LongDA: Benchmarking LLM Agents for Long-Document Data Analysis

Yiyang Li[1], Zheyuan Zhang[1], Tianyi Ma[1], Zehong Wang[1], Keerthiram Murugesan[2], Chuxu Zhang[3], Yanfang Ye[1†]
[1]University of Notre Dame, [2]IBM Research, [3]University of Connecticut
[†]Corresponding Author {yli62, yye7}@nd.edu

🤗 huggingface.co/datasets/Yiyang-Ian-Li/LongDA

Abstract

We introduce LongDA, a data analysis benchmark for evaluating LLM-based agents under documentation-intensive analytical workflows. In contrast to existing benchmarks that assume well-specified schemas and inputs, LongDA targets real-world settings in which navigating long documentation and complex data is the primary bottleneck. To this end, we manually curate raw data files, long and heterogeneous documentation, and expert-written publications from 17 publicly available U.S. national surveys, from which we extract 505 analytical queries grounded in real analytical practice. Solving these queries requires agents to first retrieve and integrate key information from multiple unstructured documents before performing multi-step computations and writing executable code, which remains challenging for existing data analysis agents. To support the systematic evaluation under this setting, we develop LongTA, a tool-augmented agent framework that enables document access, retrieval, and code execution, and evaluate a range of proprietary and open-source models. Our experiments reveal substantial performance gaps even among state-of-the-art models, highlighting the challenges researchers should consider before applying LLM agents for decision support in real-world, high-stakes analytical settings.

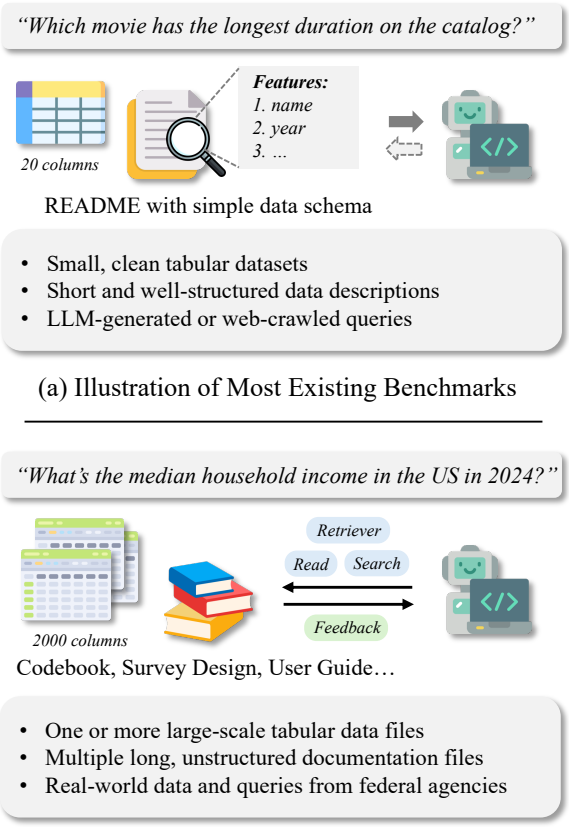

Figure 1: Comparison of existing data analysis benchmarks and LongDA.

1 Introduction

Recent advances in large language models (LLMs) (Chang et al., 2024; Ye et al., 2025) have enabled the development of LLM agents capable of assisting with data analysis tasks, including writing Python code (Hollmann et al., 2023; Guo et al., 2024; Sun et al., 2025a), executing statistical computations (Hong et al., 2025), and generating analytical reports (Zhang et al., 2025b). These agents show strong potential to reduce the manual effort required in data-driven research and decision-making, particularly in areas such as social science, economics, and public policy, where analytical workloads commonly involve complex and large-scale datasets (Sun et al., 2025b; Rahman et al., 2025; Wang et al., 2025). Accordingly, benchmarking the data analysis capabilities of LLM agents has become an important and active research direction (Zhang et al., 2024b; Hu et al., 2025).

Despite these advances, real-world data analysis remains fundamentally challenging. In practice, analytical workflows rarely begin with clean, well-documented datasets. Instead, analysts typically first navigate lengthy and heterogeneous documentation to understand dataset structures,

The Proceedings of Workshop on Document Intelligence and Understanding, pages 112–137
October 24, 2026 ©2026 Association for Computational Linguistics

Table 1: Comparison of related benchmarks. Context denotes the approximate number of context tokens per query. Syn., Hum., and Web denote synthetic, human, and web sources, respectively. ✓ indicates interactive environment.

Benchmark	Size	Data source	Query source	Interaction	Context
DS-1000 (Lai et al., 2023)	1,000	Web	Web	✗	900
Text2Analysis (He et al., 2024)	2,249	Hum.	Syn. / Hum.	✗	N/A
DSCodeBench (Ouyang et al., 2025)	1,000	Web	Syn. / Hum.	✗	640
DataSciBench (Zhang et al., 2025a)	222	N/A	Syn. / Hum. / Web	✗	200
InfiAgent-DABench (Hu et al., 2024)	257	Web	Syn.	✓	160
DSEval (Zhang et al., 2024b)	825	Web	Syn. / Hum.	✓	1,300
DACO (Wu et al., 2024)	1,942	Web	Syn. / Hum.	✓	890
DA-Code (Huang et al., 2024)	500	Web	Hum.	✓	370
DSBench (Jing et al., 2024)	466	Web	Web	✓	1,060
DABstep (Egg et al., 2025)	450	Company	Company	✓	5,200
LongDA (ours)	**505**	**Gov. agencies**	**Expert pubs.**	✓	**263,500**

variable definitions, and data preprocessing procedures (Koesten et al., 2021). This information is commonly scattered across multiple unstructured documents, such as codebooks, technical reports, and methodological appendices, which can span hundreds of thousands of tokens (Ale et al., 2025). Analysts generally need to complete this information-gathering stage before writing code to extract relevant variables and compute the results. For both humans and LLM agents, this documentation navigation stage is frequently the dominant bottleneck in real-world data analysis.

However, most existing data analysis benchmarks do not adequately capture this reality. As illustrated in Figure 1 and summarized in Table 1, prior benchmarks primarily evaluate data analysis under well-specified inputs, implicitly assuming that relevant variables, schemas, and data files are already known. Consequently, they overlook a critical component of real-world analysis: the ability to retrieve, interpret, and integrate information from long, unstructured documentation. Moreover, many benchmarks rely on LLM-generated or web-crawled queries, which may not reflect the analytical questions practitioners actually seek to answer. A more detailed discussion of related work is provided in Appendix A.

Motivated by these limitations, we introduce **LongDA**, the first data analysis benchmark that evaluates LLM agents under documentation-intensive analytical workflows. LongDA is constructed from federal agency–released data files, extensive documentation with an average length of *263k tokens*, and expert-written analytical publications. From these publications, we extract *505 real analytical queries* whose reported results are reproducible using publicly released data. To solve these queries, agents need to identify relevant variables, assign weights, and define populations before performing multi-step computations to obtain final answers. This remains a fundamental challenge for existing data analysis agents that are not designed to operate in this specific setting.

To enable systematic and controlled study of this challenge, we further develop **LongTA**, a lightweight tool-augmented agent framework that enables document access, retrieval, and code execution, serving as an evaluation scaffold and baseline for LongDA. Using LongTA, we conduct extensive experiments on a range of proprietary and open-source models, revealing substantial performance gaps and underscoring the limitations of current LLM agents in real-world analytical workflows.

Our contributions are summarized as follows:

- We introduce **LongDA**, a benchmark designed to evaluate data analysis agents on realistic datasets with long, complex documentation.
- We develop **LongTA**, a lightweight tool-augmented baseline for controlled document access and code execution for LongDA.
- We extensively evaluate state-of-the-art models, revealing substantial performance gaps and identifying information retrieval and tool-use strategy as primary bottlenecks.

2 LongDA Benchmark

2.1 Data Collection

To construct a realistic and practically meaningful data analysis benchmark, we collect seventeen U.S. national surveys spanning a broad range

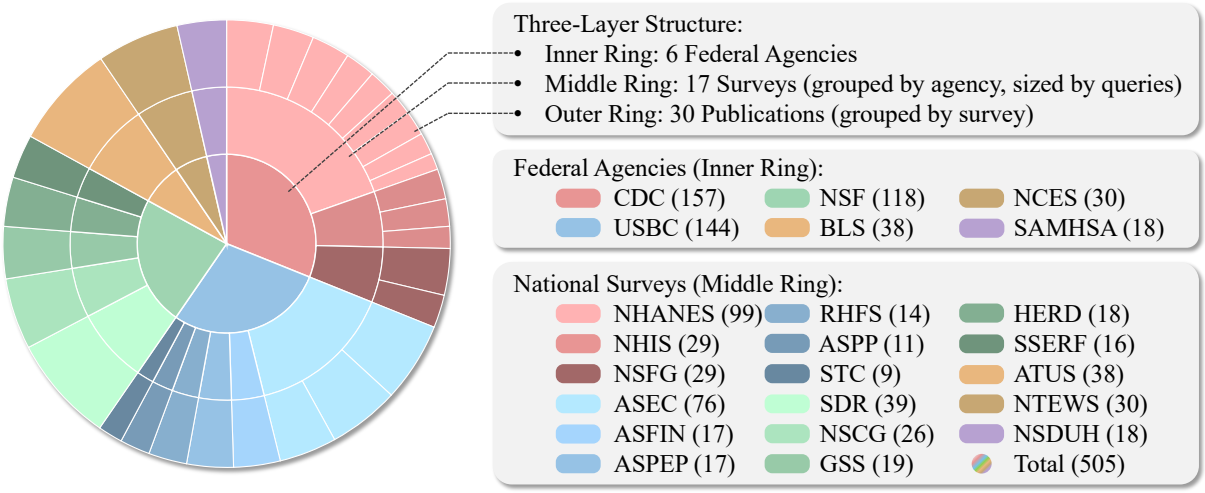

Figure 2: Hierarchical composition of the LongDA benchmark. Numbers in parentheses indicate the number of queries associated with each entity, with a total of 505 queries. Publications share the same color as their parent survey, and surveys belonging to the same federal agency are rendered in the same color family.

of domains, including public health, labor and employment, education, and the scientific workforce. These surveys are meticulously designed, conducted, and released by six U.S. federal agencies. They are routinely utilized by governments, researchers, and policymakers to provide valuable insights that inform public policy, social programs, and scientific research. For the complete list of survey abbreviations, official names, and source links, refer to Appendix B.1.

For each survey, we obtain three types of official resources from the corresponding agency websites: *(1) raw data files*, *(2) survey documentation*, and *(3) analytical publications* produced by domain experts. The documentation typically includes codebooks, methodological descriptions, and survey design reports, which are written for human analysts and commonly span long, unstructured texts. To ensure consistency and reproducibility, we collect data, documentation, and publications from a fixed survey year for each survey and organize all resources into a unified directory structure without modifying their original content. For the complete mapping between surveys and their associated publications, refer to Appendix B.2.

While all three resource types are collected during dataset construction, they play different roles in the benchmark: the publications are used exclusively for query extraction and are not accessible to agents during evaluation, while raw data files and the associated documentation are exposed to the agents when solving tasks. This design ensures that benchmark queries are grounded in real analytical practice while preventing agents from directly retrieving answers from expert-written reports.

2.2 Query and Ground Truth Extraction

After data collection, we extract queries and ground truths from the official analytical publications. Each query in LongDA consists of three components: *(1) a question*, *(2) an answer structure*, and *(3) additional information*. The *question* specifies the analytical goal based on the given survey. The *answer structure* defines the expected format of the response, which is either a single numerical value or a fixed-length list with predefined semantic elements (e.g., [*men*, *women*], [*under18*, *age18_64*, *age65plus*]). The *additional information* provides auxiliary context necessary for correctly interpreting the question. Together, these three components are the minimal information required to make each query both unambiguous for automatic evaluation and faithful to real-world analytical practice.

For each publication, we first verify that the reported results can be reproduced using the publicly released survey data, which is typically stated explicitly in the publication. We then construct queries based on reported numerical findings, together with their corresponding answer structures. When specifically noted, we extract relevant definitions or clarifications provided in the publication and include them as additional information to resolve potential linguistic ambiguities. This design ensures that queries are grounded in real-world analytical practice while being technically solvable, and encourages agents to perform genuine reasoning over data and documentation.

2.3 Benchmark Statistics

LongDA contains 505 queries in total. Figure 2 illustrates the hierarchical composition of LongDA, organized across three levels: federal agencies, surveys, and publications. In total, LongDA covers 6 U.S. federal agencies, 17 national surveys, and 30 official publications, reflecting the broad coverage of real-world data analysis scenarios.

A key characteristic of LongDA is the complexity of its accompanying documentation: with an average length of 263k tokens and a maximum length exceeding 735k tokens, LongDA possesses substantially longer contexts than existing benchmarks. The distribution of documentation lengths is shown in Figure 6. LongDA queries also exhibit structured answer formats: 221 queries (44%) require a single numerical answer, while the remaining 284 queries (56%) involve fixed-length lists with predefined semantic elements.

2.4 Evaluation Protocol

To control evaluation cost and better reflect realistic analytical workflows, queries derived from the same publication are presented to the agent in *blocks*, where each block contains all queries associated with a single publication (see Appendix C.2). Queries within a block typically revolve around shared themes or variables, allowing the agent to reuse intermediate understanding across related questions. Note that during evaluation, agents are restricted to accessing the raw data files and survey documentation corresponding to the given survey.

We evaluate agent performance along two primary dimensions: *effectiveness* and *efficiency*. Effectiveness is measured by **coverage rate** and **match rate**. Efficiency is characterized by total token consumption, total runtime over the full benchmark, and the average number of interaction steps per block. We describe each metric in detail below.

To demonstrate the agents' ability to follow task specifications and produce reliable, structured output, we compute the coverage rate, defined as the proportion of queries for which the answer syntactically conforms to the specified answer structure. Formally, given a query set $\mathcal{Q}$ and an agent's output set $\hat{\mathcal{A}}$, the coverage rate is defined as

$$\text{Coverage} = \frac{|\{\, q \in \mathcal{Q} \mid \hat{a}_q \text{ is structurally valid} \,\}|}{|\mathcal{Q}|},$$

where $\hat{a}_q$ denotes the agent's answer to query q.

The match rate measures numerical correctness under a predefined tolerance. For each query q with ground truth answer a_q and agent prediction $\hat{a}_q$, we define a tolerance parameter ϵ and set tolerance as

$$\tau(a_q) = \max(\epsilon \cdot |a_q|, 1),$$

ensuring that small-magnitude targets are not overly penalized. The prediction is considered correct if $|\hat{a}_q - a_q| \leq \tau(a_q)$ (applied element-wise for list answers). The match rate is the proportion of queries for which this condition holds.

3 LongTA Framework

To enable systematic study of LLM-based data analysis agents under documentation-intensive settings, we introduce LongTA, a lightweight tool-augmented agent framework that serves as an evaluation scaffold and baseline for LongDA. LongTA consists of a ReAct-style (Yao et al., 2022) coding agent and a set of specialized tools for interacting with raw data files and documentation.

3.1 ReAct Coding Agent

The coding agent follows an iterative ReAct loop in which the LLM interleaves tool calls for document navigation and Python code execution for numerical computation. This design matches realistic analysis workflows, where an analyst repeatedly consults documentation to identify variables and then runs code to examine the statistical results.

Block-level prompt. LongDA presents queries in blocks. For each block, we construct a prompt that contains: (i) global instructions, (ii) a list of questions, (iii) the expected answer structure for each question, (iv) additional information for each question, and (v) the list of available data files and documentation files for this survey. The prompt template is provided in Appendix C.1.

Iterative reasoning-and-action loop. Given the prompt, the agent runs for at most a fixed step budget. At each step, the agent generates a Python snippet to call tools or run analysis code. The loop terminates when the agent signals task completion or when the maximum step budget is reached.

Execution environment and safety. To support common survey analysis workloads while keeping the execution controlled, we configure the executor with an allowlist of commonly-used Python libraries (e.g., `pandas`, `numpy`, `scipy`, `statsmodels`) and standard library modules (e.g., `json`, `re`), balancing practicality and containment.

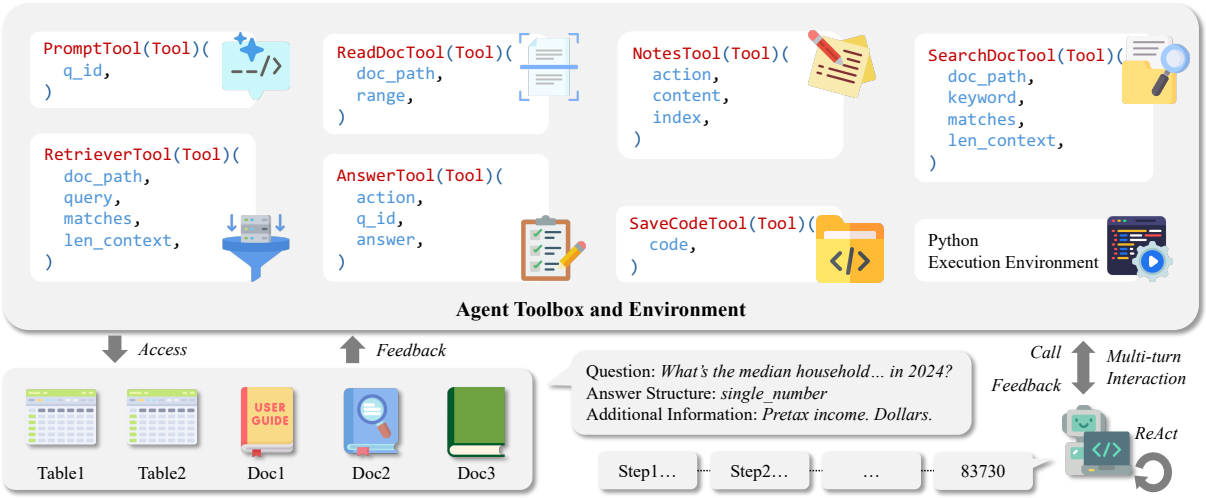

Figure 3: Illustration of LongTA. The agent solves queries through a ReAct-style multi-turn interaction, coordinating multiple tools for long-document navigation and code execution over heterogeneous data sources.

3.2 Tool Design

LongDA tasks are bottlenecked by documentation navigation: the agent must discover variable definitions, survey weights, and file schemas from long, heterogeneous documents before writing accurate analysis code. Accordingly, we design a small set of tools for document interaction, context management, and answer recording, built on top of a unified internal representation of documentation.

Unified document representation. All documentation files are exposed through a unified "document unit" abstraction. Given a document file, we parse it into a sequence of units depending on file type: PDFs are split into pages; text-like formats (TXT/PY/JSON) are split into lines; tabular formats (CSV/XLSX) are split into rows. Each unit has an identifier (e.g., "Page 3", "Line 120", "Row 15") along with the normalized text content. This design supports both keyword search and retrieval over large files while keeping tool outputs compact.

Document access and retrieval. To navigate long, heterogeneous documentation, we provide tools that support both targeted inspection and scalable retrieval. **(1)** `read_doc` enables direct inspection by returning either a short preview (when no range is specified) or user-selected units via one-based ranges (e.g., `1-3` or `2,5,7`), which is useful for reading codebook sections, methodological descriptions, or specific line/page ranges. Complementing this, **(2)** `search_doc` performs exact keyword search across units and returns snippets with a configurable context window, allowing the agent to quickly locate candidate variable names, weighting terms, or table captions. Since exact matching may be insufficient when the agent does not know the precise phrasing used in documentation, we further implement **(3)** `retriever`, a lightweight lexical retriever based on BM25 (Robertson et al., 2009): it splits each unit into overlapping chunks with configurable chunk size and overlap, builds index over all chunks, and returns the top-k relevant chunks.

Prompt revisiting and note-taking. To mitigate context loss in long interactions, we provide `prompt` for revisiting task specifications and `notes` for externalizing discovered information such as variable definitions and weight usage, which can be reused across queries within a block.

Answer recording and auditing. Automatic evaluation in LongDA requires answers to be structurally valid. We therefore enforce that final outputs are emitted through an `answer` tool. The tool validates that each answer is either a single number or a list of numbers, and stores the raw serialized value. This design separates reasoning traces from final outputs, enabling reliable automatic evaluation and post-hoc analysis. Finally, agents can use `save_code` to store the Python code used for a block, providing an audit trail that supports reproducibility checks and qualitative error analysis.

4 Experiments

4.1 Experiment Settings

We evaluate LongDA on a diverse set of state-of-the-art proprietary and open-source LLMs, including GPT-5 (OpenAI, 2025a), DeepSeek-V3.2 (Liu et al., 2025), Qwen3 series (Yang et al., 2025),

Table 2: Model performance comparison. Best results for each category are highlighted in **bold**. In (M), Out (M), and Total (M) denote the numbers of input tokens, output tokens, and total tokens (in millions), respectively.

Model	Coverage (%)	Match (%)	Avg. Steps	In (M)	Out (M)	Total (M)	Time (h)
GPT-5 (High)	**94.65**	68.91	6.20	7.38	0.87	8.25	5.74
GPT-5	91.09	**69.16**	5.50	5.75	0.65	6.40	3.58
GPT-5-mini	69.50	40.81	8.63	10.10	0.56	10.66	3.42
DeepSeek-V3.2	**67.33**	**53.00**	66.30	68.50	0.40	68.91	5.22
DeepSeek-V3.2-Thinking	61.58	47.82	37.60	35.55	0.83	36.38	9.08
Kimi-K2-0905	50.50	28.27	36.13	37.78	0.43	38.21	5.22
Qwen3-235B-A22B-Instruct	48.91	27.17	42.30	61.20	0.53	61.73	10.38
Qwen3-Coder-480B-A35B-Instruct	51.49	23.44	53.27	75.13	0.36	75.49	5.08
Qwen3-Next-80B-A3B-Instruct	49.50	21.85	21.50	22.00	0.37	22.36	1.64
GLM-4.7	27.13	19.18	81.17	119.42	0.59	120.01	8.42
GPT-OSS-120B	46.53	12.15	18.60	13.98	0.68	14.66	3.67

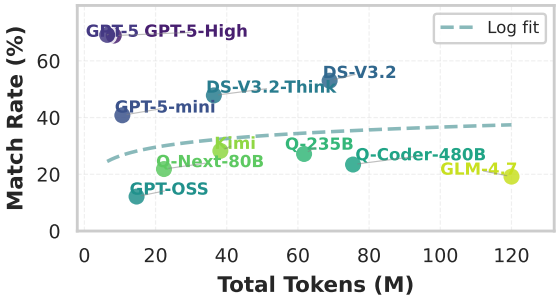

Figure 4: Match rate vs. total token consumption.

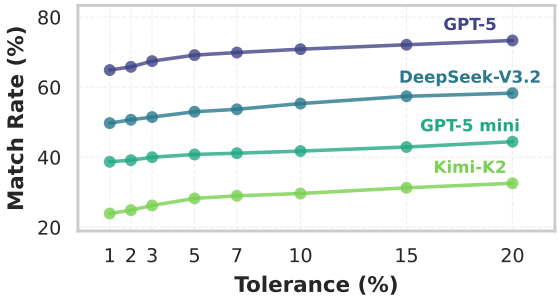

Figure 5: Match rate sensitivity to tolerance threshold.

Kimi-K2 (Team et al., 2025), GLM-4.7 (Z.ai, 2025), and GPT-OSS-120B (OpenAI, 2025b). All models are evaluated over LongTA with a budget of 100 steps and a tolerance threshold of 5% when computing the match rate.

4.2 Main Results

Overall Performance and Efficiency. Table 2 reveals substantial differences in both effectiveness and efficiency across models. The GPT-5 family achieves the highest coverage and match rates while requiring significantly fewer reasoning steps, fewer total tokens, and less wall-clock time. We attribute the strong performance of GPT-5 to three complementary factors. First, GPT-5 appears to possess richer prior knowledge of survey conventions and variable nomenclature, which often allows it to rapidly infer or approximate variable abbreviations before consulting documentation. This behavior is also reflected in our case study (Appendix D), where GPT-5 identifies many relevant keywords within the first few steps. Second, GPT-5 demonstrates substantially stronger capability in strategically leveraging the `search_doc` tool for information retrieval, as further analyzed in Section 4.3. Third, GPT-5 tends to execute larger code blocks per step (combining multiple keyword searches and sub-tasks within a single execution) rather than issuing one fine-grained tool call step by step. This execution style increases the effective information density of the context and avoids repeatedly injecting system prompts and intermediate states, thereby achieving both higher efficiency and stronger performance. Notably, even GPT-5 (High) achieves only 68.91% match rate, indicating substantial remaining headroom.

Among open-source models, DeepSeek-V3.2 exhibits comparatively stable generation behavior. However, we also observe that other models (e.g., Kimi and GLM series) sometimes fall into repetitive generation patterns. Such repetition substantially inflates output token usage and runtime, contributing to lower overall efficiency.

Scaling Behavior. Figure 4 illustrates the relationship between match rate and total token consumption. To analyze scaling behavior, we fit a logarithmic curve to the data. Note that GPT-5 and GPT-5 (High) are excluded from the fit as outliers to avoid distortion of the overall trend. Performance improves with increased token budget, though the gains exhibit diminishing returns, reflecting the intrinsic difficulty of LongDA.

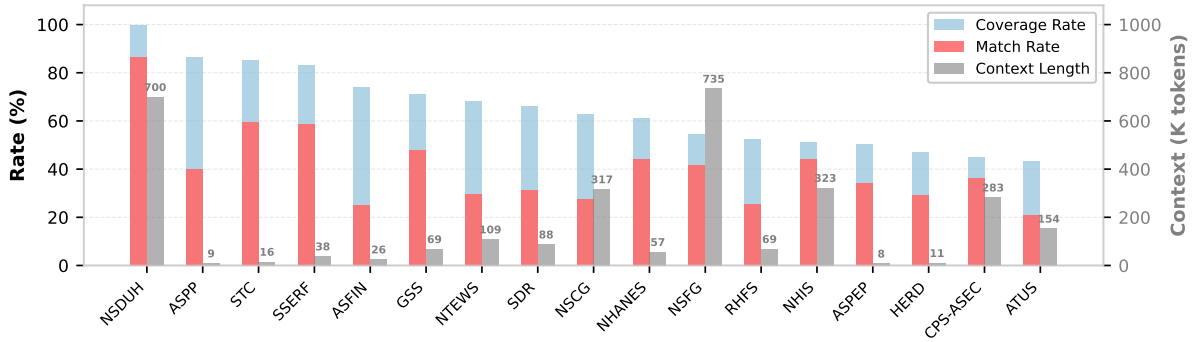

Figure 6: Survey-level performance averaged across all evaluated models.

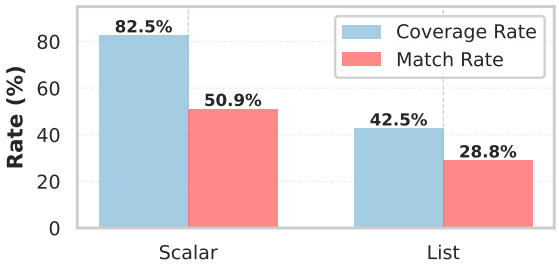

Figure 7: Performance by answer type (scalar vs. list).

Tolerance Sensitivity. Figure 5 studies sensitivity to the tolerance threshold. While match rate increases monotonically with tolerance, the relative ranking of models and the overall performance gaps remain stable, indicating that the evaluation is not overly sensitive to the exact tolerance choice.

Survey-Level Analysis. Figure 6 reports survey-level performance aggregated across all evaluated models, showing a mild negative correlation between documentation length and model performance. NSDUH is a notable exception: despite extremely long contexts, it achieves comparatively strong performance. This is likely because its publicly available microdata already includes many pre-coded variables derived from multiple raw fields, allowing models to answer queries once relevant variables are identified without requiring complex downstream computation.

Scalar vs. List Queries. Figure 7 compares scalar and list-structured queries. List queries are consistently more difficult than scalar ones, due to both increased computational complexity and the requirement to follow specific answer structures.

Effect of Explicit Reasoning. Comparing GPT-5 and DeepSeek-V3.2 with and without explicit reasoning variants, we observe that explicit reasoning does not provide clear performance improvements on LongDA. This suggests that the core difficulty of LongDA lies not in logical inference, but in discovering and extracting sufficient information from long, complex documentation. Moreover, reasoning-oriented models exhibit higher tool usage, longer outputs, and increased runtime (Table 2 and 3). We hypothesize that cautious reasoning induces excessive tool calls without substantially improving document understanding, introducing overhead with limited accuracy gains.

4.3 Tool Usage Analysis

Table 3 reports aggregate tool usage statistics. Additionally, to visualize tool-use dynamics, we extract full tool-call traces from agent execution logs and represent each block's trajectory as a sequence of tool invocations. Each trajectory is normalized to a $[0, 1]$ progress scale based on its maximum step count and discretized into 20 bins. For each bin, we compute the empirical distribution of tool usage across all trajectories of a model, producing the heatmaps in Figure 8.

Behavioral Analysis. GPT-5 variants strongly favor the `search` tool over the `retriever` and consistently invoke the `prompt` tool at the beginning of each block. Open-source models display more exploratory tool usage patterns with heavier reliance on retrieval and iterative refinement. These results indicate that successful performance on LongDA depends not only on language modeling and coding ability, but also on effective tool-use strategies for navigating long, complex documentation.

We further conduct a tool ablation study, as reported in Table 4. According to the results, removing `search_doc` leads to a substantial degradation for GPT-5, indicating that GPT-5 is particularly effective at leveraging `search_doc` for agentic problem solving in LongDA. In contrast, removing `retriever` causes only marginal fluctuations for DeepSeek-V3.2 and Qwen3-Coder. This sug-

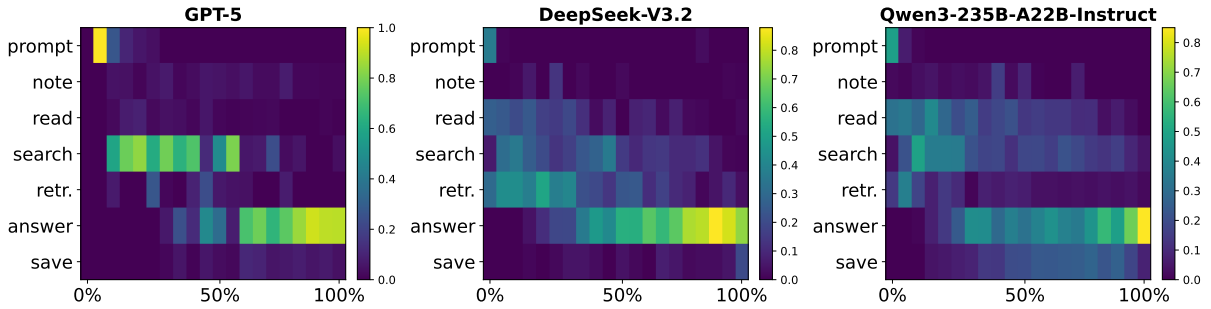

Figure 8: Normalized tool usage trajectories across three agent models. Each heatmap shows the probability distribution of tool calls over the normalized problem-solving progress (0–100%).

Table 3: Tool usage count by models.

Model	Prompt	Read	Search	Retr.	Note	Answer	Save
GPT-5	30	46	1,704	114	79	704	79
GPT-5 (High)	30	121	2,392	117	77	754	74
DeepSeek-V3.2	31	146	207	269	38	526	39
DeepSeek-V3.2-Thinking	30	122	206	393	39	893	95
Qwen3-235B-A22B-Instruct	30	309	371	146	57	596	293
Qwen3-Coder-480B-A35B-Instruct	84	187	231	15	17	528	18

Table 4: Ablation study on tool availability.

Model	Setting	Coverage (%)	Match (%)	Steps	In (M)	Out (M)	Time (h)
GPT-5	Full tools	**91.09**	**69.16**	5.50	5.75	0.65	3.58
GPT-5	w/o `search_doc`	80.20	58.30	6.03	6.09	0.65	2.84
DeepSeek-V3.2	Full tools	**67.33**	53.00	66.30	68.50	0.40	5.22
DeepSeek-V3.2	w/o `retriever`	66.93	**54.03**	65.07	67.04	0.40	5.36
Qwen3-Coder-480B-A35B-Instruct	Full tools	51.49	**23.44**	53.27	75.13	0.36	5.08
Qwen3-Coder-480B-A35B-Instruct	w/o `retriever`	**53.27**	21.52	48.00	79.48	0.40	4.27

gests that the performance gap between GPT-5 variants and other models cannot be explained solely by a preference for the `search_doc` tool. Instead, LongDA primarily challenges models' ability to strategically extract relevant evidence from long documentation and integrate retrieved information into coherent multi-step analytical workflows.

4.4 Summary of Findings

Overall, our experiments lead to three key observations. **First**, current state-of-the-art models remain far from solving documentation-intensive data analysis reliably. Even the strongest model, GPT-5 (High), achieves only 68.91% match rate, indicating substantial headroom. **Second**, success on LongDA is driven primarily by information retrieval and tool-use strategy rather than by pure reasoning capability. Models that are better at retrieving relevant evidence from long documentation and orchestrating tool calls consistently outperform others. **Third**, longer contexts and more complex answer structures amplify difficulty, reflecting the challenges of information retrieval, instruction following, and multi-step computation.

5 Conclusion

We introduce LongDA, the first benchmark for evaluating data analysis agents under documentation-intensive analytical workflows. By grounding tasks in large-scale government datasets and long, unstructured documentation, LongDA exposes fundamental challenges to current LLM-based agents, particularly in information retrieval and document navigation. Our extensive experiments show that even state-of-the-art models struggle with documentation-intensive analytical pipelines, where performance depends critically on effective evidence retrieval and strategic tool use under long-context constraints. We hope LongDA will serve as a challenging and practical testbed for advancing the development of reliable data analysis agents.

Limitations

LongDA has several limitations that should be considered when interpreting results and when using the benchmark.

Scope and representativeness. LongDA is constructed from 17 U.S. national surveys and associated publications, which provide breadth across agencies and domains but still represent a small slice of the full landscape of real-world data analysis. The benchmark may under-represent domains with substantially different data modalities (e.g., proprietary enterprise data) or workflows (e.g., streaming analytics, database-centric ETL).

Evaluation focuses on final numbers. Our primary metrics (coverage and match) evaluate whether agents produce structurally valid outputs and numerically correct answers within a tolerance. This design supports scalable automatic evaluation, but it does not directly measure intermediate analytical quality (e.g., whether the agent's code follows best practices, produces confidence intervals, or correctly handles edge cases) nor does it capture the quality of explanations and reporting narratives.

Survey design complexity is only partially exercised. While LongDA tasks require reading documentation and applying sample weights, many survey analyses in practice also involve variance estimation and design-based uncertainty quantification (e.g., strata/PSU adjustments, replicate weights, finite population corrections, and survey-specific variance estimators). LongDA currently does not include queries that explicitly evaluate variance estimation (e.g., standard errors or confidence intervals). As a result, strong performance on LongDA does not imply full competence in general complex-survey inference beyond point estimation.

Answer memorization risk. Although LongDA is designed such that agents are expected to derive answers through document retrieval and multi-step computation, this requirement is not strictly enforced. Consequently, a model may occasionally produce the correct answer from information memorized during pretraining rather than by actually performing the intended analysis. Future evaluations could mitigate this risk by verifying intermediate computation traces and required tool calls before accepting the final answer.

Ethical Considerations

LongDA is constructed from publicly released U.S. government survey data and documentation. The benchmark does not include personally identifiable information (PII) and does not require any individual-level inference. Given the high-stakes nature of real-world data analysis, outputs produced by LLM agents should not be used for automated decision-making without human verification. We report token usage and runtime to make evaluation cost explicit and to encourage more efficient and reproducible methods.

Acknowledgements

This work was partially supported by the NSF under grants IIS-2533550, IIS-2321504, IIS-2217239, CNS-2426514, and CMMI-2146076, Notre Dame Strategic Framework Research Grant (2025), and Notre Dame Poverty Research Package (2025). Any expressed opinions, findings, and conclusions or recommendations are those of the authors and do not necessarily reflect the views of the sponsors.

References

Laha Ale, Robert Gentleman, Christopher Endres, Sam Pullman, Nathan Palmer, Rafael Goncalves, and Deepayan Sarkar. 2025. Enhancing statistical analysis of real world data. *Database*, 2025:baaf073.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, and 1 others. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45.

Alex Egg, Martin Iglesias Goyanes, Friso Kingma, Andreu Mora, Leandro von Werra, and Thomas Wolf. 2025. Dabstep: Data agent benchmark for multi-step reasoning. *arXiv preprint arXiv:2506.23719*.

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.

Siyuan Guo, Cheng Deng, Ying Wen, Hechang Chen, Yi Chang, and Jun Wang. 2024. Ds-agent: Automated data science by empowering large language models with case-based reasoning. *arXiv preprint arXiv:2402.17453*.

Xinyi He, Mengyu Zhou, Xinrun Xu, Xiaojun Ma, Rui Ding, Lun Du, Yan Gao, Ran Jia, Xu Chen, Shi Han, and 1 others. 2024. Text2analysis: A benchmark of table question answering with advanced data analysis

and unclear queries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18206–18215.

Noah Hollmann, Samuel Müller, and Frank Hutter. 2023. Large language models for automated data science: Introducing caafe for context-aware automated feature engineering. *Advances in Neural Information Processing Systems*, 36:44753–44775.

Sirui Hong, Yizhang Lin, Bang Liu, Bangbang Liu, Binhao Wu, Ceyao Zhang, Danyang Li, Jiaqi Chen, Jiayi Zhang, Jinlin Wang, and 1 others. 2025. Data interpreter: An llm agent for data science. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19796–19821.

Chuxuan Hu, Dwip Dalal, and Xiaona Zhou. 2025. A dataset-centric survey of llm-agents for data science.

Xueyu Hu, Ziyu Zhao, Shuang Wei, Ziwei Chai, Qianli Ma, Guoyin Wang, Xuwu Wang, Jing Su, Jingjing Xu, Ming Zhu, and 1 others. 2024. Infiagent-dabench: Evaluating agents on data analysis tasks. *arXiv preprint arXiv:2401.05507*.

Yiming Huang, Jianwen Luo, Yan Yu, Yitong Zhang, Fangyu Lei, Yifan Wei, Shizhu He, Lifu Huang, Xiao Liu, Jun Zhao, and 1 others. 2024. Da-code: Agent data science code generation benchmark for large language models. *arXiv preprint arXiv:2410.07331*.

Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. 2025. Aide: Ai-driven exploration in the space of code. *arXiv preprint arXiv:2502.13138*.

Liqiang Jing, Zhehui Huang, Xiaoyang Wang, Wenlin Yao, Wenhao Yu, Kaixin Ma, Hongming Zhang, Xinya Du, and Dong Yu. 2024. Dsbench: How far are data science agents from becoming data science experts? *arXiv preprint arXiv:2409.07703*.

Laura Koesten, Kathleen Gregory, Paul Groth, and Elena Simperl. 2021. Talking datasets–understanding data sensemaking behaviours. *International journal of human-computer studies*, 146:102562.

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Wen-tau Yih, Daniel Fried, Sida Wang, and Tao Yu. 2023. Ds-1000: A natural and reliable benchmark for data science code generation. In *International Conference on Machine Learning*, pages 18319–18345. PMLR.

Ziming Li, Qianbo Zang, David Ma, Jiawei Guo, Tuney Zheng, Minghao Liu, Xinyao Niu, Yue Wang, Jian Yang, Jiaheng Liu, and 1 others. 2024. Autokaggle: A multi-agent framework for autonomous data science competitions. *arXiv preprint arXiv:2410.20424*.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

OpenAI. 2025a. Gpt-5 system card. Technical report, OpenAI.

OpenAI. 2025b. Introducing gpt-oss and gpt-oss-120b.

Shuyin Ouyang, Dong Huang, Jingwen Guo, Zeyu Sun, Qihao Zhu, and Jie M Zhang. 2025. Dscodebench: A realistic benchmark for data science code generation. *arXiv preprint arXiv:2505.15621*.

Mizanur Rahman, Amran Bhuiyan, Mohammed Saidul Islam, Md Tahmid Rahman Laskar, Ridwan Mahbub, Ahmed Masry, Shafiq Joty, and Enamul Hoque. 2025. Llm-based data science agents: A survey of capabilities, challenges, and future directions. *arXiv preprint arXiv:2510.04023*.

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Maojun Sun, Ruijian Han, Binyan Jiang, Houduo Qi, Defeng Sun, Yancheng Yuan, and Jian Huang. 2025a. Lambda: A large model based data agent. *Journal of the American Statistical Association*, pages 1–13.

Maojun Sun, Ruijian Han, Binyan Jiang, Houduo Qi, Defeng Sun, Yancheng Yuan, and Jian Huang. 2025b. A survey on large language model-based agents for statistics and data science. *The American Statistician*, pages 1–14.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Patara Trirat, Wonyong Jeong, and Sung Ju Hwang. 2024. Automl-agent: A multi-agent llm framework for full-pipeline automl. *arXiv preprint arXiv:2410.02958*.

Peiran Wang, Yaoning Yu, Ke Chen, Xianyang Zhan, and Haohan Wang. 2025. Large language model-based data science agent: A survey. *arXiv preprint arXiv:2508.02744*.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. 2024. Executable code actions elicit better llm agents. In *Forty-first International Conference on Machine Learning*.

Xueqing Wu, Rui Zheng, Jingzhen Sha, Te-Lin Wu, Hanyu Zhou, Mohan Tang, Kai-Wei Chang, Nanyun Peng, and Haoran Huang. 2024. Daco: Towards application-driven and comprehensive data analysis via code generation. *Advances in Neural Information Processing Systems*, 37:90661–90682.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Yanfang Ye, Zheyuan Zhang, Tianyi Ma, Zehong Wang, Yiyang Li, Shifu Hou, Weixiang Sun, Kaiwen Shi, Yijun Ma, Wei Song, and 1 others. 2025. Llms4all: A review of large language models across academic disciplines. *arXiv preprint arXiv:2509.19580*.

Z.ai. 2025. Glm-4.7: Advancing the coding capability. Technical report, Zhipu AI.

Dan Zhang, Sining Zhoubian, Min Cai, Fengzu Li, Lekang Yang, Wei Wang, Tianjiao Dong, Ziniu Hu, Jie Tang, and Yisong Yue. 2025a. Datascibench: An llm agent benchmark for data science. *arXiv preprint arXiv:2502.13897*.

Lei Zhang, Yuge Zhang, Kan Ren, Dongsheng Li, and Yuqing Yang. 2024a. Mlcopilot: Unleashing the power of large language models in solving machine learning tasks. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2931–2959.

Shaolei Zhang, Ju Fan, Meihao Fan, Guoliang Li, and Xiaoyong Du. 2025b. Deepanalyze: Agentic large language models for autonomous data science. *arXiv preprint arXiv:2510.16872*.

Shujian Zhang, Chengyue Gong, Lemeng Wu, Xingchao Liu, and Mingyuan Zhou. 2023. Automl-gpt: Automatic machine learning with gpt. *arXiv preprint arXiv:2305.02499*.

Yuge Zhang, Qiyang Jiang, XingyuHan XingyuHan, Nan Chen, Yuqing Yang, and Kan Ren. 2024b. Benchmarking data science agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5677–5700.

Yizhang Zhu, Shiyin Du, Boyan Li, Yuyu Luo, and Nan Tang. 2024. Are large language models good statisticians? *Advances in Neural Information Processing Systems*, 37:62697–62731.

A Related Work

A.1 Data Analysis Benchmarks

Existing data analysis benchmarks can be broadly categorized into single-turn and interactive settings. Single-turn benchmarks such as DS-1000 (Lai et al., 2023), DSCodeBench (Ouyang et al., 2025), and DataSciBench (Zhang et al., 2025a) primarily evaluate one-shot code generation or analytical reasoning with well-specified prompts and structured inputs, placing limited demands on document-level language understanding.

More recent interactive benchmarks (e.g., InfiAgent-DABench (Hu et al., 2024), DSEval (Zhang et al., 2024b), DA-Code (Huang et al., 2024), DSBench (Jing et al., 2024)) allow agents to perform multi-step reasoning and interact with execution environments. However, most of these benchmarks are still built on relatively small datasets with limited or simplified documentation, and therefore do not fully capture the challenge of navigating large-scale, unstructured data descriptions. In contrast, LongDA evaluates data analysis agents on real-world datasets with extensive and complex documentation, requiring language understanding, information retrieval, and tool-assisted computation.

A.2 LLM Data Analysis Agents

Recent work has increasingly explored using large language models as autonomous agents for data analysis. Early efforts primarily focused on leveraging LLMs for code generation and computational reasoning, simplifying complex analytical workflows through program synthesis and tool execution (Gao et al., 2023; Zhu et al., 2024; Wang et al., 2024). Building on this foundation, a new generation of data science agents has emerged, integrating function-calling, code interpreters, and external tools to support end-to-end analytical processes, including Data Interpreter (Hong et al., 2025), DS-Agent (Guo et al., 2024), LAMBDA (Sun et al., 2025a), and related systems (Zhang et al., 2023, 2024a; Li et al., 2024; Trirat et al., 2024; Jiang et al., 2025).

Collectively, these systems demonstrate that LLM agents can effectively automate many components of data science pipelines, such as feature engineering, hyper-parameter tuning, model training, and evaluation. However, most existing approaches assume relatively clean, well-documented inputs and place limited emphasis on navigating large-scale, heterogeneous documentation. As a result, they under-exercise the document understanding and information discovery capabilities that dominate real-world analytical workflows. Moreover, scalability, long-context management, and robust multi-step planning remain open challenges for current agent frameworks.

Table 5: Survey Abbreviations, Official Names, and Data Sources

Abbrev.	Official Survey Name	Agency	Year	Data Source
ATUS	American Time Use Survey	Bureau of Labor Statistics	2024	Link
NHANES	National Health and Nutrition Examination Survey	National Center for Health Statistics	2021–2023	Link
NHIS	National Health Interview Survey	National Center for Health Statistics	2023	Link
NSFG	National Survey of Family Growth	National Center for Health Statistics	2022–2023	Link
GSS	Graduate Student Survey	National Science Foundation	2023	Link
HERD	Higher Education R&D Survey	National Science Foundation	2023	Link
NSCG	National Survey of College Graduates	National Science Foundation	2023	Link
SDR	Survey of Doctorate Recipients	National Science Foundation	2023	Link
SSERF	Survey of Science & Engineering Research Facilities	National Science Foundation	2023	Link
NTEWS	National Training, Education, and Workforce Survey	National Center for Education Statistics	2024	Link
NSDUH	National Survey on Drug Use and Health	SAMHSA	2023	Link
ASFIN	Annual Survey of State Government Finances	U.S. Census Bureau	2021	Link
ASPEP	Annual Survey of Public Employment & Payroll	U.S. Census Bureau	2024	Link
ASPP	Annual Survey of Public Pensions	U.S. Census Bureau	2019	Link
CPS–ASEC	Current Population Survey – ASEC	U.S. Census Bureau & BLS	2024	Link
RHFS	Rental Housing Finance Survey	U.S. Census Bureau	2023	Link
STC	Survey of State Tax Collections	U.S. Census Bureau	2020	Link

Table 6: Mapping Between Surveys and Source Publications

Survey	Publication	Agency	Year
ATUS	American Time Use Survey: 2024 Results	Bureau of Labor Statistics	2024
NHANES	Data Brief No. 508: Obesity Prevalence	National Center for Health Statistics	2021–2023
NHANES	Data Brief No. 511: Hypertension Prevalence	National Center for Health Statistics	2021–2023
NHANES	Data Brief No. 515: Sleep Duration	National Center for Health Statistics	2021–2023
NHANES	Data Brief No. 516: Physical Activity	National Center for Health Statistics	2021–2023
NHANES	Data Brief No. 519: Anemia Prevalence	National Center for Health Statistics	2021–2023
NHANES	Data Brief No. 527: Depression Prevalence	National Center for Health Statistics	2021–2023
NHANES	Data Brief No. 533: Fast Food Consumption	National Center for Health Statistics	2021–2023
NHANES	Data Brief No. 538: Prescription Opioid Use	National Center for Health Statistics	2021–2023
NHANES	Data Brief No. 540: Diabetes Prevalence	National Center for Health Statistics	2021–2023
NHIS	Data Brief No. 518: Chronic Pain	National Center for Health Statistics	2023
NHIS	Data Brief No. 528: Depression Medication	National Center for Health Statistics	2023
NHIS	Data Brief No. 529: COPD Prevalence	National Center for Health Statistics	2023
NSFG	Data Brief No. 520: Contraceptive Use	National Center for Health Statistics	2022–2023
NSFG	Data Brief No. 539: Pregnancy Intentions	National Center for Health Statistics	2022–2023
GSS	NSF 25–316: Graduate Students and Postdoctorates in SEH	National Science Foundation	2023
HERD	NSF 25–313: Higher Education R&D Expenditures	National Science Foundation	2023
NSCG	NSF 25–331: National Survey of College Graduates	National Science Foundation	2023
SDR	NSF 25–320: Survey of Doctorate Recipients	National Science Foundation	2023
SSERF	NSF 25–318: Survey of S&E Research Facilities	National Science Foundation	2023
NTEWS	NSF 25–352: National Training, Education, and Workforce Survey	National Center for Education Statistics	2024
NSDUH	2023 NSDUH Annual National Report	SAMHSA	2023
ASFIN	State Government Finances Summary	U.S. Census Bureau	2021
ASPEP	Public Employment & Payroll Summary Report	U.S. Census Bureau	2024
ASPP	Public Pensions Summary	U.S. Census Bureau	2019
CPS–ASEC	Income in the United States	U.S. Census Bureau & BLS	2024
CPS–ASEC	Poverty in the United States	U.S. Census Bureau & BLS	2024
CPS–ASEC	Health Insurance Coverage in the United States	U.S. Census Bureau & BLS	2024
RHFS	Rental Housing Finance Survey	U.S. Census Bureau	2023
STC	State Government Tax Collections Summary Report	U.S. Census Bureau	2020

B Survey Metadata

B.1 Survey Abbreviations, Full Names, and Official Links

Table 5 provides the complete list of survey abbreviations used throughout the paper, together with their official full names, releasing agencies, and public data access links. This information is provided to ensure reproducibility and facilitate independent verification of the data sources.

B.2 Survey–Publication Mapping

Table 6 summarizes the correspondence between surveys and the official analytical publications from which benchmark queries are extracted. Each query in LongDA is derived from exactly one publication associated with a specific survey, and publications are never exposed to agents during evaluation.

C Prompt Design

C.1 Prompt Template

This section presents the complete prompt template used by the ReAct-style agent in all experiments. The prompt explicitly specifies the global task description, available tools, question blocks, answer formats, and execution constraints, and is shown verbatim in Figure 9 to ensure full reproducibility of the experimental setup.

C.2 Question Block Example

To further illustrate the structure of the task input provided to the agent, Figure 10 presents a concrete example of a question block drawn from the `ASFIN` survey. The example contains multiple queries, their required answer formats, and the associated data and documentation files.

Prompt Template

Task: Analyze the survey data and associated documentation to answer the following questions.

Survey: `{survey}`

Questions and Expected Answer Structures: `{qa_block}`

Available Data Files: `{paths/to/the/data/files}`

Available Documentation Files: `{paths/to/the/doc/files}`

Instructions:

1. **Understand the Task**
   - Use the `prompt` tool to review all questions, expected answer structures, and additional information.
   - Identify which data files and documentation are required.
2. **Gather Information**
   - Use `retriever` or `search_doc` to locate relevant documentation.
   - Use `read_doc` to inspect specific files or sections.
   - Record key information (e.g., variable definitions, column meanings, survey weights) using the `notes` tool.
   - Do *not* rely on prior knowledge; always verify from documentation.
3. **Analyze Data**
   - Write and execute Python code to load and process the data.
   - Apply sampling weights when required by the survey design.
4. **Format and Save Answers**
   - Format each answer *exactly* according to the specified answer structure.
   - Respect unit requirements (e.g., percentages, counts, thousands, millions).
   - Report all numerical values with **one decimal place**.
   - Use the `answer` tool with `action='add'` to store each answer by question ID.
   - Use `save_code` to save the analysis code after completing each block.
5. **Working Strategy**
   - Questions may be solved incrementally; solving all at once is not required.
   - Use `answer` with `action='view'` to inspect saved answers.
   - Use `notes` with `action='list'` to review recorded information.

Figure 9: Prompt template used by the agent for solving LongDA tasks.

Example Question Block (ASFIN 2021)

Survey: ASFIN — Annual Survey of State Government Finances (2021)

Questions and Expected Answer Structures

- **Q1:** What was the total state government revenue in 2021?
  Answer Structure: `single_number`
  Additional Info: Trillions of dollars.

  ...

- **Q4:** What percentage of total 2021 revenue came from intergovernmental revenue and insurance trust revenue?
  Answer Structure: `[intergovernmental_revenue, insurance_trust_revenue]`
  Additional Info: NA.

  ...

- **Q11:** What was liquor store revenue, utility revenue, miscellaneous general revenue, and current charge revenue in 2021?
  Answer Structure: `[liquor_store_revenue, utility_revenue, miscellaneous_general_revenue, current_charges_revenue]`
  Additional Info: Billions of dollars.

  ...

Available Data Files

- `benchmark/ASFIN/data/21state35.txt`

Available Documentation Files

- `benchmark/ASFIN/docs/2021 F13_Form.pdf`
- `benchmark/ASFIN/docs/statetechdoc2021.pdf`
- `benchmark/ASFIN/docs/public-use-file-layout.xlsx`
- `benchmark/ASFIN/docs/itemcodes.xlsx`
- `benchmark/ASFIN/docs/government-ids.xlsx`

Instructions: ...

Figure 10: Concrete example of a LongDA question block.

D Case Study: Agent Interaction Example

To illustrate how an agent solves LongDA tasks end-to-end, we present a real interaction trace produced by GPT-5 on a block from the NHANES survey (NCHS Data Brief No. 508). The block contains multiple related prevalence queries with strict answer-structure constraints. Throughout the interaction, the agent has access only to the NHANES microdata and accompanying documentation. In this example, GPT-5 successfully answers the entire block.

D.1 Overview of the Block

This NHANES block asks for adult (age $\geq$ 20) obesity and severe obesity prevalence, both overall and stratified by sex, age group, and education group. Solving the block requires the agent to (i) identify the correct BMI variable, (ii) determine the appropriate sampling weight for examination-based analysis, (iii) map demographic variables to subgroup definitions, and (iv) implement weighted estimators in Python while satisfying the required output ordering.

D.2 Step-by-Step Interaction

Step 1: Re-establishing task context. The agent begins by invoking the `prompt` tool to re-check the full task specification, including question IDs, answer structures, and auxiliary constraints. This behavior is consistent with the strategy we observe for GPT-5 across blocks: it refreshes global task requirements early to minimize downstream formatting errors.

Step 2–4: Documentation navigation and variable identification. The primary challenge of this block lies in identifying the correct variables and analytic guidance from long, heterogeneous documentation. GPT-5 first uses `search_doc` to locate the BMI field in the anthropometry codebook `BMX_L.pdf`. It identifies `BMXBMI` and confirms its definition as body mass index (kg/m^2), already rounded to one decimal place in the released data.

In parallel, it queries the demographics codebook `DEMO_L.pdf` to locate:

- `RIDAGEYR` for age,
- `RIAGENDR` for sex (coded as 1/2),
- `DMDEDUC2` for adult education categories.

The agent then explicitly searches for weight variables in `DEMO_L.pdf`, discovering both `WTMEC2YR` and `WTINT2YR`, and correctly determines that `WTMEC2YR` is the appropriate 2-year MEC examination weight for analyses involving examination measures such as BMI. This distinction is critical: using interview weights would produce biased estimates.

Finally, GPT-5 consults pregnancy-related documentation and records the discovered variable semantics, coding schemes, subgroup mappings, and weight selection using the `notes` tool. This externalization of key analytical decisions enables clean, reusable code across all queries in the block.

Step 5: Single-pass computation via executable Python. After resolving all documentation-dependent uncertainties, GPT-5 performs most of the computation in a single extended code execution. Specifically, it:

1. Loads `DEMO_L.xpt` and `BMX_L.xpt` and merges them on respondent identifier `SEQN`;
2. Filters to adults aged $\geq$ 20 with valid BMI and positive MEC weights (`WTMEC2YR` $>$ 0);
3. Constructs indicator variables for obesity (BMI $\geq$ 30) and severe obesity (BMI $\geq$ 40);
4. Defines subgroup variables:
   - age groups: 20–39, 40–59, 60+;
   - education groups: `hs_or_less` (DMDEDUC2 $\in \{1, 2, 3\}$), `some_college` (4), `bachelors_plus` (5), excluding refused/unknown.
5. Implements a weighted prevalence estimator

$$\widehat{p} \;=\; 100 \cdot \frac{\sum_i w_i y_i}{\sum_i w_i},$$

   using `WTMEC2YR` as w_i and the indicator variable as y_i.

The agent then computes each query by slicing the dataset into the required subpopulation and applying the same estimator. All outputs are rounded to one decimal place to match benchmark formatting constraints.

Answer recording and audit trail. Upon completing the computations, GPT-5 records each result using the `answer` tool under the corresponding

question ID, guaranteeing strict adherence to the required output structures. It also invokes `save_code` to preserve the full analysis pipeline, enabling reproducibility, auditing, and qualitative error analysis.

Step 6: Finalization. Finally, the agent emits a structured summary of all answers via `final_answer`, serving as a human-readable recap and signaling termination of the interaction. Benchmark evaluation, however, relies exclusively on the structured outputs recorded through the `answer` tool.

D.3 Implications for LongDA Evaluation

This example illustrates several core properties that LongDA is explicitly designed to evaluate:

- **Long-document grounding.** Correct solutions require discovering variable definitions, coding schemes, and weight selection rules from documentation rather than relying on prior knowledge alone.
- **Tool-assisted analytical workflow.** Successful execution depends on iterative tool use for information retrieval followed by executable computation.
- **Block-level reuse of intermediate knowledge.** Once variables and weights are identified, the same intermediate understanding supports multiple queries within the block.
- **Strict output compliance.** The agent must satisfy rigid structural and ordering constraints to enable automatic evaluation.

D.4 Summary

Overall, this interaction exemplifies the intended LongDA workflow: the agent first resolves documentation-dependent uncertainties and then executes a coherent analysis program to answer multiple related questions correctly. Such tasks go beyond code synthesis alone and require reliable agentic document navigation and tool use under realistic constraints.

Example Trace: GPT-5 on NHANES (Data Brief No. 508)

This appendix shows an execution trace (abridged only by formatting) for a LongDA block. The agent alternates between tool calls (e.g., `prompt`, `search_doc`, `read_doc`) and Python code execution, and records final answers via the `answer` tool.

Step 1

Code Action

```
# Step 1: Review the full prompt to confirm
↪  tasks
full_prompt = prompt()
print(full_prompt)
```

Observations

```
Analyze the survey data and relevant
↪  documents to answer the following
↪  questions.
Survey: NHANES

Questions and Expected Answer Structures:
Question ID: NHANES_NCHS Data Brief No.
↪  508_q1
Q1: What was the overall prevalence of
↪  obesity among adults age 20 and older?
Expected Answer Structure: single_number
Additional Info: Body mass index (BMI):
↪  Calculated as weight in kilograms divided
↪  by height in meters squared, rounded to
↪  one decimal place. Obesity: Defined as a
↪  BMI of greater than or equal to 30.

Question ID: NHANES_NCHS Data Brief No.
↪  508_q2
Q2: What were the obesity prevalences for
↪  men and women?
Expected Answer Structure: [men, women]
Additional Info: NA.

Question ID: NHANES_NCHS Data Brief No.
↪  508_q3
Q3: What were the obesity prevalences for
↪  age groups 20-39, 40-59, and 60+?
Expected Answer Structure: [age20_39,
↪  age40_59, age60plus]
Additional Info: NA.

Question ID: NHANES_NCHS Data Brief No.
↪  508_q4
Q4: What were the obesity prevalences by
↪  education level?
Expected Answer Structure: [hs_or_less,
↪  some_college, bachelors_plus]
Additional Info: NA.

Question ID: NHANES_NCHS Data Brief No.
↪  508_q5
Q5: What was the overall prevalence of
↪  severe obesity among adults?
Expected Answer Structure: single_number
Additional Info: Severe obesity: Defined as
↪  a BMI of greater than or equal to 40.

Question ID: NHANES_NCHS Data Brief No.
↪  508_q6
Q6: What were the severe obesity prevalences
↪  for age groups 20-39, 40-59, and 60+?
Expected Answer Structure: [age20_39,
↪  age40_59, age60plus]
Additional Info: NA.

Question ID: NHANES_NCHS Data Brief No.
↪  508_q7
```

```
Q7: What were the severe obesity prevalences
↪  for men and women?
Expected Answer Structure: [men, women]
Additional Info: NA.

Question ID: NHANES_NCHS Data Brief No.
↪  508_q8
Q8: What were the severe obesity prevalences
↪  for men by age group?
Expected Answer Structure: [men20_39,
↪  men40_59, men60plus]
Additional Info: NA.

Question ID: NHANES_NCHS Data Brief No.
↪  508_q9
Q9: What were the severe obesity prevalences
↪  for women by age group?
Expected Answer Structure: [women20_39,
↪  women40_59, women60plus]
Additional Info: NA.


Available data files:
[Omitted for simplicity]

Available documentation files:
[Omitted for simplicity]

Instructions:
You are analyzing survey data to answer
↪  specific questions. Follow this workflow:

1. UNDERSTAND THE TASK
   - Use `prompt` tool to review all
   ↪  questions, expected answer structures,
   ↪  and additional info
   - Identify which data files and
   ↪  documentation you need

2. GATHER INFORMATION
   - Use `retriever` or `search_doc` to find
   ↪  relevant documentation about the
   ↪  dataset
   - Use `read_doc` to read specific
   ↪  documentation files when needed
   - Record key information (column names,
   ↪  meanings, important variables) using
   ↪  `notes` tool
   - Do NOT rely on your own knowledge -
   ↪  always seek information from the
   ↪  documentation

3. ANALYZE DATA
   - Write Python code to load and analyze
   ↪  the data
   - Apply sampling weights if required by
   ↪  the survey design

4. FORMAT AND SAVE ANSWERS
   - Format your answer EXACTLY as specified
   ↪  in the "Expected Answer Structure"
   - Pay attention to unit requirements
   ↪  (e.g., percentages, counts, thousands,
   ↪  millions, billions)
   - All numbers should have ONE decimal
   ↪  place (e.g., 45.3, not 45 or 45.27)
   - Use `answer` tool with action='add' to
   ↪  save each answer with its question ID
   - Use `save_code` to store the analysis
   ↪  code for each block after finishing
   ↪  the block

5. WORKING STRATEGY
   - You can answer questions one by one - no
   ↪  need to solve all at once
   - Use `answer` tool with action='view' to
   ↪  check answers you've already saved
   - Use `notes` tool with action='list' to
   ↪  review your documented findings
```

Step 2

Code Action

```
# Step 2: Gather documentation info: search
↪  for BMI and weight variables in BMX and
↪  DEMO docs

bmx_doc = "benchmark/NHANES/docs/Examinatio
↪  n/BMX_L.pdf"
demo_doc = "benchmark/NHANES/docs/Demograph
↪  ics/DEMO_L.pdf"
exam_overview_doc = "benchmark/NHANES/docs/
↪  Examination/August 2021-August 2023
↪  Examination Data - Continuous NHANES.pdf"
demo_overview_doc = "benchmark/NHANES/docs/
↪  Demographics/August 2021-August 2023
↪  Demographics Data - Continuous
↪  NHANES.pdf"

# Search for BMI variable in BMX_L.pdf
bmi_search = search_doc(doc_path=bmx_doc,
↪  keyword="BMXBMI", max_matches=5,
↪  context_chars=200)
print("BMX BMI search:\
", bmi_search)

# Search for what BMI means text
bmi_text_search =
↪  search_doc(doc_path=bmx_doc,
↪  keyword="Body Mass Index", max_matches=5,
↪  context_chars=200)
print("\
BMX Body Mass Index search:\
", bmi_text_search)

# Search for exam weights in examination
↪  overview (expect WTMEC...)
exam_weight_search =
↪  search_doc(doc_path=exam_overview_doc,
↪  keyword="weight", max_matches=10,
↪  context_chars=200)
print("\
Exam overview weight search:\
", exam_weight_search)

# Search for MEC weight variable name
↪  specifically
mec_search =
↪  search_doc(doc_path=exam_overview_doc,
↪  keyword="WTMEC", max_matches=10,
↪  context_chars=200)
print("\
Exam overview WTMEC search:\
", mec_search)
```

```
# Search demographic variables: age, sex,
↪  education in DEMO_L.pdf
age_search = search_doc(doc_path=demo_doc,
↪  keyword="RIDAGEYR", max_matches=5,
↪  context_chars=200)
print("\
DEMO age search:\
", age_search)

sex_search = search_doc(doc_path=demo_doc,
↪  keyword="RIAGENDR", max_matches=5,
↪  context_chars=200)
print("\
DEMO sex search:\
", sex_search)

educ_search = search_doc(doc_path=demo_doc,
↪  keyword="DMDEDUC2", max_matches=5,
↪  context_chars=250)
print("\
DEMO education search:\
", educ_search)

# Also search the demographics overview for
↪  weights guidance
demo_weight_search =
↪  search_doc(doc_path=demo_overview_doc,
↪  keyword="weight", max_matches=10,
↪  context_chars=200)
print("\
Demographics overview weight search:\
", demo_weight_search)
```

Observations

```
Execution logs:
BMX BMI search:
 [benchmark/NHANES/docs/Examination/BMX_L.p⌋
 ↪  df] Found 3 matches for
 ↪  'BMXBMI'.
Page 3: ...Body Mass Index (BMXBMI) : Body
↪  Mass Index (BMI) w as calculated as
↪  weight in kilogr ams divided b y height
↪  in meters squared, and then rounded to
↪  one decimal place. BMI Category -
↪  Children/Adolescents (BMDBMIC) : This v
↪  ar...
Page 14: ...Variable Name: BMXBMI SAS Label:
↪  Body Mass Index (kg/m**2) English Text:
↪  Body Mass Index (kg/m**2) Target: Both
↪  males and females 2 YEARS - 150
↪  YEARSBMXBMI - Body Mass Index (kg/m**2)
↪  Code or Value Value Description Count...
Page 14: ...Variable Name: BMXBMI SAS Label:
↪  Body Mass Index (kg/m**2) English Text:
↪  Body Mass Index (kg/m**2) Target: Both
↪  males and females 2 YEARS - 150
↪  YEARSBMXBMI - Body Mass Index (kg/m**2)
↪  Code or Value Value Description Count
↪  Cumulative Skip to Item 11.1 to 74.8
↪  Range of V alues 8471 8471 . Missing 389
↪  8860...

BMX Body Mass Index search:
 [benchmark/NHANES/docs/Examination/BMX_L.p⌋
 ↪  df] Found 5 matches for 'Body Mass
 ↪  Index'.
Page 3: ...Body Mass Index (BMXBMI) : Body
↪  Mass Index (BMI) w as calculated as
↪  weight in kilogr ams divided b y height
↪  in meters squared, and then rounded to
↪  one decimal place. BMI Category -
↪  Children/Adolescents (BMDBMIC) : T...
Page 3: ...Body Mass Index (BMXBMI) : Body
↪  Mass Index (BMI) w as calculated as
↪  weight in kilogr ams divided b y height
↪  in meters squared, and then rounded to
↪  one decimal place. BMI Category -
↪  Children/Adolescents (BMDBMIC) : This v
↪  ariable w as created ...
Page 14: ...Variable Name: BMXBMI SAS Label:
↪  Body Mass Index (kg/m**2) English Text:
↪  Body Mass Index (kg/m**2) Target: Both
↪  males and females 2 YEARS - 150
↪  YEARSBMXBMI - Body Mass Index (kg/m**2)
↪  Code or Value Value Description Count
↪  Cumulative Skip to Item 11...
Page 14: ...Variable Name: BMXBMI SAS Label:
↪  Body Mass Index (kg/m**2) English Text:
↪  Body Mass Index (kg/m**2) Target: Both
↪  males and females 2 YEARS - 150
↪  YEARSBMXBMI - Body Mass Index (kg/m**2)
↪  Code or Value Value Description Count
↪  Cumulative Skip to Item 11.1 to 74.8
↪  Range of V alues 8471 8471 . ...
Page 14: ...Variable Name: BMXBMI SAS Label:
↪  Body Mass Index (kg/m**2) English Text:
↪  Body Mass Index (kg/m**2) Target: Both
↪  males and females 2 YEARS - 150
↪  YEARSBMXBMI - Body Mass Index (kg/m**2)
↪  Code or Value Value Description Count
↪  Cumulative Skip to Item 11.1 to 74.8
↪  Range of V alues 8471 8471 . Missing 389
↪  8860...

Exam overview weight search:
 [benchmark/NHANES/docs/Examination/August
 ↪  2021-August 2023 Examination Data -
 ↪  Continuous NHANES.pdf] No matches for
 ↪  'weight'.

Exam overview WTMEC search:
 [benchmark/NHANES/docs/Examination/August
 ↪  2021-August 2023 Examination Data -
 ↪  Continuous NHANES.pdf] No matches for
 ↪  'WTMEC'.

DEMO age search:
 [benchmark/NHANES/docs/Demographics/DEMO_L⌋
 ↪  .pdf] Found 4 matches for
 ↪  'RIDAGEYR'.
Page 2: ...entifies whether a participant w
↪  as both interviewed at home or b y
↪  telephone and examined in the mobile
↪  examination center (MEC), or w as only
↪  interviewed but nev er participated in
↪  the examination. RIDAGEYR: Age in y ears
↪  at the time of the screening interview
↪  for surv ey participants between the
↪  ages of 1 and 79 y ears of age. Due to
↪  disclosure concerns, all responses of
↪  participants aged 80 y ears and...
```

```
Page 2: ...sure concerns, all responses of
↪  participants aged 80 y ears and older
↪  are coded as "80. " In NHANES A ugust
↪  2021- August 2023, the weighted mean age
↪  for participants 80 y ears and older is
↪  85 y ears. RIDAGEYR w as calculated
↪  based on the participant 's date of
↪  birth. In r are cases, if the actual
↪  date of birth w as missing but the
↪  participant 's age in y ears w as pro
↪  vided, the reported age w as used. RI...
Page 10: ...Variable Name: RIDAGEYR SAS
↪  Label: Age in y ears at screening English
↪  Text: Age in y ears of the participant at
↪  the time of screening. Individuals 80
↪  and o ver are top-coded at 80 y ears of
↪  age. Target: Both males and femal...
Page 10: ...ars at screening English Text:
↪  Age in y ears of the participant at the
↪  time of screening. Individuals 80 and o
↪  ver are top-coded at 80 y ears of age.
↪  Target: Both males and females 0 YEARS -
↪  150 YEARSRIDAGEYR - Age in years at
↪  screening Code or Value Value
↪  Description Count Cumulative Skip to
↪  Item 0 to 79 Range of V alues 11408 11408
↪  80 80 years of age and o ver 525 11933 .
↪  Missing 0 11933...

DEMO sex search:
 [benchmark/NHANES/docs/Demographics/DEMO_L⌋
 ↪  .pdf] Found 2 matches for
 ↪  'RIAGENDR'.
Page 9: ...Variable Name: RIAGENDR SAS Label:
↪  Gender English Text: Gender of the
↪  participant. Target: Both males and
↪  females 0 YEARS - 150 YEARSRIAGENDR -
↪  Gender Code or Value Value Description
↪  Count Cumulative Skip to Item 1 Male
↪  557...
Page 9: ...Variable Name: RIAGENDR SAS Label:
↪  Gender English Text: Gender of the
↪  participant. Target: Both males and
↪  females 0 YEARS - 150 YEARSRIAGENDR -
↪  Gender Code or Value Value Description
↪  Count Cumulative Skip to Item 1 Male
↪  5575 5575 2 Female 6358 11933 . Missing
↪  0 11933...

DEMO education search:
 [benchmark/NHANES/docs/Demographics/DEMO_L⌋
 ↪  .pdf] Found 5 matches for
 ↪  'DMDEDUC2'.
Page 3: ...ded into 6 categories: 1 = Less
↪  than 1 y ear, 2 = 1 to 4 y ears, 3 = 5 to
↪  9 y ears, 4 = 10 to 14 y ears, 5 = 15 to
↪  19 y ears, and 6 = 20 y ear or more. F or
↪  participants who did not report the month
↪  of their arriv al, a v alue of 7 (July) w
↪  as used. DMDEDUC2: the highest gr ade or
↪  lev el of education completed b y adults
↪  20 y ears and older . DMDMARTZ: The
↪  marital status question w as ask ed of
↪  people 14 y ears of age and older . Due
↪  to disclosure risk, marital status is
↪  only released for people 20 y ea...
Page 17: ...SAS Label: Country of birth
↪  English Text: Country of birth Target:
↪  Both males and females 0 YEARS - 150
↪  YEARSDMDBORN4 - Country of birth Code or
↪  Value Value Description Count Cumulative
↪  Skip to Item 1 Born in 50 US states or W
↪  ashington, 10039 10039 DMDEDUC2 2 Others
↪  1875 11914 77 Refused 0 11914 DMDEDUC2
↪  99 Don't know 0 11914 DMDEDUC2 . Missing
↪  19 11933...
Page 17: ...y of birth Target: Both males
↪  and females 0 YEARS - 150 YEARSDMDBORN4 -
↪  Country of birth Code or Value Value
↪  Description Count Cumulative Skip to Item
↪  1 Born in 50 US states or W ashington,
↪  10039 10039 DMDEDUC2 2 Others 1875 11914
↪  77 Refused 0 11914 DMDEDUC2 99 Don't know
↪  0 11914 DMDEDUC2 . Missing 19 11933...
Page 17: ...nd females 0 YEARS - 150
↪  YEARSDMDBORN4 - Country of birth Code or
↪  Value Value Description Count Cumulative
↪  Skip to Item 1 Born in 50 US states or W
↪  ashington, 10039 10039 DMDEDUC2 2 Others
↪  1875 11914 77 Refused 0 11914 DMDEDUC2
↪  99 Don't know 0 11914 DMDEDUC2 . Missing
↪  19 11933...
Page 19: ...Variable Name: DMDEDUC2 SAS
↪  Label: Education lev el - Adults 20+
↪  English Text: What is the highest gr ade
↪  or lev el of school {y ou ha ve/SP has}
↪  completed or the highest degree {y ou ha
↪  ve/s/he has} receiv ed? English
↪  Instructions: HAND CARD DMQ1 READ HAND
↪  CARD CA TEGORIE...

Demographics overview weight search:
 [benchmark/NHANES/docs/Demographics/August
 ↪  2021-August 2023 Demographics Data -
 ↪  Continuous NHANES.pdf] Found 1 matches
 ↪  for 'weight'.
Page 1: ...rview of Sample Design,
↪  Nonresponse Bias Assessment, and
↪  Analytic Guidelines SAS Universal Viewer
↪  Data User Agreement Data File Name Doc
↪  File Data File Date Published
↪  Demographic Variables and Sample Weights
↪  DEMO_L Doc DEMO_L Data [XPT - 2.5 MB]
↪  September 2024...
Last output from code snippet:
None
```

Step 3

Code Action

```
# Step 2 (continued): find weight variables
↪  in DEMO_L documentation
demo_doc = "benchmark/NHANES/docs/Demograph⌋
↪  ics/DEMO_L.pdf"

wts_mec_search =
↪  search_doc(doc_path=demo_doc,
↪  keyword="WTMEC", max_matches=10,
↪  context_chars=240)
print("DEMO_L search for WTMEC:\
", wts_mec_search)
```

```
wts_int_search =
↪  search_doc(doc_path=demo_doc,
↪  keyword="WTINT", max_matches=10,
↪  context_chars=240)
print("\
DEMO_L search for WTINT:\
", wts_int_search)

wts_overall_search =
↪  search_doc(doc_path=demo_doc,
↪  keyword="weight", max_matches=10,
↪  context_chars=240)
print("\
DEMO_L search for \
```

Observations

```
Execution logs:
DEMO_L search for WTMEC:
 [benchmark/NHANES/docs/Demographics/DEMO_L⌋
 ↪  .pdf] Found 3 matches for
 ↪  'WTMEC'.
Page 5: ...protect confidentialit y and to
↪  reduce disclosure risk and are described
↪  in NHANES Analytic Guidelines . Analysts
↪  should review the Guidelines carefully
↪  prior to analyzing surv ey data. Sample
↪  Weights: The 2- year sample weights
↪  (WTINT2YR, WTMEC2YR, depending on the
↪  analysis) should be used for all NHANES
↪  A ugust 2021- August 2023 analyses.
↪  Please refer to the NHANES Analytic
↪  Guidelines and the on-line NHANES T
↪  utorial for further details on the use
↪  of sample weights and other ana...
Page 29: ...Variable Name: WTMEC2YR SAS
↪  Label: Full sample 2- year MEC exam
↪  weight English Text: Full sample 2 y ear
↪  MEC exam weight. Target: Both males and
↪  females 0 YEARS - 150 YEARSWTMEC2YR -
↪  Full sample 2-year MEC exam weight Code
↪  or Value Value Description Count Cumu...
Page 29: ...Variable Name: WTMEC2YR SAS
↪  Label: Full sample 2- year MEC exam
↪  weight English Text: Full sample 2 y ear
↪  MEC exam weight. Target: Both males and
↪  females 0 YEARS - 150 YEARSWTMEC2YR -
↪  Full sample 2-year MEC exam weight Code
↪  or Value Value Description Count
↪  Cumulative Skip to Item 4581.595095 to
↪  227108.296958 Range of V alues 8860 8860
↪  0 Not MEC Examined 3073 11933 . Missing
↪  0 11933...

DEMO_L search for WTINT:
 [benchmark/NHANES/docs/Demographics/DEMO_L⌋
 ↪  .pdf] Found 3 matches for
 ↪  'WTINT'.
Page 5: ...e used to protect confidentialit
↪  y and to reduce disclosure risk and are
↪  described in NHANES Analytic Guidelines .
↪  Analysts should review the Guidelines
↪  carefully prior to analyzing surv ey
↪  data. Sample Weights: The 2- year sample
↪  weights (WTINT2YR, WTMEC2YR, depending
↪  on the analysis) should be used for all
↪  NHANES A ugust 2021- August 2023
↪  analyses. Please refer to the NHANES
↪  Analytic Guidelines and the on-line
↪  NHANES T utorial for further details on
↪  the use of sample weights and...
Page 28: ...Variable Name: WTINT2YR SAS
↪  Label: Full sample 2- year interview
↪  weight English Text: Full sample 2 y ear
↪  interview weight. Target: Both males and
↪  females 0 YEARS - 150 YEARSWTINT2YR -
↪  Full sample 2-year interview weight Code
↪  or Value Value Description Count C...
Page 28: ...Variable Name: WTINT2YR SAS
↪  Label: Full sample 2- year interview
↪  weight English Text: Full sample 2 y ear
↪  interview weight. Target: Both males and
↪  females 0 YEARS - 150 YEARSWTINT2YR -
↪  Full sample 2-year interview weight Code
↪  or Value Value Description Count
↪  Cumulative Skip to Item 4584.463196 to
↪  170968.343177 Range of V alues 11933
↪  11933 . Missing 0 11933...

DEMO_L search for 'weight':
 [benchmark/NHANES/docs/Demographics/DEMO_L⌋
 ↪  .pdf] Found 10 matches for
 ↪  'weight'.
Page 1: ...cific to this cy cle, please see
↪  the "Analytic Notes" section below .
↪  This demogr aphics file pro vides
↪  information on the following topics:
↪  Survey participant 's household
↪  interview and examination status;
↪  Interview and examination sample weights;
↪  Masked variance units; The six -month
↪  time period when the examination w as
↪  performed; Pregnancy status; The ratio
↪  of family income to po verty guidelines;
↪  Total number of people in the Household;
↪  Demogr aphic information about the
↪  house...
Page 2: ...of the screening interview for
↪  surv ey participants between the ages of
↪  1 and 79 y ears of age. Due to disclosure
↪  concerns, all responses of participants
↪  aged 80 y ears and older are coded as
↪  "80. " In NHANES A ugust 2021- August
↪  2023, the weighted mean age for
↪  participants 80 y ears and older is 85 y
↪  ears. RIDAGEYR w as calculated based on
↪  the participant 's date of birth. In r
↪  are cases, if the actual date of birth w
↪  as missing but the participant 's age in
↪  y ears w as pro vided,...
```

```
Page 5: ...sample design variance units.
↪  MVUs are used to protect confidentialit
↪  y and to reduce disclosure risk and are
↪  described in NHANES Analytic Guidelines .
↪  Analysts should review the Guidelines
↪  carefully prior to analyzing surv ey
↪  data. Sample Weights: The 2- year sample
↪  weights (WTINT2YR, WTMEC2YR, depending
↪  on the analysis) should be used for all
↪  NHANES A ugust 2021- August 2023
↪  analyses. Please refer to the NHANES
↪  Analytic Guidelines and the on-line
↪  NHANES T utorial for further deta...
Page 5: .... MVUs are used to protect
↪  confidentialit y and to reduce disclosure
↪  risk and are described in NHANES
↪  Analytic Guidelines . Analysts should
↪  review the Guidelines carefully prior to
↪  analyzing surv ey data. Sample Weights:
↪  The 2- year sample weights (WTINT2YR,
↪  WTMEC2YR, depending on the analysis)
↪  should be used for all NHANES A ugust
↪  2021- August 2023 analyses. Please refer
↪  to the NHANES Analytic Guidelines and
↪  the on-line NHANES T utorial for further
↪  details on the use of sample wei...
Page 5: ...ghts (WTINT2YR, WTMEC2YR,
↪  depending on the analysis) should be used
↪  for all NHANES A ugust 2021- August 2023
↪  analyses. Please refer to the NHANES
↪  Analytic Guidelines and the on-line
↪  NHANES T utorial for further details on
↪  the use of sample weights and other
↪  analytic issues. Disclosure risks and
↪  issues pertaining to confidentialit y
↪  protection prev ent NCHS from releasing
↪  all the NHANES demogr aphic v ariables
↪  publicly . Additional information and
↪  instructions for requesting use of ...
Page 28: ...Variable Name: WTINT2YR SAS
↪  Label: Full sample 2- year interview
↪  weight English Text: Full sample 2 y ear
↪  interview weight. Target: Both males and
↪  females 0 YEARS - 150 YEARSWTINT2YR -
↪  Full sample 2-year interview weight Code
↪  or Value Value Description Count
↪  Cumulative Skip to Item 4584.463196 to
↪  170968.343177...
Page 28: ...Variable Name: WTINT2YR SAS
↪  Label: Full sample 2- year interview
↪  weight English Text: Full sample 2 y ear
↪  interview weight. Target: Both males and
↪  females 0 YEARS - 150 YEARSWTINT2YR -
↪  Full sample 2-year interview weight Code
↪  or Value Value Description Count
↪  Cumulative Skip to Item 4584.463196 to
↪  170968.343177 Range of V alues 11933
↪  11933 . Missing 0 11933...
Page 28: ...Variable Name: WTINT2YR SAS
↪  Label: Full sample 2- year interview
↪  weight English Text: Full sample 2 y ear
↪  interview weight. Target: Both males and
↪  females 0 YEARS - 150 YEARSWTINT2YR -
↪  Full sample 2-year interview weight Code
↪  or Value Value Description Count
↪  Cumulative Skip to Item 4584.463196 to
↪  170968.343177 Range of V alues 11933
↪  11933 . Missing 0 11933...
```

```
Page 29: ...Variable Name: WTMEC2YR SAS
↪  Label: Full sample 2- year MEC exam
↪  weight English Text: Full sample 2 y ear
↪  MEC exam weight. Target: Both males and
↪  females 0 YEARS - 150 YEARSWTMEC2YR -
↪  Full sample 2-year MEC exam weight Code
↪  or Value Value Description Count
↪  Cumulative Skip to Item 4581.595095 to
↪  227108.296958 R...
Page 29: ...Variable Name: WTMEC2YR SAS
↪  Label: Full sample 2- year MEC exam
↪  weight English Text: Full sample 2 y ear
↪  MEC exam weight. Target: Both males and
↪  females 0 YEARS - 150 YEARSWTMEC2YR -
↪  Full sample 2-year MEC exam weight Code
↪  or Value Value Description Count
↪  Cumulative Skip to Item 4581.595095 to
↪  227108.296958 Range of V alues 8860 8860
↪  0 Not MEC Examined 3073 ...

DEMO_L page 19 (likely DMDEDUC2 details):
 [benchmark/NHANES/docs/Demographics/DEMO_L⌋
 ↪  .pdf] Pages: Page
 ↪  19
Page 19: Variable Name: DMDEDUC2 SAS Label:
↪  Education lev el - Adults 20+ English
↪  Text: What is the highest gr ade or lev
↪  el of school {y ou ha ve/SP has}
↪  completed or the highest degree {y ou ha
↪  ve/s/he has} receiv ed? English
↪  Instructions: HAND CARD DMQ1 READ HAND
↪  CARD CA TEGORIES IF NECES SARY ENTER
↪  HIGHEST LEVEL OF SCHOOL Target: Both
↪  males and females 20 YEARS - 150
↪  YEARSDMDEDUC2 - Education level - Adults
↪  20+ Code or ValueValue Description Count
↪  Cumulative Skip to Item 1 Less than 9th
↪  gr ade 373 373 2 9-11th gr ade (Includes
↪  12th gr ade with no diploma)666 1039 3
↪  High school gr aduate/GED or equiv alent
↪  1749 2788 4 Some college or AA degree
↪  2370 5158 5 College gr aduate or abo ve
↪  2625 7783 7 Refused 0 7783 9 Don't know
↪  11 7794 . Missing 4139 11933
Last output from code snippet:
None
```

Step 4

Code Action

```
# Step 2 (additional): check for pregnancy
↪  variable in DEMO_L to see if exclusions
↪  are required
demo_doc = "benchmark/NHANES/docs/Demograph⌋
↪  ics/DEMO_L.pdf"
preg_search = search_doc(doc_path=demo_doc,
↪  keyword="preg", max_matches=10,
↪  context_chars=200)
print("DEMO_L pregnancy search:\
", preg_search)

# Record key notes about variables and
↪  weights
```

```
notes(action="add", content="Use BMX_L for
↪  BMI: BMXBMI is Body Mass Index (kg/m**2),
↪  rounded to one decimal. Use DEMO_L for
↪  demographics: RIDAGEYR (age in years),
↪  RIAGENDR (1=Male, 2=Female), DMDEDUC2
↪  (education for adults 20+). Use WTMEC2YR
↪  for exam-based analyses per DEMO_L
↪  (2-year MEC exam weight).", index=0)
notes(action="add", content="Education
↪  grouping for adults 20+: hs_or_less =
↪  DMDEDUC2 in [1,2,3]; some_college = 4;
↪  bachelors_plus = 5. Exclude DMDEDUC2 in
↪  [7,9] or missing for education-specific
↪  estimates.", index=0)
notes(action="add", content="Age groups:
↪  20-39, 40-59, 60+ using RIDAGEYR (80+
↪  top-coded to 80, acceptable).", index=0)
print(notes(action="list", content="",
↪  index=0))
```

Observations

```
Execution logs:
DEMO_L pregnancy search:
 [benchmark/NHANES/docs/Demographics/DEMO_L⌋
 ↪  .pdf] Found 10 matches for
 ↪  'preg'.
Page 1: ...ics: Survey participant 's
↪  household interview and examination
↪  status; Interview and examination sample
↪  weights; Masked variance units; The six
↪  -month time period when the examination w
↪  as performed; Pregnancy status; The
↪  ratio of family income to po verty
↪  guidelines; Total number of people in
↪  the Household; Demogr aphic information
↪  about the household reference person;
↪  and Other selected demogr aphic...
Page 3: ...d for people 20 y ears of age and
↪  older , and recoded from the original 6
↪  categories to 3 categories (1 =
↪  Married/Living with partner; 2 =
↪  Widowed/Div orced/Separ ated; 3 = Nev er
↪  married). RIDEXPRG: Pregnancy status at
↪  the time of the health examination w as
↪  ascertained for females 8-59 y ears of
↪  age. Due to disclosure risk, pregnancy
↪  status is released only for women 20-44
↪  y ears of age. RIDEXPRG v ...
Page 3: ...rtner; 2 = Widowed/Div orced/Separ
↪  ated; 3 = Nev er married). RIDEXPRG:
↪  Pregnancy status at the time of the
↪  health examination w as ascertained for
↪  females 8-59 y ears of age. Due to
↪  disclosure risk, pregnancy status is
↪  released only for women 20-44 y ears of
↪  age. RIDEXPRG v alues are based on self
↪  -reported pregnancy status and urine
↪  pregnancy test results. P eople who
↪  reported they were pregnant at ...
Page 3: ...alth examination w as ascertained
↪  for females 8-59 y ears of age. Due to
↪  disclosure risk, pregnancy status is
↪  released only for women 20-44 y ears of
↪  age. RIDEXPRG v alues are based on self
↪  -reported pregnancy status and urine
↪  pregnancy test results. P eople who
↪  reported they were pregnant at the time
↪  of exam were assumed to be pregnant
↪  (RIDEXPRG=1). Those who reported they
↪  were not pregnant or did no...
Page 3: ...tained for females 8-59 y ears of
↪  age. Due to disclosure risk, pregnancy
↪  status is released only for women 20-44
↪  y ears of age. RIDEXPRG v alues are
↪  based on self -reported pregnancy status
↪  and urine pregnancy test results. P
↪  eople who reported they were pregnant at
↪  the time of exam were assumed to be
↪  pregnant (RIDEXPRG=1). Those who
↪  reported they were not pregnant or did
↪  not know their pregnancy stat...
Page 3: ...e risk, pregnancy status is
↪  released only for women 20-44 y ears of
↪  age. RIDEXPRG v alues are based on self
↪  -reported pregnancy status and urine
↪  pregnancy test results. P eople who
↪  reported they were pregnant at the time
↪  of exam were assumed to be pregnant
↪  (RIDEXPRG=1). Those who reported they
↪  were not pregnant or did not know their
↪  pregnancy status were further classified
↪  based on the results of the ...
Page 3: ...men 20-44 y ears of age. RIDEXPRG
↪  v alues are based on self -reported
↪  pregnancy status and urine pregnancy
↪  test results. P eople who reported they
↪  were pregnant at the time of exam were
↪  assumed to be pregnant (RIDEXPRG=1).
↪  Those who reported they were not
↪  pregnant or did not know their pregnancy
↪  status were further classified based on
↪  the results of the urine pregnancy test.
↪  If the respondent reported...
Page 3: ...elf -reported pregnancy status and
↪  urine pregnancy test results. P eople
↪  who reported they were pregnant at the
↪  time of exam were assumed to be pregnant
↪  (RIDEXPRG=1). Those who reported they
↪  were not pregnant or did not know their
↪  pregnancy status were further classified
↪  based on the results of the urine
↪  pregnancy test. If the respondent
↪  reported "no" or "don' t know" and the
↪  urine test result w as po...
Page 3: ...and urine pregnancy test results.
↪  P eople who reported they were pregnant
↪  at the time of exam were assumed to be
↪  pregnant (RIDEXPRG=1). Those who
↪  reported they were not pregnant or did
↪  not know their pregnancy status were
↪  further classified based on the results
↪  of the urine pregnancy test. If the
↪  respondent reported "no" or "don' t
↪  know" and the urine test result w as
↪  positiv e, the respondent w as co...
```

```
Page 3: ...t the time of exam were assumed to
↪  be pregnant (RIDEXPRG=1). Those who
↪  reported they were not pregnant or did
↪  not know their pregnancy status were
↪  further classified based on the results
↪  of the urine pregnancy test. If the
↪  respondent reported "no" or "don' t
↪  know" and the urine test result w as
↪  positiv e, the respondent w as coded as
↪  pregnant (RIDEXPRG=1). If the respondent
↪  reported "no" and the urine...
Notes:
1. Use BMX_L for BMI: BMXBMI is Body Mass
↪  Index (kg/m**2), rounded to one decimal.
↪  Use DEMO_L for demog...
2. Education grouping for adults 20+:
↪  hs_or_less = DMDEDUC2 in [1,2,3];
↪  some_college = 4; bachelors_plu...
3. Age groups: 20-39, 40-59, 60+ using
↪  RIDAGEYR (80+ top-coded to 80,
↪  acceptable).
Last output from code snippet:
None
```

Step 5

Code Action

```
# Check BMX documentation for any mention of
↪  excluding pregnant women
bmx_doc = "benchmark/NHANES/docs/Examinatio⌋
↪  n/BMX_L.pdf"
preg_search_bmx =
↪  search_doc(doc_path=bmx_doc,
↪  keyword="preg", max_matches=10,
↪  context_chars=200)
print("BMX_L pregnancy search:\
", preg_search_bmx)

# Proceed to data analysis
import pandas as pd
import numpy as np

# Load data
demo_path = "benchmark/NHANES/data/Demograp⌋
↪  hics/DEMO_L.xpt"
bmx_path = "benchmark/NHANES/data/Examinati⌋
↪  on/BMX_L.xpt"

demo = pd.read_sas(demo_path,
↪  format="xport")
bmx = pd.read_sas(bmx_path, format="xport")

# Keep only necessary columns to reduce
↪  memory
demo_cols = ["SEQN", "RIDAGEYR", "RIAGENDR",
↪  "DMDEDUC2", "WTMEC2YR"]
demo = demo[demo_cols]
bmx_cols = ["SEQN", "BMXBMI"]
bmx = bmx[bmx_cols]

# Merge
df = demo.merge(bmx, on="SEQN", how="inner")

# Convert to numeric (ensure dtypes)
for col in ["RIDAGEYR", "RIAGENDR",
↪  "DMDEDUC2", "WTMEC2YR", "BMXBMI"]:
    df[col] = pd.to_numeric(df[col],
    ↪  errors="coerce")

# Restrict to adults 20+ and with valid
↪  weight and BMI
adults = df[(df["RIDAGEYR"] >= 20) &
↪  (df["WTMEC2YR"] > 0) &
↪  df["BMXBMI"].notna()]

# Define obesity and severe obesity
adults = adults.copy()
adults["obese"] = (adults["BMXBMI"] >=
↪  30).astype(int)
adults["severe_obese"] = (adults["BMXBMI"]
↪  >= 40).astype(int)

# Age groups
def age_group(age):
    if age >= 20 and age <= 39:
        return "20-39"
    elif age >= 40 and age <= 59:
        return "40-59"
    else:
        return "60+"
adults["age_group"] =
↪  adults["RIDAGEYR"].apply(age_group)

# Education groups: hs_or_less=[1,2,3],
↪  some_college=4, bachelors_plus=5
def educ_group(x):
    if x in [1, 2, 3]:
        return "hs_or_less"
    elif x == 4:
        return "some_college"
    elif x == 5:
        return "bachelors_plus"
    else:
        return np.nan
adults["educ_group"] =
↪  adults["DMDEDUC2"].apply(educ_group)

# Helper to compute weighted prevalence (%)
def weighted_pct(data, indicator_col,
↪  weight_col="WTMEC2YR"):
    w = data[weight_col]
    y = data[indicator_col]
    num = (w * y).sum()
    den = w.sum()
    if den == 0 or np.isnan(den):
        return np.nan
    return float(100.0 * num / den)

# Q1: overall prevalence of obesity among
↪  adults 20+
q1_val = weighted_pct(adults, "obese")
q1 = round(q1_val, 1) if pd.notna(q1_val)
↪  else np.nan
print("Q1 overall obesity prevalence:", q1)

# Q2: obesity prevalences for men and women
↪  [men, women]
men = adults[adults["RIAGENDR"] == 1]
women = adults[adults["RIAGENDR"] == 2]
q2_vals = [round(weighted_pct(men, "obese"),
↪  1), round(weighted_pct(women, "obese"),
↪  1)]
print("Q2 obesity men, women:", q2_vals)
```

```
# Q3: obesity by age groups [20-39, 40-59,
↪  60+]
q3_vals = []
for grp in ["20-39", "40-59", "60+"]:
    d = adults[adults["age_group"] == grp]
    q3_vals.append(round(weighted_pct(d,
    ↪  "obese"), 1))
print("Q3 obesity by age groups:", q3_vals)

# Q4: obesity by education level [hs_or_less,
↪  some_college, bachelors_plus]
educ_order = ["hs_or_less", "some_college",
↪  "bachelors_plus"]
q4_vals = []
for eg in educ_order:
    d = adults[adults["educ_group"] == eg]
    q4_vals.append(round(weighted_pct(d,
    ↪  "obese"), 1))
print("Q4 obesity by education level:",
↪  q4_vals)

# Q5: overall prevalence of severe obesity
↪  among adults
q5_val = weighted_pct(adults,
↪  "severe_obese")
q5 = round(q5_val, 1) if pd.notna(q5_val)
↪  else np.nan
print("Q5 overall severe obesity
↪  prevalence:", q5)

# Q6: severe obesity by age groups [20-39,
↪  40-59, 60+]
q6_vals = []
for grp in ["20-39", "40-59", "60+"]:
    d = adults[adults["age_group"] == grp]
    q6_vals.append(round(weighted_pct(d,
    ↪  "severe_obese"), 1))
print("Q6 severe obesity by age groups:",
↪  q6_vals)

# Q7: severe obesity prevalences for men and
↪  women [men, women]
q7_vals = [round(weighted_pct(men,
↪  "severe_obese"), 1),
↪  round(weighted_pct(women,
↪  "severe_obese"), 1)]
print("Q7 severe obesity men, women:",
↪  q7_vals)

# Q8: severe obesity prevalences for men by
↪  age group [men20-39, men40-59, men60+]
q8_vals = []
for grp in ["20-39", "40-59", "60+"]:
    d = men[men["age_group"] == grp]
    q8_vals.append(round(weighted_pct(d,
    ↪  "severe_obese"), 1))
print("Q8 severe obesity men by age:",
↪  q8_vals)

# Q9: severe obesity prevalences for women
↪  by age group [women20-39, women40-59,
↪  women60+]
q9_vals = []
for grp in ["20-39", "40-59", "60+"]:
    d = women[women["age_group"] == grp]
    q9_vals.append(round(weighted_pct(d,
    ↪  "severe_obese"), 1))
print("Q9 severe obesity women by age:",
↪  q9_vals)

# Save answers
ans1 = answer(action="add",
↪  q_id="NHANES_NCHS Data Brief No. 508_q1",
↪  answer=str(q1))
ans2 = answer(action="add",
↪  q_id="NHANES_NCHS Data Brief No. 508_q2",
↪  answer=[q2_vals[0], q2_vals[1]])
ans3 = answer(action="add",
↪  q_id="NHANES_NCHS Data Brief No. 508_q3",
↪  answer=[q3_vals[0], q3_vals[1],
↪  q3_vals[2]])
ans4 = answer(action="add",
↪  q_id="NHANES_NCHS Data Brief No. 508_q4",
↪  answer=[q4_vals[0], q4_vals[1],
↪  q4_vals[2]])
ans5 = answer(action="add",
↪  q_id="NHANES_NCHS Data Brief No. 508_q5",
↪  answer=str(q5))
ans6 = answer(action="add",
↪  q_id="NHANES_NCHS Data Brief No. 508_q6",
↪  answer=[q6_vals[0], q6_vals[1],
↪  q6_vals[2]])
ans7 = answer(action="add",
↪  q_id="NHANES_NCHS Data Brief No. 508_q7",
↪  answer=[q7_vals[0], q7_vals[1]])
ans8 = answer(action="add",
↪  q_id="NHANES_NCHS Data Brief No. 508_q8",
↪  answer=[q8_vals[0], q8_vals[1],
↪  q8_vals[2]])
ans9 = answer(action="add",
↪  q_id="NHANES_NCHS Data Brief No. 508_q9",
↪  answer=[q9_vals[0], q9_vals[1],
↪  q9_vals[2]])

print(ans1, ans2, ans3, ans4, ans5, ans6,
↪  ans7, ans8, ans9)

# Save the analysis code
save_code("""
Loaded DEMO_L and BMX_L, merged by SEQN.
↪  Filtered adults 20+ with valid BMI and
↪  WTMEC2YR > 0.
Defined obesity (BMXBMI >= 30) and severe
↪  obesity (BMXBMI >= 40).
Constructed age groups (20-39, 40-59, 60+)
↪  and education groups (hs_or_less=[1,2,3],
↪  some_college=4, bachelors_plus=5).
Computed weighted prevalences using WTMEC2YR
↪  for overall and subgroups, expressed as
↪  percentages with one decimal place.
Recorded answers using the answer tool.
""")
```

Observations

```
Execution logs:
BMX_L pregnancy search:
 [benchmark/NHANES/docs/Examination/BMX_L.p⌋
 ↪  df] Found 3 matches for
 ↪  'preg'.
```

```
Page 2: ... considered identifiable and pose
↪  a disclosure concern. The body weight
↪  data for participants who had limb
↪  amputations were set to "missing. " This
↪  data file includes body measures for
↪  women who were pregnant at the time of
↪  their health examination. Pregnancy
↪  status at the time of the health
↪  examination is indicated b y the
↪  variable, RIDEXPRG, in the Demogr aphic
↪  data file. RIDEXPRG v alues are report...
Page 2: ...cern. The body weight data for
↪  participants who had limb amputations
↪  were set to "missing. " This data file
↪  includes body measures for women who
↪  were pregnant at the time of their
↪  health examination. Pregnancy status at
↪  the time of the health examination is
↪  indicated b y the variable, RIDEXPRG, in
↪  the Demogr aphic data file. RIDEXPRG v
↪  alues are reported for women 20-44 y
↪  ears of age. RIDEXPRG for sev...
Page 2: ...status at the time of the health
↪  examination is indicated b y the
↪  variable, RIDEXPRG, in the Demogr aphic
↪  data file. RIDEXPRG v alues are reported
↪  for women 20-44 y ears of age. RIDEXPRG
↪  for sev eral pregnant women who were
↪  outside of this age range are not
↪  reported due to disclosure concerns. The
↪  body measures data for these
↪  participants are also not reported. The
↪  NHANES Anthropometry Procedures Manu...
Q1 overall obesity prevalence: 40.3
Q2 obesity men, women: [39.2, 41.3]
Q3 obesity by age groups: [35.6, 46.4, 38.9]
Q4 obesity by education level: [44.5, 45.2,
↪  31.6]
Q5 overall severe obesity prevalence: 9.4
Q6 severe obesity by age groups: [9.6, 11.9,
↪  6.6]
Q7 severe obesity men, women: [6.7, 12.1]
Q8 severe obesity men by age: [6.1, 9.2, 4.3]
Q9 severe obesity women by age: [13.0, 14.6,
↪  8.4]
Answer recorded for question NHANES_NCHS
↪  Data Brief No. 508_q1 Answer recorded
↪  for question NHANES_NCHS Data Brief No.
↪  508_q2 Answer recorded for question
↪  NHANES_NCHS Data Brief No. 508_q3 Answer
↪  recorded for question NHANES_NCHS Data
↪  Brief No. 508_q4 Answer recorded for
↪  question NHANES_NCHS Data Brief No.
↪  508_q5 Answer recorded for question
↪  NHANES_NCHS Data Brief No. 508_q6 Answer
↪  recorded for question NHANES_NCHS Data
↪  Brief No. 508_q7 Answer recorded for
↪  question NHANES_NCHS Data Brief No.
↪  508_q8 Answer recorded for question
↪  NHANES_NCHS Data Brief No. 508_q9
Last output from code snippet:
Code stored for block NHANES_NCHS Data Brief
↪  No. 508.
```

Step 6

Code Action

```
summary = {
    "NHANES_NCHS Data Brief No. 508_q1": q1,
    "NHANES_NCHS Data Brief No. 508_q2":
    ↪  q2_vals,
    "NHANES_NCHS Data Brief No. 508_q3":
    ↪  q3_vals,
    "NHANES_NCHS Data Brief No. 508_q4":
    ↪  q4_vals,
    "NHANES_NCHS Data Brief No. 508_q5": q5,
    "NHANES_NCHS Data Brief No. 508_q6":
    ↪  q6_vals,
    "NHANES_NCHS Data Brief No. 508_q7":
    ↪  q7_vals,
    "NHANES_NCHS Data Brief No. 508_q8":
    ↪  q8_vals,
    "NHANES_NCHS Data Brief No. 508_q9":
    ↪  q9_vals
}
final_answer(summary)
```

Observations

```
Execution logs:
Last output from code snippet:
{'NHANES_NCHS Data Brief No. 508_q1': 40.3,
↪  'NHANES_NCHS Data Brief No. 508_q2':
↪  [39.2, 41.3], 'NHANES_NCHS Data Brief No.
↪  508_q3': [35.6, 46.4, 38.9], 'NHANES_NCHS
↪  Data Brief No. 508_q4': [44.5, 45.2,
↪  31.6], 'NHANES_NCHS Data Brief No.
↪  508_q5': 9.4, 'NHANES_NCHS Data Brief No.
↪  508_q6': [9.6, 11.9, 6.6], 'NHANES_NCHS
↪  Data Brief No. 508_q7': [6.7, 12.1],
↪  'NHANES_NCHS Data Brief No. 508_q8':
↪  [6.1, 9.2, 4.3], 'NHANES_NCHS Data Brief
↪  No. 508_q9': [13.0, 14.6, 8.4]}
```