8
Gemini-Pro	37.9	51.0 / 31.1
GPT-4	52.1	76.0 / 39.5
Program of Thoughts Prompting		
Llama-2-chat	21.7	12.0 / 26.8
AgentLM	22.4	16.0 / 25.8
WizardMath	34.1	38.0 / 32.1
CodeLlama-i	27.2	26.0 / 27.9
Deepseek-coder-i	39.0	47.0 / 34.7
Gemini-Pro	43.1	56.0 / 36.3
GPT-4	59.0	86.0 / **44.7**
ReAct-style Prompting		
Llama-2-chat	3.4	9.0 / 0.5
AgentLM	11.0	5.0 / 14.2
WizardMath	19.7	28.0 / 15.3
CodeLlama-i	4.8	13.0 / 0.5
Deepseek-coder-i	29.0	42.0 / 22.1
Gemini-Pro	35.2	52.0 / 26.3
GPT-4	57.2	86.0 / 42.1
Code Interpreter Assistants		
GPT-3.5 Turbo	47.2	65.0 / 37.9
GPT-4	**59.7**	**89.0** / 44.2

Table 3: Performance of models on QRText. Numbers are accuracies in percentages (%). Best results are in bold, and model-level best results are underlined. The suffix -i indicates -instruct.

yond correlation. For all models evaluated, the gap between model performance and the random baseline is larger for statistical questions than causal questions, showing that models obtain better statistical reasoning abilities than causal reasoning abilities. Even the best-performed model GPT-4 only answers 51% of causal questions correctly, demonstrating the large room for improvement in causal reasoning.

5 Analysis

We go deeper into the primary difficulties models face in addressing the task of quantitative reasoning with data. This may provide insights into how to design methods to better solve our task. We also provide an error analysis of GPT-4 in Appendix A.6.

5.1 Difficulty of Data Analysis

We evaluate models[7] on our auxiliary benchmark QRText, and compare the performance on QRText and QRData to quantify the difficulty of data analysis. As shown in Table 3, GPT-4 achieves the best accuracy of 59.7% on QRText, and Deepseek-coder-instruct gets the best accuracy of 39.0% among open-source LLMs.

The single-turn reasoning methods, CoT and PoT prompting, work best for all models except GPT-4 on QRText. For GPT-4, the performance gap between the code interpreter assistant and PoT is merely 0.7%, compared to 8.8% on QRData. These demonstrate that single-turn reasoning is almost adequate for quantitative reasoning with text, while multi-turn reasoning is more necessary for quantitative reasoning with data.

Model	Acc. w/ data	Acc. w/o data
Random	35.7	35.7
GPT-3.5 Turbo	**35.1**	25.3
Gemini-Pro	45.4	**48.8**
GPT-4	50.7	**60.0**

Table 4: Performance of models on multiple-choice causal questions of QRData with (w/) and without (w/o) access to the data. Better results are in bold.

All models perform better on QRText than QRData from 1.8% to 11.5%, and the gap is larger for smaller models. If we control the knowledge and skills required by restricting questions to the same source (OpenIntro Statistics), models perform 6% better on average. These provide evidence that most models have difficulty in data analysis.

5.2 Difficulty of Causal Reasoning

We observe the performance gap between statistical reasoning and causal reasoning on QRData, and the gap remains on QRText. GPT-4 achieves 89% accuracy in statistical questions on QRText, but only about half the accuracy in causal questions. This exhibits the unique difficulty of causal reasoning regardless of data analysis.

Table 5 provides a case where GPT-4 with the code interpreter assistant mis-predicts a causal question from QRData. When asked to predict the causal relation between two variables *L L1 radiculopathy* and *R L5 radiculopathy*, GPT-4 proposes a wrong plan of calculating the conditional probabilities in Step 2. As correlation does not imply causation, although GPT-4 successfully executes the plan, it makes a wrong prediction.

Easier Said than Done. The unsatisfactory performance on causal questions appears to differ from the findings of some previous papers [Willig *et al.*, 2022; Kıcıman *et al.*, 2023], which show that LLMs are capable of causal reasoning in tasks like causal discovery *without data*. To figure out the reason for the inconsistency, we conduct an experiment to evaluate if the provided data interfered with the models. Specifically, we evaluate the best three models on multiple-choice causal questions of QRData *without giving them the data*[8], and compare their performance with the original setting. We use the best reasoning method for each model: ReAct-style prompting for Gemini-Pro, and code interpreter assistants for GPT-3.5 Turbo and GPT-4.

As shown in Table 4, two out of three models perform better without data access, revealing that it is easier for these LLMs to speculate causal relations than to conduct causal

---

[7] We do not experiment with TableLlama as the questions do not involve tables.

[8] Note that this setting is different with QRText. Without data access, questions of QRData are not self-contained, but rely on models' commonsense to predict answers.

Question	Which cause-and-effect relationship is more likely? Please answer with A, B, or C. A. R L5 radiculopathy causes L L1 radiculopathy B. L L1 radiculopathy causes R L5 radiculopathy C. No causal relationship exists
Gold Answer	C
GPT-4 Prediction with Data Access	

Step 1:
Code Interpreter
	`# Display the first few records to understand the data structure`
	`data.head()`

Step 2:	I will examine the data for the following patterns: - If patients with “R L5 radiculopathy” also frequently have “L L1 radiculopathy” and not the other way around, it may suggest that A is more likely. - Conversely, if patients with “L L1 radiculopathy” also frequently have “R L5 radiculopathy” and not the other way around, it may suggest that B is more likely. - If there is no clear pattern or both conditions occur independently, it may suggest that C is more likely.

Step 3:
Code Interpreter
	`# Probability of L L1 given R L5`
	`P_L_L1_given_R_L5 = co_occurrence_count / R_L5_count if R_L5_count else 0`
	`# Probability of R L5 given L L1`
	`P_R_L5_given_L_L1 = co_occurrence_count / L_L1_count if L_L1_count else 0`
	`P_L_L1_given_R_L5, P_R_L5_given_L_L1`

Step 4:	Based on these conditional probabilities, it is more likely that “L L1 radiculopathy” causes ‘R L5 radiculopathy” over the other proposed causal direction. Final answer: B.
GPT-4 Prediction without Data Access	
Prediction	Given both ‘R L5’ and ‘L L1’ radiculopathies are on different sides of the body and different spinal levels, it’s not common for one to cause the other according to standard medical understanding. It is more likely that they occur independently due to separate causes. It’s likely that if both conditions are present in a patient, they arose independently or due to a systemic issue affecting the spine broadly, rather than one causing the other. Final answer: C.

Table 5: An example of GPT-4 prediction on QRData. GPT-4 correctly predicts the answer without data access, but predicts wrongly when the data is provided. The outputs are abridged and the full outputs are in Appendix A.5.

reasoning with data. The strong causal reasoning abilities exhibited by models in previous works come from their internal knowledge, but models can hardly use their internal knowledge to analyze the data provided.

In the case of Table 5, in the absence of data access, GPT-4 correctly predicts that there is no causal relationship between the two variables. However, when the data is provided, it is misled to calculate the conditional probabilities.

6 Related Work

6.1 Data-related Benchmarks

Table Question Answering. Table QA [Pasupat and Liang, 2015; Zhong *et al.*, 2017] asks models to answer questions about information from given tables. It evaluates models' abilities to parse queries, look up information from tables, and integrate the retrieved information to reach an answer. Datasets like Spider [Yu *et al.*, 2018] and SEDE [Hazoom *et al.*, 2021] focus more on the semantic parsing part and design complex questions that need compositional generalization over SQL queries. Some datasets present the challenges of retrieving information from hierarchical tables [Cheng *et al.*, 2022; Zhao *et al.*, 2022] or a fusion of textual and tabular data [Chen *et al.*, 2020; Zhu *et al.*, 2021]. Datasets like FinQA [Chen *et al.*, 2021] and TabMWP [Lu *et al.*, 2022b] require numerical reasoning to figure out the answer, but they mainly focus on basic arithmetic, while our benchmark requires more advanced quantitative reasoning skills.

Furthermore, although some benchmarks require multi-hop reasoning, they can be solved in a single turn, whereas our benchmark needs multi-turn reasoning between the model and data.

Code Generation. Code generation tasks require models to produce code to meet certain intents. Recent benchmarks like DS-1000 [Lai *et al.*, 2023] and ExeDS [Huang *et al.*, 2022] explore LLMs' abilities to generate data science code, like training models using machine learning packages and making visualizations. KnowledgeMath [Zhao *et al.*, 2023] and BizBench [Koncel-Kedziorski *et al.*, 2023] evaluate if models could generate code for financial reasoning with tables. To our knowledge, none of them focus on advanced quantitative reasoning. Only a concurrent dataset DAEval [Hu *et al.*, 2024] involves correlation analysis as a component of question concepts, and none of them include causal reasoning questions.

6.2 Quantitative Reasoning Abilities

There is a growing interest in exploring language models' abilities of quantitative reasoning. Various benchmarks focus on mathematical reasoning [Hendrycks *et al.*, 2021; Cobbe *et al.*, 2021], and recent works broaden the analysis to the field of science [Lu *et al.*, 2022a; Chen *et al.*, 2023a].

Another line of work focuses on the specific aspect of causal reasoning of LLMs. Bhagavatula *et al.* [2019]; Qin *et al.* [2019]; Liu *et al.* [2023] explore the ability of commonsense causal reasoning like abductive reasoning and counterfactual reasoning. Willig *et al.* [2022]; Gao *et al.* [2023]; Kıcıman *et al.* [2023] analyze the mastery of causal knowledge like cause-effect pairs. Jin *et al.* [2023a,b] evaluate the utilization of causal inference skills.

In this work, we extend the research to quantitative reasoning *with data*, which poses new challenges for LLMs.

7 Conclusion

We propose QRDATA, a benchmark of advanced quantitative reasoning with data, to explore LLMs' statistical and causal reasoning abilities with real-world data. We carefully construct the benchmark with questions from teaching and research materials, and compare natural language reasoning, program-based reasoning, and agent reasoning among a wide range of models. Experimental results reveal that models have difficulty in *data analysis* and *causal reasoning*. Even though powerful LLMs like GPT-4 have acquired causal knowledge, they can hardly integrate them with the provided data. By proposing this benchmark, we hope to elicit attention to the data-based quantitative reasoning ability. We call for more specialized methods to enhance the ability, like training models in the agent reasoning style with data as input, and equipping models with more quantitative reasoning knowledge.

A Appendix

A.1 Question Collection of QRDATA

We provide the number of questions collected from different sources in Table 6. Table 8 summarizes the topics covered in QRDATA, and examples of questions related to these topics.

Causal Discovery. We collect both pairwise causal discovery and full graph causal discovery questions from the Neuropathic pain diagnosis dataset following Kıcıman *et al.* [2023].

We sample 50 causal relations like *L1 radiculopathy* $\rightarrow$ *IBS* from the gold causal graph, and generate pairwise causal discovery questions asking models to predict the direction of the causal relation. The questions are in the form of *Which cause-and-effect relationship is more likely? A. R L1 radiculopathy causes IBS B. IBS causes R L1 radiculopathy*.

In full graph causal discovery, there are three possible relations between two variables A and B: $A \rightarrow B$, $B \rightarrow A$, and no causal relation exists. We sample 25 variable pairs that have causal relations in the causal graph and 25 variable pairs that do not have causal relations, and form 50 questions in the form of *Which cause-and-effect relationship is more likely? A. headache causes groin pain B. groin pain causes headache C. No causal relationship exists*.

For the Flow cytometry and Arctic sea ice datasets, we only collect the more challenging full graph causal discovery questions. As the causal graph of Arctic sea ice contains double-sided edges, we ask models to predict from four possible choices like *Which cause-and-effect relationship is more likely? A. Residual_cloud_cover causes Residual_longwave B. Residual_longwave causes Residual_cloud_cover C. The causal relation is double sided between Residual_cloud_cover and Residual_longwave D. No causal relationship exists*.

Source	Question
OpenIntro Statistics	112
Quantitative Social Science	35
Causal Inference for the Brave and True	24
Flow Cytometry	50
IHDP	30
Jobs	10
Neuropathic Pain Diagnosis	100
Arctic Sea Ice	50
Overall	411

Table 6: Count of questions from different sources of QRDATA.

Failure Reason	Percentage (%)
Data Analysis	34%
Knowledge Recall	44%
Equation Formulation	16%
Calculation	6%

Table 7: Error analysis of GPT-4 with the code interpreter assistant on QRDATA.

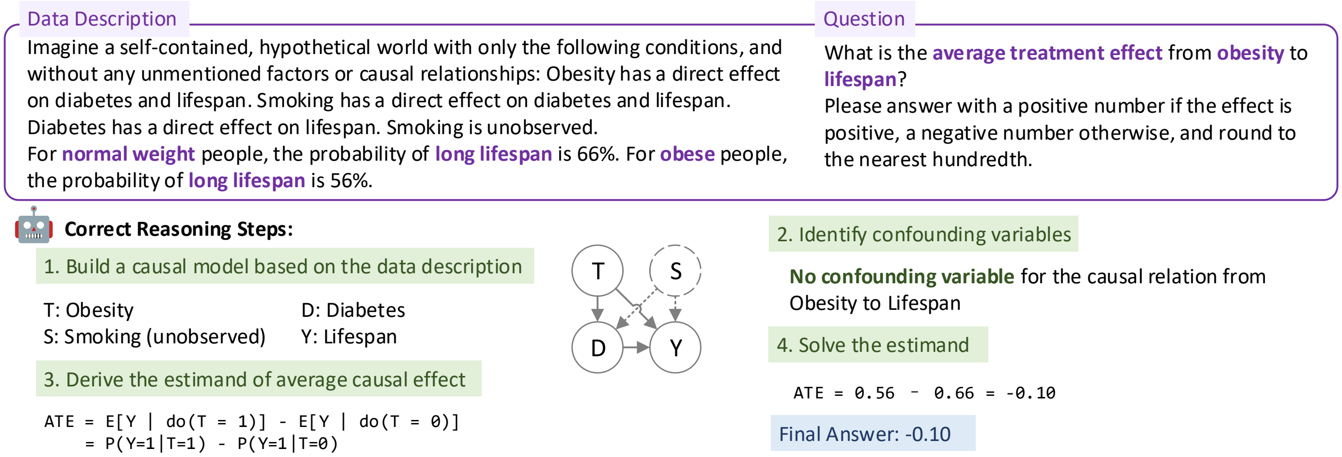
Causal Estimation. We collect numerical questions of estimating *average treatment effect (ATE)*, *average treatment effect on the treated (ATT)*, and *average treatment effect on the control (ATC)* from the IHDP dataset. As all the outcomes and counterfactual outcomes are simulated using real covariates, we calculate the values of these estimates as the gold answers. In the Jobs dataset, all the treatments were from randomized experiments, so we can calculate the true ATT values. We collect questions of estimating ATT from the dataset.

A.2 Question Collection of QRTEXT

QRTEXT contains 290 questions in total, 140 multiple choice questions and 150 numerical questions. We collect 100 statistical reasoning questions from OpenIntro Statistics, 94 causal reasoning questions from CLADDER, and 96 causal reasoning questions from CORR2CAUSE. Figure 4 exhibits an example question of QRTEXT.

The OpenIntro Statistics questions are collected from exercises with solutions from the textbook. CORR2CAUSE contains causal discovery questions of determining the causal relationship between the variables given correlational statements. We randomly sample CORR2CAUSE questions from the original test set.

CLADDER contains causal questions where the conditions are described in natural language. It classifies questions into three rungs: associational (rung 1), interventional (rung 2), and counterfactual (rung 3). We only consider questions of rungs 2 and 3 because these are related to causal reasoning. We randomly sample causal estimation questions querying Average Treatment Effect (ATE), Effect of the Treatment on the Treated (ETT), Natural Direct Effect (NDE), and Natural Indirect Effect (NIE), and transform the original yes-or-no questions into numerical questions using the provided metadata. This is to balance the ratio of multiple choice questions and numerical questions, controlling the random performance



Data Description

Imagine a self-contained, hypothetical world with only the following conditions, and without any unmentioned factors or causal relationships: Obesity has a direct effect on diabetes and lifespan. Smoking has a direct effect on diabetes and lifespan. Diabetes has a direct effect on lifespan. Smoking is unobserved.
For **normal weight** people, the probability of **long lifespan** is 66%. For **obese** people, the probability of **long lifespan** is 56%.

Question

What is the **average treatment effect** from **obesity** to **lifespan**?
Please answer with a positive number if the effect is positive, a negative number otherwise, and round to the nearest hundredth.

Correct Reasoning Steps:

1. Build a causal model based on the data description

T: Obesity
D: Diabetes
S: Smoking (unobserved)
Y: Lifespan

2. Identify confounding variables

No confounding variable for the causal relation from Obesity to Lifespan

3. Derive the estimand of average causal effect

```
ATE = E[Y | do(T = 1)] - E[Y | do(T = 0)]
    = P(Y=1|T=1) - P(Y=1|T=0)
```

4. Solve the estimand

```
ATE = 0.56 – 0.66 = -0.10
```

Final Answer: -0.10

Figure 4: An example question of QRTEXT and correct reasoning steps. The question is converted from CLADDER.

to be close to QRDATA.

A.3 Prompt Design

Table 9 shows an example prompt for Chain-of-Thought reasoning. In real prompts, we feed all the content of the data sheets until it meets the length limit of 3500 tokens. We shuffle the data to avoid the bias towards values in the front. After Chain-of-Thought reasoning, we use GPT-3.5 Turbo for answer extraction, and the prompt template is shown in Table 10.

Table 11 shows an example prompt for Program-of-Thoughts reasoning. We shuffle the data and provide the first ten lines in the prompt. An example prompt for ReAct-style reasoning is in Table 12, and an example prompt for code interpreter assistants is in Table 13. When the question is based on multiple data sheets, we demonstrate the rows of these data sheets respectively in the prompts.

A.4 Implementation Details

We conduct model inference with temperature 0 and maximum new tokens 512. We limit the maximum reasoning steps to 10 in ReAct-style prompting. Experiments are conducted on NVIDIA A800 GPUs. It takes five minutes to evaluate a 7B model on QRDATA with CoT/PoT prompting on a single GPU, and one hour to evaluate a 7B model with ReAct-style prompting.

A.5 Case Study

Table 14 demonstrates the full output of GPT-4 on the example case from QRDATA, and Table 15 demonstrates the full output of GPT-4 without access to the data.

A.6 Error Analysis

To have a quantitative view of model failures, we conduct an error analysis for the best-performed model GPT-4 with the code interpreter assistant on QRDATA. We randomly sample 50 incorrect predicted instances, and annotate their main reason for mis-prediction. As shown in Table 7, 34% of the mis-predictions originate from data analysis. The model may fail to retrieve the correct columns according to the instruction of questions, or misinterpret the data values. 44% of the errors occur in knowledge recall, like choosing a wrong hypothesis test, or using statistical methods to address a causal question. Even if the model recalls the correct knowledge to solve the question, it formulates wrong equations in 16% of the cases. And 6% of errors occur in calculation. This is also the main reason for errors when humans conduct the task, because state-of-the-art causal methods cannot perfectly reach the correct answer.

Ethics Statement

Intellectual Property. We follow the permission of licenses during the data collection process. And the collected data should not be used commercially.

Potential Risks. Our evaluation reveals that current LLMs are imperfect in conducting quantitative reasoning with data. We suggest readers not trust solutions provided by LLMs.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. Abductive commonsense reasoning. In *International Conference on Learning Representations*, 2019.

Ethan Bueno de Mesquita and Anthony Fowler. *Thinking clearly with data: A guide to quantitative reasoning and analysis*. Princeton University Press, 2021.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online, November 2020. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

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

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. TheoremQA: A theorem-driven question answering dataset. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7889–7901, Singapore, December 2023. Association for Computational Linguistics.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. Longlora: Efficient fine-tuning of long-context large language models. *arXiv preprint arXiv:2309.12307*, 2023.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. HiTab: A hierarchical table dataset for question answering and natural language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1094–1110, Dublin, Ireland, May 2022. Association for Computational Linguistics.

Liying Cheng, Xingxuan Li, and Lidong Bing. Is gpt-4 a good data analyst? *arXiv preprint arXiv:2305.15038*, 2023.

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

David M Diez, Christopher D Barr, and Mine Cetinkaya-Rundel. *OpenIntro statistics*. OpenIntro Boston, MA, USA:, 2012.

Jinglong Gao, Xiao Ding, Bing Qin, and Ting Liu. Is ChatGPT a good causal reasoner? a comprehensive evaluation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11111–11126, Singapore, December 2023. Association for Computational Linguistics.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y Wu, YK Li, et al. Deepseek-coder: When the large language model meets programming–the rise of code intelligence. *arXiv preprint arXiv:2401.14196*, 2024.

Moshe Hazoom, Vibhor Malik, and Ben Bogin. Text-to-SQL in the wild: A naturally-occurring dataset based on stack exchange data. In *Proceedings of the 1st Workshop on Natural Language Processing for Programming (NLP4Prog 2021)*, pages 77–87, Online, August 2021. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

Xueyu Hu, Ziyu Zhao, Shuang Wei, Ziwei Chai, Guoyin Wang, Xuwu Wang, Jing Su, Jingjing Xu, Ming Zhu, Yao Cheng, et al. Infiagent-dabench: Evaluating agents on data analysis tasks. *arXiv preprint arXiv:2401.05507*, 2024.

Yiyi Huang, Matthäus Kleindessner, Alexey Munishkin, Debvrat Varshney, Pei Guo, and Jianwu Wang. Benchmarking of data-driven causality discovery approaches in the interactions of arctic sea ice and atmosphere. *Frontiers in big Data*, 4:642182, 2021.

Junjie Huang, Chenglong Wang, Jipeng Zhang, Cong Yan, Haotian Cui, Jeevana Priya Inala, Colin Clement, and Nan Duan. Execution-based evaluation for data science code generation models. In *Proceedings of the Fourth Workshop on Data Science with Human-in-the-Loop (Language Advances)*, pages 28–36, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics.

Kosuke Imai. *Quantitative social science: an introduction*. Princeton University Press, 2018.

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, LYU Zhiheng, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, et al. Cladder: A benchmark to assess causal reasoning capabilities of language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Zhijing Jin, Jiarui Liu, Zhiheng Lyu, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona Diab, and Bernhard Schölkopf. Can large language models infer causation from correlation? *arXiv preprint arXiv:2306.05836*, 2023.

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. *arXiv preprint arXiv:2305.00050*, 2023.

Rik Koncel-Kedziorski, Michael Krumdick, Viet Lai, Varshini Reddy, Charles Lovering, and Chris Tanner. Bizbench: A quantitative reasoning benchmark for business and finance. *arXiv preprint arXiv:2311.06602*, 2023.

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Wen-tau Yih, Daniel Fried, Sida

Wang, and Tao Yu. Ds-1000: A natural and reliable benchmark for data science code generation. In *International Conference on Machine Learning*, pages 18319–18345. PMLR, 2023.

Xiao Liu, Da Yin, Chen Zhang, Yansong Feng, and Dongyan Zhao. The magic of IF: Investigating causal reasoning abilities in large language models of code. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9009–9022, Toronto, Canada, July 2023. Association for Computational Linguistics.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *The Eleventh International Conference on Learning Representations*, 2022.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*, 2023.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China, July 2015. Association for Computational Linguistics.

Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. Counterfactual story reasoning and generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5043–5053, Hong Kong, China, November 2019. Association for Computational Linguistics.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International conference on machine learning*, pages 3076–3085. PMLR, 2017.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *NeurIPS 2023 Workshop on Regulatable ML*, 2023.

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

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

Ruibo Tu, Kun Zhang, Bo Bertilson, Hedvig Kjellstrom, and Cheng Zhang. Neuropathic pain diagnosis simulator for causal discovery algorithm evaluation. *Advances in Neural Information Processing Systems*, 32, 2019.

Matthew J Vowels, Necati Cihan Camgoz, and Richard Bowden. D'ya like dags? a survey on structure learning and causal discovery. *ACM Computing Surveys*, 55(4):1–36, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Moritz Willig, Matej Zečević, Devendra Singh Dhami, and Kristian Kersting. Can foundation models talk causality? In *UAI 2022 Workshop on Causal Representation Learning*, 2022.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2022.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*, 2023.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. Agenttuning: Enabling generalized agent abilities for llms. *arXiv preprint arXiv:2310.12823*, 2023.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. Tablellama: Towards open large generalist models for tables. *arXiv preprint arXiv:2311.09206*, 2023.

Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. MultiHiertt: Numerical reasoning over multi hierarchical tabular and textual data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6588–6600, Dublin, Ireland, May 2022. Association for Computational Linguistics.

Yilun Zhao, Hongjun Liu, Yitao Long, Rui Zhang, Chen Zhao, and Arman Cohan. Knowledgemath: Knowledge-intensive math word problem solving in finance domains. *arXiv preprint arXiv:2311.09797*, 2023.

Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*, 2017.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online, August 2021. Association for Computational Linguistics.

Topic	**Description**	**Example**
Probability	Calculating the likelihood of events happening.	What is the probability a randomly selected person who was not inoculated died from smallpox?
Distribution	Understanding how data points are spread out.	Is it reasonable to model the proportion of US adults who support expanding wind turbines using a normal distribution?
Estimation	Approximating parameters based on samples drawn from the population, and quantifying the uncertainty of estimates.	Use a Poisson distribution to approximate the data. What is the event rate of the Poisson distribution?
Hypothesis Testing	Using statistical tests to assess the validity of hypotheses.	Given the hypothesis that the proportion of college graduates who support off-shore drilling in California is equal to that of non-college graduates, will you accept or reject the hypothesis with a significance level of 5%?
Prediction	Using statistical models to predict future values based on existing data trends.	Fit a linear regression model using the Democratic margin in the two-party vote share as the response variable and the perceived competence for Democratic candidates as the predictor. What is the estimated slope of the model?

(a) Topics of statistical reasoning.

Topic	**Description**	**Example**
Confounding	Recognizing confounders that might influence the observed relationship between two variables. For example, a variable that causes both the treatment and the outcome is a confounder.	We are estimating the causal effect of the new drug on the number of days the patient stays in the hospital, and we already controlled for the severity. Should we also control for the 'hospital' variable?
Causal Discovery	Inferring the causal relations from the data. For example, distinguishing between correlation and causation and determining the causal direction between two variables.	Which cause-and-effect relationship is more likely? A. R L5 radiculopathy causes L L1 radiculopathy B. L L1 radiculopathy causes R L5 radiculopathy C. No causal relationship exists
Causal Effect Estimation	Estimating the causal effect between two variables using methods like propensity score matching and doubly robust estimation.	What is the average treatment effect of the growth mindset on the achievement score? Please conduct a doubly robust estimation and use logistic regression to estimate the propensity score.
Instrumental Variables	Addressing confounding bias with instrumental variables. An instrumental variable causes the treatment and is only correlated with the outcome through the treatment.	What is the average additional percentage wage increase associated with each additional year of education based on the instrumental variable of whether the individual is born in the last quarter (q4)?
Panel Data	Applying methods like difference-in-differences and synthetic control to estimate the causal effect in panel data. These methods differ from those used to estimate causal effects in a single period.	What is the average increase in bank deposits per customer in Porto Alegre after the billboard intervention, as estimated by the difference-in-differences approach?

(b) Topics of causal reasoning.

Table 8: Topics covered in QRDATA and examples of questions related to these topics.

Data Description:

Public health has improved and evolved, but has the public's knowledge changed with it? This data set rosling_responses.csv contains sample responses for two survey questions posed by Hans Rosling during lectures to a wide array of college-educated audiences.

rosling_responses:
— — question — response — prob_random_correct —
— 0 — children_in_2100 — correct — 0.333333 —
— 1 — children_in_2100 — incorrect — 0.333333 —
— 2 — children_in_2100 — correct — 0.333333 —
— 3 — children_with_1_or_more_vaccination — incorrect — 0.333333 —
— 4 — children_in_2100 — incorrect — 0.333333 —
...

Task:

You are a data analyst and good at quantitative reasoning. You are required to respond to a quantitative question using the provided data. The description and the table are listed above. Please analyze the table to answer the question. Do not write any code in your answer. Ensure that your final answer is positioned at the very end of your output, adhering to the format 'Final answer: [answer]'. The final answer should be a number or a short phrase and should be written in a new line.

Question:

Can we reject the notion that the performance of college-educated adults was different than random guessing in the children_with_1_or_more_vaccination question with 95% confidence? Please answer with "yes" or "no".

Response:

Let's think step by step.

Table 9: Example prompt for Chain-of-Thought reasoning. In real prompts, we feed all the content of the data sheets until it meets the length limit.

Extract the final answer from the given solution as a numeric value or a short phrase for the question. If you cannot extract an answer, return ”None”.

You should either return ”None” or the final answer without any additional words.

Question: {question}

Solution: {answer}

Final Answer:

Table 10: Prompt template for answer extraction after Chain-of-Thought reasoning. {question} is the question from the benchmark, and {answer} is the answer generated by models.

You are a data analyst and good at quantitative reasoning. You are required to respond to a quantitative question using the provided data. The description and the question can be found below. Please analyze the first 10 rows of the table and write python code to analyze the whole table. You can use any python library. The returned value of the program is supposed to be the answer. The format of the code should be

```python
def solution():
    # import libraries if needed
    # load data
    # write code to get the answer
    # return answer
```

Data Description:

Public health has improved and evolved, but has the public's knowledge changed with it? This data set rosling_responses.csv contains sample responses for two survey questions posed by Hans Rosling during lectures to a wide array of college-educated audiences.

First 10 rows of the data:

rosling_responses:
— — question — response — prob_random_correct —
— 0 — children_in_2100 — correct — 0.333333 —
— 1 — children_in_2100 — incorrect — 0.333333 —
— 2 — children_in_2100 — correct — 0.333333 —
— 3 — children_with_1_or_more_vaccination — incorrect — 0.333333 —
— 4 — children_in_2100 — incorrect — 0.333333 —
— 5 — children_in_2100 — incorrect — 0.333333 —
— 6 — children_in_2100 — correct — 0.333333 —
— 7 — children_in_2100 — incorrect — 0.333333 —
— 8 — children_in_2100 — incorrect — 0.333333 —
— 9 — children_in_2100 — incorrect — 0.333333 —

Question:

Can we reject the notion that the performance of college-educated adults was different than random guessing in the children_with_1_or_more_vaccination question with 95% confidence? Please answer with "yes" or "no".

Response:

`` ```python ``

Table 11: Example prompt for Program-of-Thoughts reasoning.

Data Description:

Public health has improved and evolved, but has the public's knowledge changed with it? This data set rosling_responses.csv contains sample responses for two survey questions posed by Hans Rosling during lectures to a wide array of college-educated audiences.

You are working with a pandas dataframe in Python. The name of the dataframe is 'df'.

You should use the tools below to answer the question posed of you:

python_repl_ast: A Python shell. Use this to execute python commands. Input should be a valid python command. When using this tool, sometimes output is abbreviated - make sure it does not look abbreviated before using it in your answer.

Use the following format:

Question: the input question you must answer
Thought: you should always think about what to do
Action: the action to take, should be python_repl_ast
Action Input: the input to the action, should be the code to execute
Observation: the result of the action
... (this Thought/Action/Action Input/Observation can repeat N times)
Thought: I now know the final answer
Final Answer: the final answer to the original input question, just provide the answer WITHOUT any other words

Note that you should import the libraries if needed.
DO NOT create any plotting.
For all outputs in code, THE print() function MUST be called.
If you use Action in this step, stop after generating the Action Input and await the execution outcome from python_repl_ast.
If you output the Final Answer in this step, do not use Action.

Here is an example of using the python_repl_ast:
Action: python_repl_ast
Action Input:

```
import pandas as pd
print(df.head())
```

This is the result:

```
— — question — response — prob_random_correct —
— 0 — children_with_1_or_more_vaccination — correct — 0.333333 —
— 1 — children_with_1_or_more_vaccination — correct — 0.333333 —
— 2 — children_with_1_or_more_vaccination — incorrect — 0.333333 —
— 3 — children_with_1_or_more_vaccination — incorrect — 0.333333 —
— 4 — children_with_1_or_more_vaccination — incorrect — 0.333333 —
```

Begin!
Question:

Can we reject the notion that the performance of college-educated adults was different than random guessing in the children_with_1_or_more_vaccination question with 95% confidence? Please answer with "yes" or "no".

Table 12: Example prompt for ReAct-style reasoning.

Assistant Instruction	You are a data analyst and good at quantitative reasoning. You are required to respond to a quantitative question using the provided data. The description and the question can be found below. Please just start answering and don't ask me any questions. Put your output in one message. Ensure that your final answer is positioned at the very end of your output, adhering to the format 'Final answer: [answer]
User	Data Description: Public health has improved and evolved, but has the public's knowledge changed with it? This data set rosling_responses.csv contains sample responses for two survey questions posed by Hans Rosling during lectures to a wide array of college-educated audiences. Question: Can we reject the notion that the performance of college-educated adults was different than random guessing in the children_with_1_or_more_vaccination question with 95% confidence? Please answer with "yes" or "no".

Table 13: Example prompt for the code interpreter assistants.

Data Description

The neuropathic pain diagnosis mainly consists of symptom diagnosis, pattern diagnosis, and pathophysiological diagnosis. The csv file Neuropathic_11.csv contains neuropathic pain diagnosis records in the form of tables of which the row represents different patients and the column represents different diagnostic labels.

Question

Which cause-and-effect relationship is more likely?

A. R L5 radiculopathy causes L L1 radiculopathy B. L L1 radiculopathy causes R L5 radiculopathy

C. No causal relationship exists

Please answer with A, B, or C.

Gold Answer

C

GPT-4 Prediction with Data Access

Step 1:
Code Interpreter

```
import pandas as pd
# Reading the dataset
file_path = ‘/mnt/data/Neuropathic_11.csv’
data = pd.read_csv(file_path)
# Display the first few records to understand the data structure
data.head()
```

Step 2:

Let’s proceed with analyzing the relevant columns to identify if there are any cause-and-effect relationships between the “R L5 radiculopathy” and “L L1 radiculopathy” conditions.

I will start by filtering these two columns from the dataset and then conducting an analysis to identify the likelihood of either of the radiculopathies causing the other. I will examine the data for the following patterns:

- If patients with “R L5 radiculopathy” also frequently have “L L1 radiculopathy” and not the other way around, it may suggest that A is more likely.
- Conversely, if patients with “L L1 radiculopathy” also frequently have “R L5 radiculopathy” and not the other way around, it may suggest that B is more likely.
- If there is no clear pattern or both conditions occur independently, it may suggest that C is more likely.

To do this analysis, I will compute conditional probabilities:

- P(L L1 — R L5)
- P(R L5 — L L1)

Then I will compare these probabilities to make an inference regarding the most likely cause-and-effect relationship.

Step 3:
Code Interpreter

```
# Extract the relevant columns
R_L5_col = ‘R L5 radiculopathy’
L_L1_col = ‘L L1 radiculopathy’
# Filter the dataset for these two columns
radiculopathy_data = data_corrected[[R_L5_col, L_L1_col]]
# Calculate the occurrence of each condition and their co-occurrence
R_L5_count = radiculopathy_data[R_L5_col].sum()
L_L1_count = radiculopathy_data[L_L1_col].sum()
co_occurrence_count = radiculopathy_data[(radiculopathy_data[R_L5_col] == 1) &
(radiculopathy_data[L_L1_col] == 1)].shape[0]
# Calculate the conditional probabilities
# Probability of L L1 given R L5
P_L_L1_given_R_L5 = co_occurrence_count / R_L5_count if R_L5_count else 0
# Probability of R L5 given L L1
P_R_L5_given_L_L1 = co_occurrence_count / L_L1_count if L_L1_count else 0
P_L_L1_given_R_L5, P_R_L5_given_L_L1
```

Step 4:

The conditional probabilities calculated are as follows:

- P(L L1 — R L5) $\approx$ 0.0075
- P(R L5 — L L1) $\approx$ 0.4122

These probabilities suggest that the occurrence of “L L1 radiculopathy” does not significantly increase when “R L5 radiculopathy” is present, given that P(L L1 — R L5) is quite low. On the other hand, the occurrence of “R L5 radiculopathy” is notably higher when “L L1 radiculopathy” is present, as P(R L5 — L L1) is much higher.

Therefore, based on these conditional probabilities, it is more likely that “L L1 radiculopathy” causes ‘R L5 radiculopathy” over the other proposed causal direction.

Final answer: B

Table 14: Full outputs of GPT-4 on the example case from QRDATA.

GPT-4 Prediction without Data Access	
Prediction	To address which cause-and-effect relationship is more likely between 'R L5 radiculopathy' and 'L L1 radiculopathy', we would generally look for statistical relationships in the data that may imply causality. Specifically, we would examine the frequency with which one condition appears before the other. If 'R L5 radiculopathy' generally appears before 'L L1 radiculopathy', it would lend support to the idea that A is more likely. Conversely, if 'L L1 radiculopathy' appears before 'R L5 radiculopathy', we might lean towards B. However, without access to the actual data, we can only reason based on general medical knowledge and the understanding of how radiculopathy typically presents. Radiculopathy is a condition resulting from a compressed nerve in the spine that can cause pain, numbness, or weakness in various parts of the body. Typically, this condition is localized and not known to cause another radiculopathy on the opposite side and a different vertebrae level of the body. Given both 'R L5' and 'L L1' radiculopathies are on different sides of the body and different spinal levels, it's not common for one to cause the other according to standard medical understanding. It is more likely that they occur independently due to separate causes. However, unilateral radiculopathy doesn't generally cause contralateral radiculopathy at a different spinal level. This is because the nerve root compression that leads to radiculopathy is typically localized, and it would be unusual for such an issue on one side of the spine to directly cause a similar issue on the opposite side at a different level. Therefore, it's likely that if both conditions are present in a patient, they arose independently or due to a systemic issue affecting the spine broadly, rather than one causing the other. Therefore, based on a qualitative argument and standard medical understanding, the answer is most likely that there is no causal relationship between the two conditions, which corresponds to option C. Final answer: C.

Table 15: Full outputs of GPT-4 on the example case from QRData when it does not have access to the data.