Supplementary Methods

Datasets. We used 4 datasets, each consisting of data files ("Data", Fig. 1B; Supplementary Datasets A-E) and metadata items (the human-provided products "Data file description" and "General description of dataset", Fig. 1B; Supplementary Data Descriptions A-E). (A) "Health Indicators" dataset[25]. A clean unweighted subset of CDC's Behavioral Risk Factor Surveillance System (BRFSS) 2015 annual dataset[41], downloaded from Kaggle[25]. It contains 253,680 survey responses each with 22 features related to diabetes and different health indicators, with no missing values. No change in the dataset was made; data-to-paper was provided with the csv file as downloaded from Kaggle. (B) "Social Network" dataset[26]. A directed graph of Twitter interactions among the 117th Congress members[26]. Two data files were provided to data-to-paper: (*i*) a csv file containing a list of directed unweighted edges, representing Twitter engagements among Congress members (downloaded from Stanford Network Analysis Project[49], with the weights removed), and (*ii*) a csv file containing the affiliations of each Congress member, including their Chamber, Party and State (downloaded from FRAC[50]). (C) "Infection" dataset[27]. A prospective, multicenter cohort study dataset collected between August 2020 and March 2022, involving hospital employees from ten healthcare networks in Eastern/Northern Switzerland. Two data files were provided: (a) *TimeToInfection.csv*, containing time-interval data on vaccination status, infection events, and related factors for each health worker, with a focus on day intervals and relevant variables such as age, sex, and BMI; (ii) *Symptoms.csv*, detailing symptoms for health workers who tested positive for SARS-CoV-2, including comorbidities, infection variant, and symptom count. Missing data is indicated with empty strings. (D) "Treatment policy" dataset (a test case to reproduce Saint-Fleur et al.[28]). A dataset on treatment and outcomes of non-vigorous infants admitted to the Neonatal Intensive Care Unit (NICU), before and after a change to treatment guidelines was implemented. As input to data-to-paper, the file downloaded from Saint-Fleur et al.[28] was converted into a csv file, with minor cleanups: converting column headers into alphanumeric names, converting string binary into integer binary, and removing the following irrelevant columns: 'RACE', 'RACE IN TWO CATEGORIES', 'ETHNICITY', 'Singleton /Multiple', 'Maternal Diabetes…', 'PRETERM VS TERM', 'ROUTINE RESUSCITATION…', 'Respiratory Support', 'Exposure to xrays', 'X-Ray finding' (without removing these columns, the "Data exploration" step of data-to-paper occasionally created too large output files leading to breaking the token limit of ChatGPT). (E) "Treatment Optimization" dataset (a test case to reproduce Shim et al.[29]). A dataset of 967 pediatric patients, which received mechanical ventilation after undergoing surgery, including an x-ray-based determination of the optimal tracheal tube intubation depth and a set of personalized patient attributes to be used in machine learning and formula-based

models to predict this optimal depth. As input for data-to-paper, we removed irrelevant columns, leaving only the ones used in the original study: 'tube', 'sex', 'age_c', 'ht', 'wt', 'tube_depth_G'. For datasets C and D, we further provided data-to-paper with the research goal of their respective original studies. Research goals and dataset descriptions have been formulated in an iterative and empirical process: We consulted with ChatGPT on best phrasing and terminologies, tested them in pilot runs, identified misunderstandings and vague or ill-defined statements, and adapted the descriptions accordingly. Dataset descriptions and file descriptions for all datasets, as well as the research goal for datasets D,E, are provided in Supplementary Data Descriptions A-E.

Devising data descriptions and research goals. The data descriptions and research goals provided to data-to-paper at the beginning of the research process are in the Supplementary Information. These prompts have been designed in an iterative and empirical process. We note that short, concise and well structured descriptions are less error prone than lengthy, unclear and unstructured descriptions.

Execution of data-to-paper. For each run, data-to-paper is provided with a dataset, its associated metadata, and an optional research goal and proceeds automatically through the stepwise research process (Fig. 1A,B). In open-goal modality, data-to-paper runs through the entire research process (Fig. 1A,B). In fixed-goal modality, the research goal is provided and the steps for choosing a research goal are skipped (“Fixed-goal modality”, Fig. 1A). Human interactions are implemented as a simple user approval at each research step (autopilot mode; user is only overseeing) or with complete human review through an interactive app (co-pilot mode; user can provide reviewing comments at each step, beside the highly technical “Table Design” step). For each dataset, we performed multiple data-to-paper runs, as follows. (A) “Health Indicators” dataset. We ran data-to-paper in an open-goal modality with this dataset and its associated metadata for 5 full research cycles (Supplementary Runs and Manuscripts A1-5). Overseeing the process, we aborted and restarted the 5th run 3 times after the “Goal validation” step, when observing that the chosen Research goal was too similar to goals of prior research cycles. (B) “Social Network” dataset. We ran data-to-paper in an open-goal modality with this dataset and its associated metadata for 5 full research cycles (Supplementary Runs and Manuscripts B1-5). To minimize overlapping goals in repeated runs, a list of the already-chosen previous goals was presented as part of the “mission prompt” of the “Research goal” step. (C) “Infection” dataset. We ran data-to-paper in an open-goal modality with this dataset and its associated metadata for 4 full research cycles (Supplementary Runs and Manuscripts C1-4). In addition, we ran data-to-paper in co-pilot mode in open-goal modality, of which we provide one

example manuscript (Supplementary Manuscript Ch). (D) “Treatment Policy” dataset. We ran data-to-paper in a fixed-goal modality for 10 research cycles with this dataset and its associated metadata and research goal (Supplementary Runs and Manuscripts D1-10) We also ran this data-set in open-goal modality (see example Supplementary Manuscript Do). (E) “Treatment Optimization” dataset. We ran data-to-paper in a fixed-goal modality for 10 full research cycles with this dataset and its associated metadata and research goal (Supplementary Runs and Manuscripts Ea1-10). We also ran this data-set in open-goal modality (see example Supplementary Manuscript Eo). We then ran data-to-paper with 5 modified research goals (Supplementary Data Descriptions Eb, Ec, Eai, Ebi, Eci) for 10 times per goal (Supplementary Runs and Manuscripts Eb1-10, Ec1-10, Eai1-10, Ebi1-10, Eci1-10). As these additional 50 runs were only used to annotate analysis failure, we terminated them after the “Title & abstract” step (to save unnecessary api calls). In addition, we ran data-to-paper in co-pilot mode for three times on the original goal (Supplementary Data Description Ea). During each of these runs, we provided several review comments, typically in the code writing step (Supplementary Runs and Manuscripts Eh1-3).

Overview of data-to-paper implementation. We implement data-to-paper as a chained list of research steps, each designed to create one or more research products based on a provided subset of prior research products (Fig. 1A,B). Each such research step is implemented as a distinct “Performer conversation”, which specifies LLM identity, relevant prior research products and a step-specific “mission prompt” requesting the LLM to create a focal product. Product extracted from the LLM response undergoes rule-based review and programmatic feedback requesting corrections is sent back to the LLM. For certain research steps, once the product passes rule-based review it can also be sent for LLM review, which is implemented in a parallel “Reviewer conversation” (“Review”, “LLM reviewer agent” in Fig. 1A,C respectively). The research step terminates with a final product that has passed both rule-based and LLM-based review. Once all steps are completed, a manuscript is assembled and compiled from the products of all relevant steps. All steps are recorded in log files *(i)* “openai_responses.txt”, which contains the cached responses from the LLM and any human intervention, *(ii)* “semantic_scholar_responses.bin”, which contains the cached responses from the Semantic Scholar API, and *(iii)* “code_runner_cache.pkl”, which contains the cached outputs of the code run, guaranteeing identical numerical results for each run of the same code. This allows “replaying” the run, thus recreating the same manuscript, even when run on different environments. In addition, we log other files like the actual final code (“data_exploration.py”, “data_analysis.py” and “data_to_latex.py”), the final latex file (“paper.tex”) and the API usage cost (“api_usage_cost.json”) for user convenience.

Devising prompts. The prompts used by data-to-paper in each of the research steps are listed in Table S1. These prompts have been designed in an iterative and empirical process. First, we devised an initial version for each of the prompts, focusing on the key aspect of their focal task (dark brown text, Table S1). Additionally, we added to each prompt formatting instructions for the research product (light blue and red text, Table S1). Then, we tested ChatGPT responses through multiple pilot runs, identified wrong or inadequate responses, and adapted the prompts with additional details and specifications (light brown text, Table S1). In cases where ChatGPT still failed to consistently respond as expected, we also added one-shot examples (green text, Table S1).

Message types. Messages in a conversation are designated as either SYSTEM, USER, or ASSISTANT (per OpenAI API terminology[32]). SYSTEM and USER messages are programmatically composed by data-to-paper. ASSISTANT messages are created by the LLM. We also implement LLM-surrogating ASSISTANT messages, which are messages created programmatically by data-to-paper, but are attributed to the ASSISTANT (namely, they appear to the LLM as if they were created by it).

Performer conversation. At the onset of each research step, a distinct Performer conversation is initiated and programmatically pre-filled with a list of "context messages": (*i*) "system prompt" defining the identity of the performer LLM agent ("Performer system prompt", Table S1); (*ii*) "provided prior products", a list of USER messages providing the LLM with a predefined subset of research products of prior steps, with each such USER message followed by an LLM-surrogating acknowledgment message (Fig. 1B, Figs. S1,S4; "Provided prior products", Table S1); and (*iii*) USER message describing to the LLM what it is requested to do in the current step ("Performer mission prompt", Table S1). This pre-filled Performer conversation is then sent to the LLM API[32–34] to request an initial response (Figs. S1,S4). The requested research product is then extracted from this initial LLM response and undergoes rule-based product review.

Rule-based product review. At each research step, the LLM is requested to send a response containing a specific product, with specific formatting (Fig. 1B; Tables S1,S7). Then, data-to-paper extracts the requested product from the LLM response based on its expected formatting (Tables S1,S6; for example, when requesting a "LaTex text" product, we expect the product to be enclosed within triple backticks). Failure to extract the product is translated into a feedback message sent back to the LLM (for example: "*You sent 2 triple-backtick blocks. Please send the latex as a single triple-backtick 'latex' block*"). Once the product is extracted successfully, it is programmatically refined according to a set of step-specific auto-refinement rules (Table S7, asterisk-marked rules). Then, the refined

product is checked according to a set of step-specific test rules, including formatting, text length and correct referencing (for exhaustive list see Table S7). Failure to pass any of these rules is translated into a corresponding feedback message sent back to the LLM (see example in Fig. S5).

Information tracing. To follow information flow through all steps, data-to-paper keeps track of the specific code lines producing each file output, the translation of these outputs into tables and the incorporation of numbers from the table in the Results section. Specifically, to track numeric results in the Results section, we programmatically assign a unique label for each numeric value appearing in the prior products for the Results writing step, and present these products in the context messages with the numeric values formatted as LaTex hypertargets with their corresponding labels (Fig. 1B). We then complemented the mission prompt of the Result writing step with instructions requesting the LLM agent to wrap each numeric value that it writes with a LaTex hyperlink matching the corresponding label ("Performer mission prompt: additional instructions for data-chained manuscripts", Table S1). To allow the LLM to include numeric values which are not direct output of the code, but are rather arithmetically derived from them (like changing units, translating regression coefficients to odds ratios, etc), we further provide it with the option of using a specific syntax, \num(<formula>, "explanation"), where it can provide arithmetic formula to derive new values from values created by the code output, and provide an explanation. A rule-based feedback was added to algorithmically verify that, either as stand-alone or within a \num formula, each numeric value mentioned in the section is hyperlinked, and that the target of each link correctly matches the corresponding label provided in the prior product context. Upon compilation, the \num commands are replaced with their value and a "Notes" appendix is added listing all formulas with their explanation. To further safeguard against hallucinated or missing values, the Results "mission prompt" instructs the LLM to use a designated placeholder (specifically '[unknown]') for missing numeric values, detection of this or other placeholder in the LLM response leads to data-to-paper aborting the entire research cycle (for the list of placeholders see "Results", Table S7).

Data-chained manuscripts. Reflecting the tracing of information during the "Data Analysis", "Table Design" and "Results" writing steps, data-to-paper creates manuscripts that "chain" results, methods and data, where each numeric value is recursively linked to the specific lines of codes that created it. In particular, a numeric value in the "Results" section can be linked to the "Notes" appendix, and from there to a specific value in a table, and from there to the output file that was used to create this table and finally to the specific code lines which

generated this output file (Supplementary Manuscripts C1-4, Ch, Do, Eo, Eh1-3; Note that prior manuscripts were created without this feature and do not have hyperlinks).

Reviewer conversation. For a subset of research steps, data-to-paper also performs an LLM review after the successful completion of rule-based product review ("Review", Fig. 1A; Fig. 2, Figs. S2,S6). LLM review is implemented in a "Reviewer conversation", which parallels the Performer conversation of the given step, but with inversion of the USER-ASSISTANT roles (Fig. 1C; Fig. 2A; see examples in Fig. 2B, Fig. S6). In parallel to its corresponding Performer conversation, this Reviewer conversation is pre-filled with the following list of context messages: (*i*) "system prompt" defining the identity of the LLM reviewer agent; (*ii*) the list of "provided prior products" for the focal step; and (*iii*) An LLM-surrogating message with the "Performer mission prompt" (namely, the "Performer mission prompt" is casted as an ASSISTANT-side message, thereby appearing as if it was created by the LLM reviewer agent). Then, the extracted product coming from the Performer conversation is presented as a USER-side message together with step-specific review instructions, in which the LLM reviewer agent is requested to choose between accepting the provided product, or providing constructive feedback ("Reviewer mission prompt", Table S1; Fig. 1C; Fig. 2A; Fig. S2). The pre-filled conversation is then sent to the LLM API to request a response from the Reviewer agent. If the Reviewer response contains feedback, it is transferred to the Performer conversation as if it were a USER-side message, requesting the LLM performer agent to provide a new response with an accordingly refined product.

Coding steps. For each of the three coding steps ("Data exploration", "Data analysis", "Table design"), we extract Python code (enclosed within a triple-backtick block) from the LLM response, and test this code at four levels: (*i*) *Static analysis:* Check that the code conforms to a step-specific requested structure ("Python code - Static checks", Table S7); (*ii*) *Runtime analysis:* Syntax errors, runtime errors, warnings, as well as violations of other restrictions are caught and evaluated during code execution ("Python code - Runtime checks", Table S7); (*iii*) *Package-specific guardrails:* Noting common ChatGPT coding mistakes, we wrapped the packages that we allow importing, adding multiple guardrails to monitor, control and restrict unsafe functionalities, as well as to allow rule-based review of p-value formatting (Table S8); (*iv*) *Output analysis:* Check that all the requested output files are created and contain the requested information with the requested formatting ("Numerical data checks", Table S7). Encountered issues from these 4 check levels are translated into a feedback message sent back to the LLM. As a new feedback message is added, older feedback-response message pairs are removed from the conversation (to avoid exceeding the token limits). Once the LLM-provided code passes all tests, we proceed to LLM product

review: data-to-paper provides a message that shows the LLM the code output and asks it to check the code and the output and provide a list of issues and suggested corrections (see "Reviewer mission prompts" for "Data exploration" and "Data analysis" steps in Table S1). If the LLM returns suggestions for improvement, data-to-paper requests making these corrections and enters a new phase of code debugging as described above. If there are no suggestions for improvements, we end the coding step with the code and the output files it created as the corresponding research products.

Citation retrieval. For the two literature search steps ("Literature search I", "Literature search II", Fig. 1A; Table S1), data-to-paper augments the LLM with Semantic Scholar Academic Graph API[30], an external citation database and search service. This direct citation retrieval, along with algorithmic checks restricting LLM's memory-retrieved citations (Rule-based product review; Table S7), ensures that only valid citations are included in the resulting paper. These literature-search steps start with a "Devise queries" step, in which the LLM is requested to provide a list of queries for each of a predefined set of scopes (scopes for "Literature search I": "Dataset", "Questions"; scopes for the "Literature search II": "Background", "Dataset", "Methods", "Results"; see "Literature search I" and "Literature search II" in Table S1). Then, data-to-paper calls the citation API[30] to retrieve a list of citations for each of the LLM-provided queries (see example in Fig. S7). For each citation, the API provides: (*i*) *Search rank*; (*ii*) *BibTeX ID*; (*iii*) *Title*; (*iv*) *Journal and year*; (*v*) *One-sentence paper summary (TLDR)*[51]; (*vi*) *Citation influence*[52]; (*vii*) *Title and abstract embedding*[42]. Citations for each of the scopes are then filtered and sorted either by Search rank or by Title and abstract embedding similarity to the title and abstract embedding of the currently written paper (parameters in Table S9). For the runs with datasets C, D, where we attempt reproducing a specific original study, we manually excluded the citation of the original paper. The sorted lists of papers for each scope are then provided as prior products for steps in which the LLM is requested to refer to literature citations (Table S1; Fig. 1B, Fig. S7).

LLM selection. We compared the performance of Llama 2, Codellama and ChatGPT models in two critical research steps: (*i*) Research goal and (*ii*) Data analysis. For both tests, we used the "Health Indicators" dataset. In (*i*), we ran the research goal step of data-to-paper 10 times each either with gpt-3.5-turbo or Llama-2-70b-chat-hf, all provided with the same prior product context (Table S2). We manually annotated the goals, scoring analysis-related factors, either corresponding to true features of the dataset, or to hallucinated features not part of the dataset (Table S2, Fig. S3A). In (*ii*) we ran the data analysis step of data-to-paper 10 times each with either gpt-3.5-turbo, gpt-4,

CodeLlama-34b-Instruct, Llama-2-70b-chat or Llama-2-7b-chat and evaluated for each run the number of programmatic feedback rounds until the code passes rule-based review (Fig. S3B, Supplementary Coding Runs).

LLMs and parameters. As the underlying LLM, we used OpenAI conversational ChatGPT models[32] (the open-source models we tested created hallucinated research goals and were not able to consistently converge in the data analysis coding step; LLM selection). The OpenAI models used were either gpt-3.5-turbo-0613, gpt-3.5-turbo-16k-0613, or gpt-4 (all with a knowledge cutoff of September 2021). For each research step, we assigned one of these specific models as a nominal model based on the expected conversation length of the step as well as the presumed difficulty and performance during pilot runs ("LLM", Table S1). Starting from this initial nominal model for each step, data-to-paper can automatically upgrade the model during a conversation: switching to a "stronger" model if such a model exists, when a rule-based feedback request is not resolved, and switching to model with larger context window size, if such one exists, if the number of tokens exceeds the maximum of the step's nominal model. For all models, we use default model parameters, except for the model's temperature which was specifically set for some of the steps (In particular, setting a temperature of 1 for the "Research goal" step).

Paper assembly and compilation. To produce the final manuscript, data-to-paper assembles a single LaTex file, combining the different manuscript-part products ("Paper assembly", Fig. 1A,B). It then automatically compiles this file, together with the list of citations retrieved from "Literature search II", into a pdf, watermarked "Created by data-to-paper (AI)".

Manual review of created manuscripts. We manually vetted each created manuscript and its respective run record (Supplementary Manuscripts and Runs A1-5, B1-5, C1-4, D1-10, Ea1-10, Eb1-10, Ec1-20, Eai1-10, Ebi1-10, Eci1-10). For the manuscripts, we verified: (*i*) that the data analysis and code are correctly performed, using adequate statistical methodologies; (*ii*) that every statement in the text involving numeric information corresponds to the correct numeric value from the output of the data analysis; (*iii*) that every citation fits the context in which it was referenced; (*iv*) the overall exactness of the text; and (*v*) the quality of the overall text and wording. The manuscripts were highlighted to reflect correctly-put statements (green), imperfect, or atypical statements (yellow), minor errors (orange), and major errors (red). Of note, for Supplementary Manuscripts C1-4 we only highlighted major errors in data analysis code, which heavily impacted the accuracy of following steps.

Human co-piloting. Human co-piloting is incorporated by allowing the user to add review comments in each step after the rule-based and LLM-review have completed. If such human review is added, data-to-paper initiates a new cycle of Performer answers with rule-based checks. This process repeats iteratively until the user approves the research product of the step. We have created an app with a user interface that allows the user to follow the LLM conversation in each step and add review comments as needed.

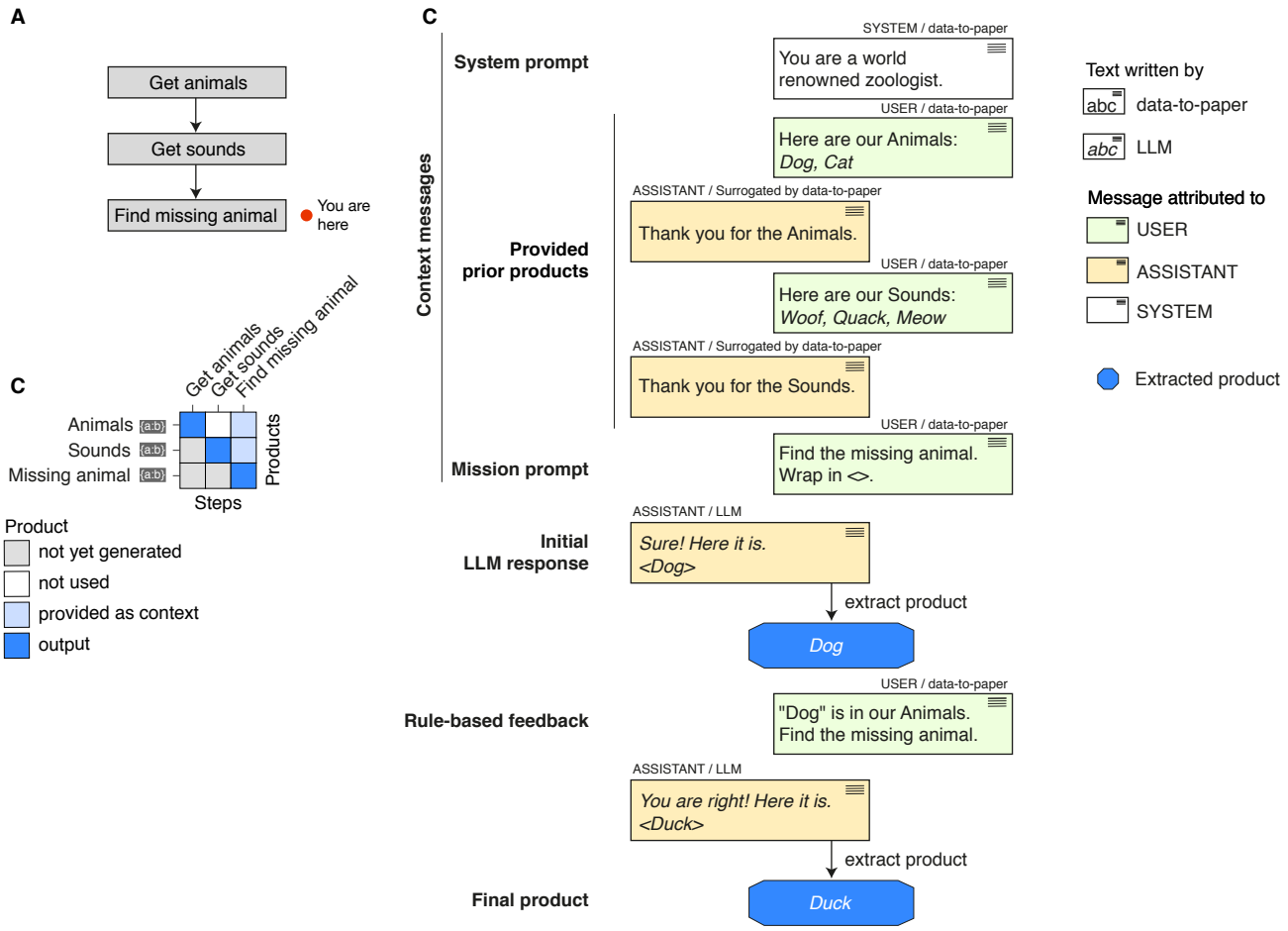

Figure S1. Toy example of a research step implementation. A. The toy research path consists of 3 steps: composing a list of animals, composing a list of sounds and identifying the missing animal (equivalent to Fig. 1A). **B.** Products created and used in each step. The last step of the research path is provided with the products of the two prior steps (equivalent to Fig. 1B). **C.** An implementation of the "Finding missing animal" toy research step. The conversation is filled with "context messages": (i) a "system prompt" (white message box); (ii) "provided prior products", pairs of messages where the first is a USER-side message containing one of the prior research products (green message box), and the second is a short acknowledgment message ("Thank you for the <product>", orange message box) that is programmatically written by data-to-paper (un-italicized text) yet attributed to the ASSISTANT (namely, appearing to the LLM as if it was written by it, hence LLM-surrogating; Methods); (iii) "mission prompt" indicating the requested product (here, name of the missing animal) and the formatting (here enclosed with <>; see real formatting instructions in Table S1). This programmatically filled conversation is sent to the LLM API, which returns an LLM-authored response message (italicized text, orange message box). The research product is then automatically extracted from this LLM response according to its predefined formatting (blue octagon). The extracted product undergoes rule-based checks, and upon failure an appropriate feedback message is sent to the LLM as a USER-side message ("Rule-based feedback", green message box). The LLM replies with a corrected response, from which the final product is extracted (blue octagon). See also a real example of a research step in Fig. S4.

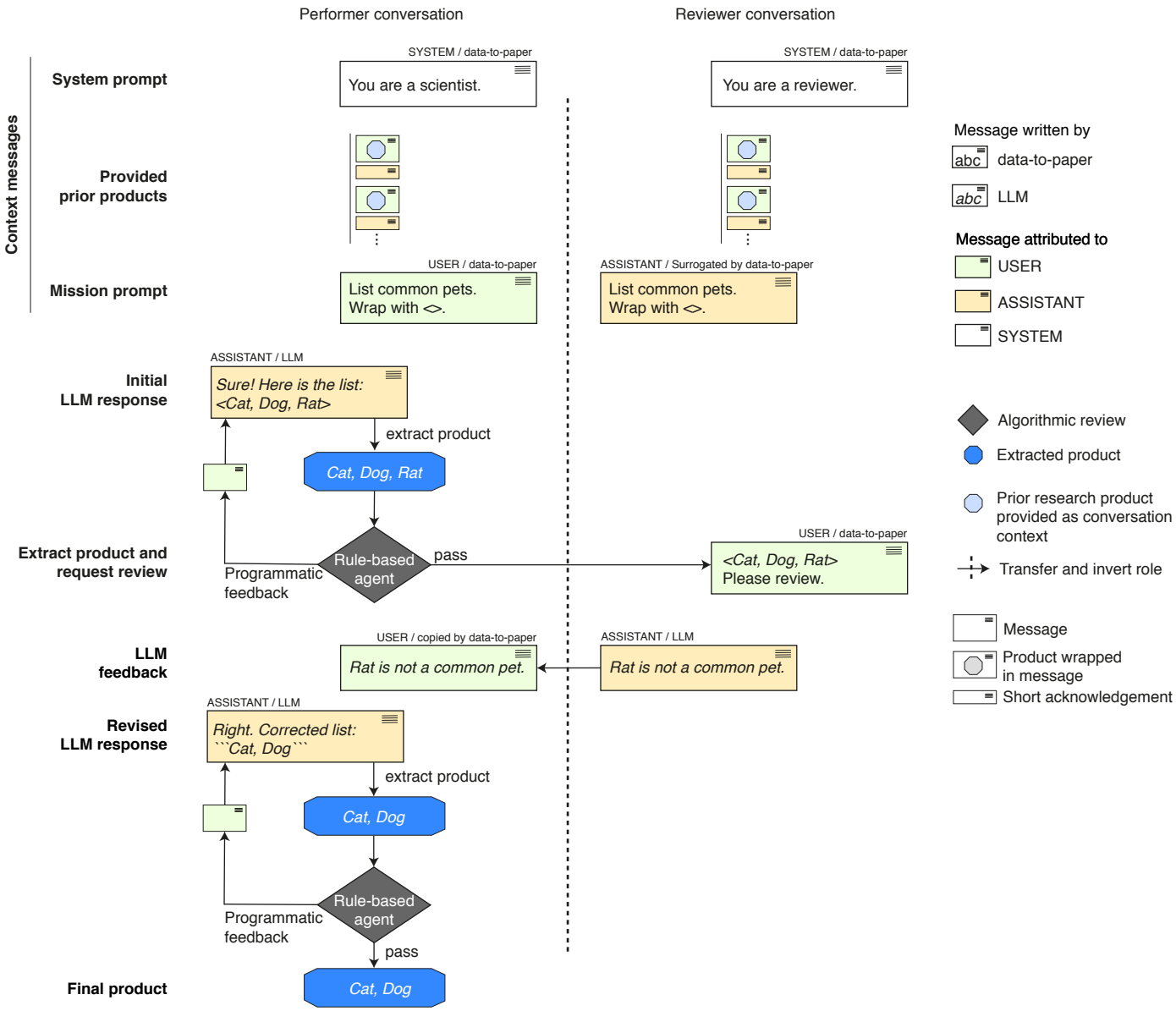

Figure S2. Internal LLM review is implemented by transferring messages between two role-inverted LLM conversations. Layout of the Performer conversation (left) and the parallel role-inverted Reviewer conversation (right; see also Fig. 2A, Methods). First, the two conversations are programmatically filled with a list of "context messages", including a "system prompt", messages providing prior products, and the "mission prompt" (as in Fig. 1C, Fig. S1; Methods). Notably, the mission prompt is added as a USER-side message (green message box) in the Performer conversation and as an LLM-surrogating ASSISTANT message (orange message box) in the Reviewer conversation (Methods). Second, data-to-paper requests an LLM response for the Performer conversation ("Initial LLM response", orange box), extracts the requested product, and performs rule-based checks, providing programmatic feedback (upwards going green message box; Methods; Fig. S1). Third, once the product passes rule-based review, it is sent for LLM review; a USER-side message containing the extracted product and review instructions is appended to the reviewer conversation ("Extract product and request review", green message box; Table S1), and a response is requested from the LLM reviewer ("LLM feedback", orange message box). Fourth, data-to-paper copies the review message, appending it to the performer conversation as if it were a USER-side message ("LLM feedback", green message box). Finally, a new LLM response is requested from the performer, in which it corrects the product according to the feedback from the LLM reviewer agent ("Revised LLM response", orange message box). The revised product undergoes rule-based review, until it passes, which terminates the step with a final product.

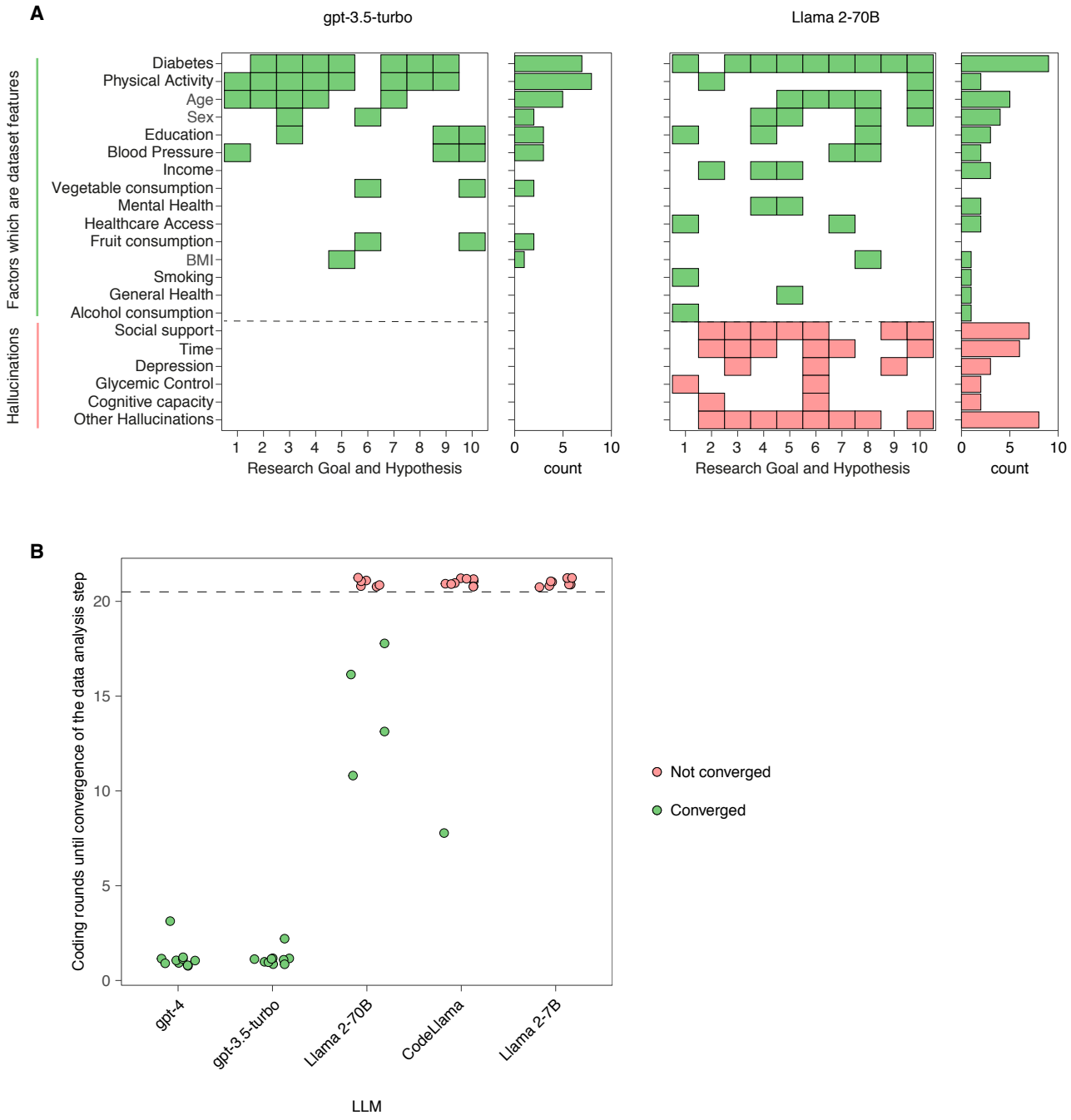

Figure S3. Current state-of-the-art open-source LLMs are not able to perform key data-to-paper steps. A. Evaluation of the performance of different LLMs in the "Research goal" step. Starting from the same "context messages", including Data exploration results, we ran the "Research goal" step on the "Health Indicators" dataset for 10 times with either gpt-3.5-turbo or Llama 2-70B, and manually vetted the resulting research goals, annotating factors as either being dataset features or hallucinations (Table S2). While gpt-3.5-turbo only used factors present in the dataset in the research goal and hypotheses, Llama 2-70B hallucinated in all 10 research, providing goals that included factors or information which were not present in the dataset. **B.** Evaluation of LLMs in the "Data analysis" step. Each LLM performed the "Data analysis" step on the "Health Indicators" dataset for 10 times with programmatic review only. A maximum of 20 coding rounds was allowed. While gpt-3.5-turbo and gpt-4 converged to functional code within 1-3 coding rounds (median = 1), Llama 2-7B never converged, CodeLlama converged only once after 8 coding rounds, and Llama 2-70B in only 4 runs with 13-18 coding rounds (Supplementary Coding Runs).

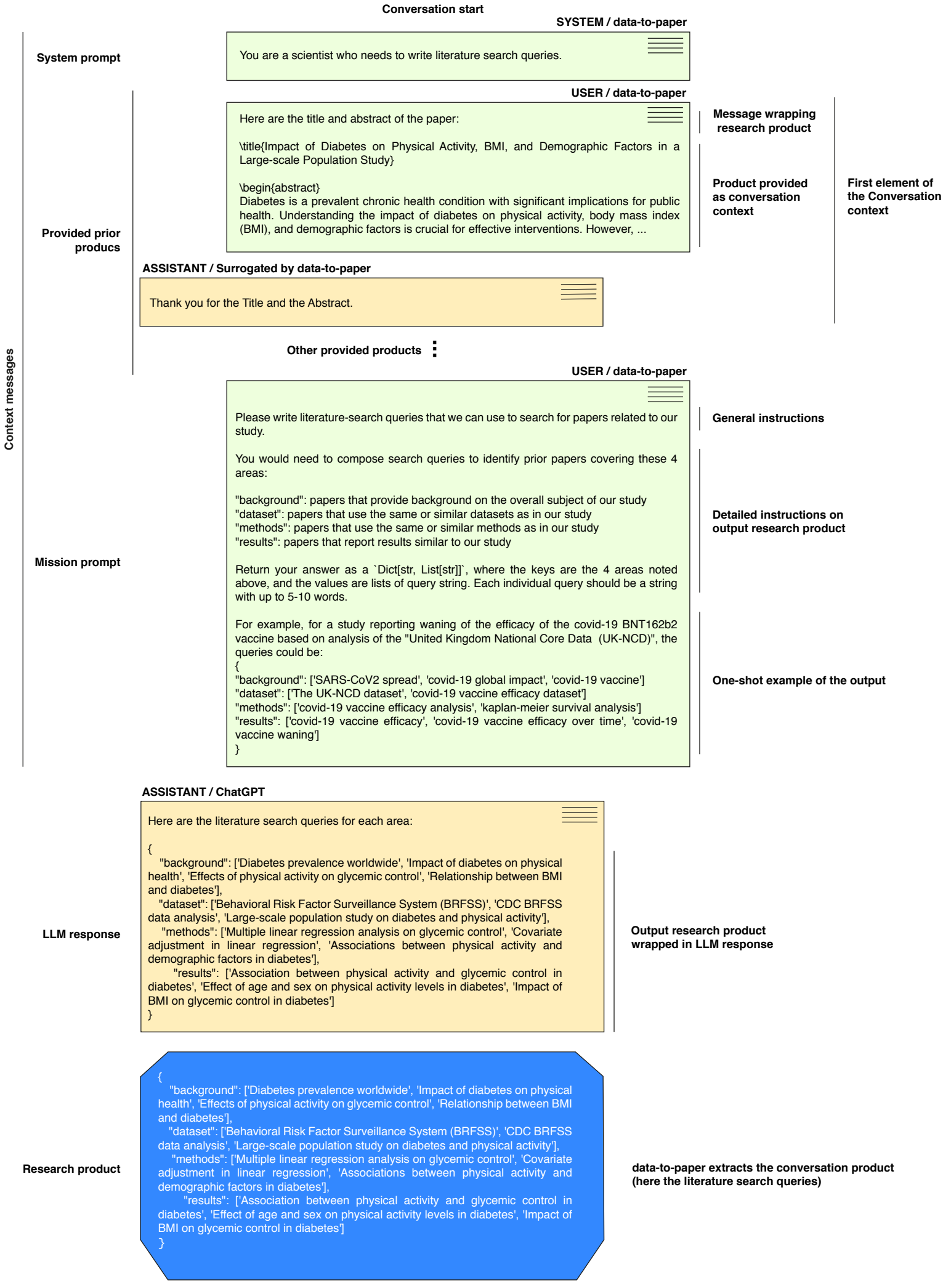
Conversation start
SYSTEM / data-to-paper
System prompt
You are a scientist who needs to write literature search queries.
USER / data-to-paper
Here are the title and abstract of the paper:
\title{Impact of Diabetes on Physical Activity, BMI, and Demographic Factors in a Large-scale Population Study}
\begin{abstract}
Diabetes is a prevalent chronic health condition with significant implications for public health. Understanding the impact of diabetes on physical activity, body mass index (BMI), and demographic factors is crucial for effective interventions. However, ...
Message wrapping research product
Product provided as conversation context
First element of the Conversation context
Provided prior producs
ASSISTANT / Surrogated by data-to-paper
Thank you for the Title and the Abstract.
Other provided products
USER / data-to-paper
Please write literature-search queries that we can use to search for papers related to our study.
You would need to compose search queries to identify prior papers covering these 4 areas:
"background": papers that provide background on the overall subject of our study
"dataset": papers that use the same or similar datasets as in our study
"methods": papers that use the same or similar methods as in our study
"results": papers that report results similar to our study
Return your answer as a `Dict[str, List[str]]`, where the keys are the 4 areas noted above, and the values are lists of query string. Each individual query should be a string with up to 5-10 words.
For example, for a study reporting waning of the efficacy of the covid-19 BNT162b2 vaccine based on analysis of the "United Kingdom National Core Data (UK-NCD)", the queries could be:
{
"background": ['SARS-CoV2 spread', 'covid-19 global impact', 'covid-19 vaccine']
"dataset": ['The UK-NCD dataset', 'covid-19 vaccine efficacy dataset']
"methods": ['covid-19 vaccine efficacy analysis', 'kaplan-meier survival analysis']
"results": ['covid-19 vaccine efficacy', 'covid-19 vaccine efficacy over time', 'covid-19 vaccine waning']
}
General instructions
Detailed instructions on output research product
One-shot example of the output
Context messages
Mission prompt
ASSISTANT / ChatGPT
Here are the literature search queries for each area:
{
"background": ['Diabetes prevalence worldwide', 'Impact of diabetes on physical health', 'Effects of physical activity on glycemic control', 'Relationship between BMI and diabetes'],
"dataset": ['Behavioral Risk Factor Surveillance System (BRFSS)', 'CDC BRFSS data analysis', 'Large-scale population study on diabetes and physical activity'],
"methods": ['Multiple linear regression analysis on glycemic control', 'Covariate adjustment in linear regression', 'Associations between physical activity and demographic factors in diabetes'],
"results": ['Association between physical activity and glycemic control in diabetes', 'Effect of age and sex on physical activity levels in diabetes', 'Impact of BMI on glycemic control in diabetes']
}
LLM response
Output research product wrapped in LLM response
{
"background": ['Diabetes prevalence worldwide', 'Impact of diabetes on physical health', 'Effects of physical activity on glycemic control', 'Relationship between BMI and diabetes'],
"dataset": ['Behavioral Risk Factor Surveillance System (BRFSS)', 'CDC BRFSS data analysis', 'Large-scale population study on diabetes and physical activity'],
"methods": ['Multiple linear regression analysis on glycemic control', 'Covariate adjustment in linear regression', 'Associations between physical activity and demographic factors in diabetes'],
"results": ['Association between physical activity and glycemic control in diabetes', 'Effect of age and sex on physical activity levels in diabetes', 'Impact of BMI on glycemic control in diabetes']
}
Research product
data-to-paper extracts the conversation product (here the literature search queries)

Figure S4. An example of a research step conversation. An example Performer conversation for the "Literature search II" step for the "Health Indicators" dataset (Supplementary Run A2), composed of the following list of messages (letterheaded boxes, titled with "attribution / source"; attributions: ASSISTANT, USER or SYSTEM; sources: data-to-paper or ChatGPT): "context messages", including (i) "system prompt", a SYSTEM message defining the identity of the LLM agent; (ii) "provided prior products", a list of USER-ASSISTANT message pairs providing relevant products of prior steps (here, for simplicity, only the "Title & abstract" prior product is shown out of the 4 context products provided in this step, Fig. 1B). To mimic a conversation, each USER-side message providing a product is followed by an LLM-surrogating acknowledgment message; (iii) "mission prompt", a USER-side message providing instructions for the LLM, including general instructions, detailed instructions and formatting instructions (Table S1). This programmatically filled conversation is sent to OpenAI ChatGPT API[32]. Based on the prior product context and the specific instructions for the current step, the LLM then generates a response message that should contain the requested research product (here a dictionary containing the list of queries for each requested scope). The LLM-created research product is automatically extracted from the message by data-to-paper (blue octagonal box). These extracted products then further undergo rule-based and LLM-based review (Fig. 1C; Fig. 2A; Figs. S1, S2, S5, S6).

A

USER / data-to-paper

Mission prompt

Please write literature-search queries that we can use to search for papers related to our study.
...
Return your answer as a `Dict[str, List[str]]`, where the keys are the 4 areas noted above, and the values are lists of query string. Each individual query should be **a string with up to 5-10 words.**

ASSISTANT / ChatGPT

Initial LLM response

{...
"results": ['Association between physical activity and glycemic control in diabetes', **'Effect of age and sex on physical activity levels in diabetes'**, 'Impact of BMI on glycemic control in diabetes']
...
}

USER / data-to-paper

Rule-based feedback

Queries should be 5-10 word long.

The following queries are too long:
"Effect of age and sex on physical activity levels in diabetes"

Please return your complete response again, with these queries shortened.

ASSISTANT / ChatGPT

Revised LLM response

Apologies for the confusion. Here is the revised version with shorter queries:
{...
"results": ['physical activity and glycemic control', **'age and sex impact on physical activity'**, 'BMI and glycemic control in diabetes'
...
}

B

USER / data-to-paper

Mission prompt

Write a complete Python code to analyze the data and create dataframes as basis for scientific Tables for our paper.

The code must have the following sections (with these exact capitalized headers):

`# IMPORT`
...
SAVE ADDITIONAL RESULTS
...

ASSISTANT / ChatGPT

Initial LLM response

```
# IMPORT
...
additional_results = {
 'Total number of observations': len(df_cleaned),
 'Pseudo R-squ of regression model': result.prsquared
}
with open('additional_results.pkl', 'wb') as f:
    pickle.dump(additional_results, f)
...
```

USER / data-to-paper

Rule-based feedback

Your code must contain the following sections:
('# IMPORT', '# LOAD DATA', '# DATASET PREPARATIONS', '# DESCRIPTIVE STATISTICS', '# PREPROCESSING', '# ANALYSIS', '# SAVE ADDITIONAL RESULTS').
But I could not find these headers:
['# SAVE ADDITIONAL RESULTS'].
Please rewrite the complete code again with all the required sections.

ASSISTANT / ChatGPT

Revised LLM response

I apologize for the oversight. Here is the corrected code:

```
# IMPORT
...
# SAVE ADDITIONAL RESULTS
additional_results = {
 'Total number of observations': len(df_cleaned),
 'Pseudo R-squ of regression model': result.prsquared
}
with open('additional_results.pkl', 'wb') as f:
    pickle.dump(additional_results, f)
...
```

Figure S5. Example of rule-based review. Two examples are shown for algorithmic rule-based review of products extracted from LLM responses (Supplementary Manuscripts A2,4). **A.** Devising literature-search queries. In the "Literature search II" step, a "mission prompt" message requests the LLM to write search queries, for each of several scopes, indicating the allowed length of these queries ("Mission prompt"; Table S1). The LLM performer agent returns a message containing a list of queries for each scope ("Initial LLM response"). Detecting that one of these queries is violating the length limitation, the rule-based agent issues an algorithmic feedback message to be sent back to the LLM, requesting to shorten this specific query ("Rule-based feedback", bold and highlighted text). The LLM is then replying with a set of new queries, now properly adhering to the length limitation (bold and highlighted text, "Revised LLM response"). **B.** Data analysis code. A "mission prompt" requests to write code for data analysis, specifying the format of the code and its output (bold and highlighted text, "Mission prompt"). LLM responds with a code that omits one of the required headers ("Initial LLM response"). Static code check by data-to-paper catches that mistake and relays feedback response to the LLM, requesting to rewrite the code with all required sections, specifically mentioning the missing header (bold and highlighted text, "Rule-based feedback"). The LLM then resends the corrected code with the additional required header (bold and highlighted text, "Revised response").

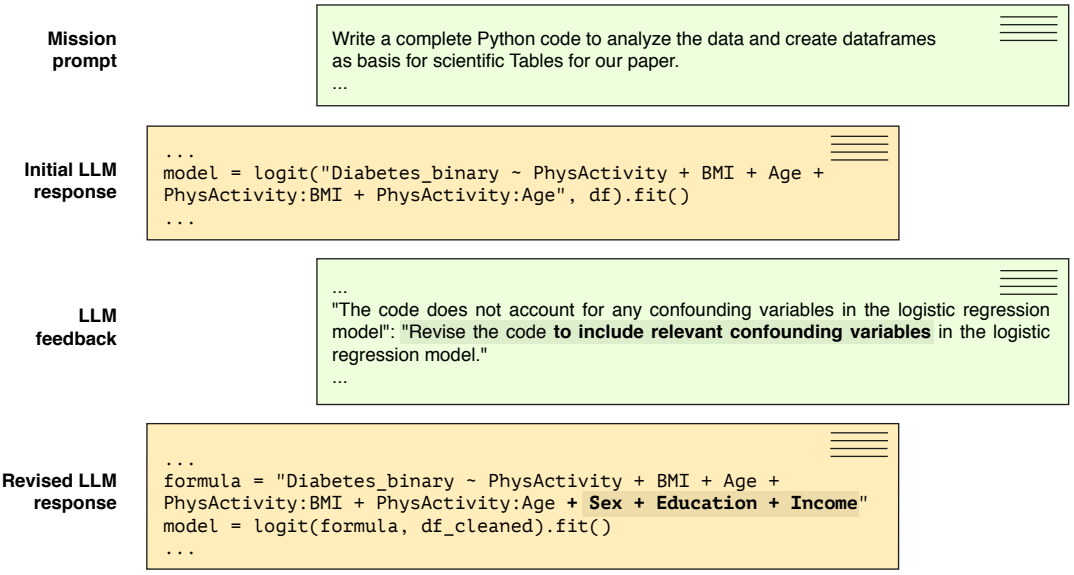

Figure S6. Example of LLM review. An initial code (Supplementary Manuscript A4), created in the “Data analysis” step, tests for interactions between variables, but does not adequately account for confounding factors (“Initial LLM response”). Following an LLM review feedback indicating this problem (bold and highlighted text, “LLM feedback”), the model is adequately augmented with additional confounding factors (bold and highlighted text, “Revised response”).

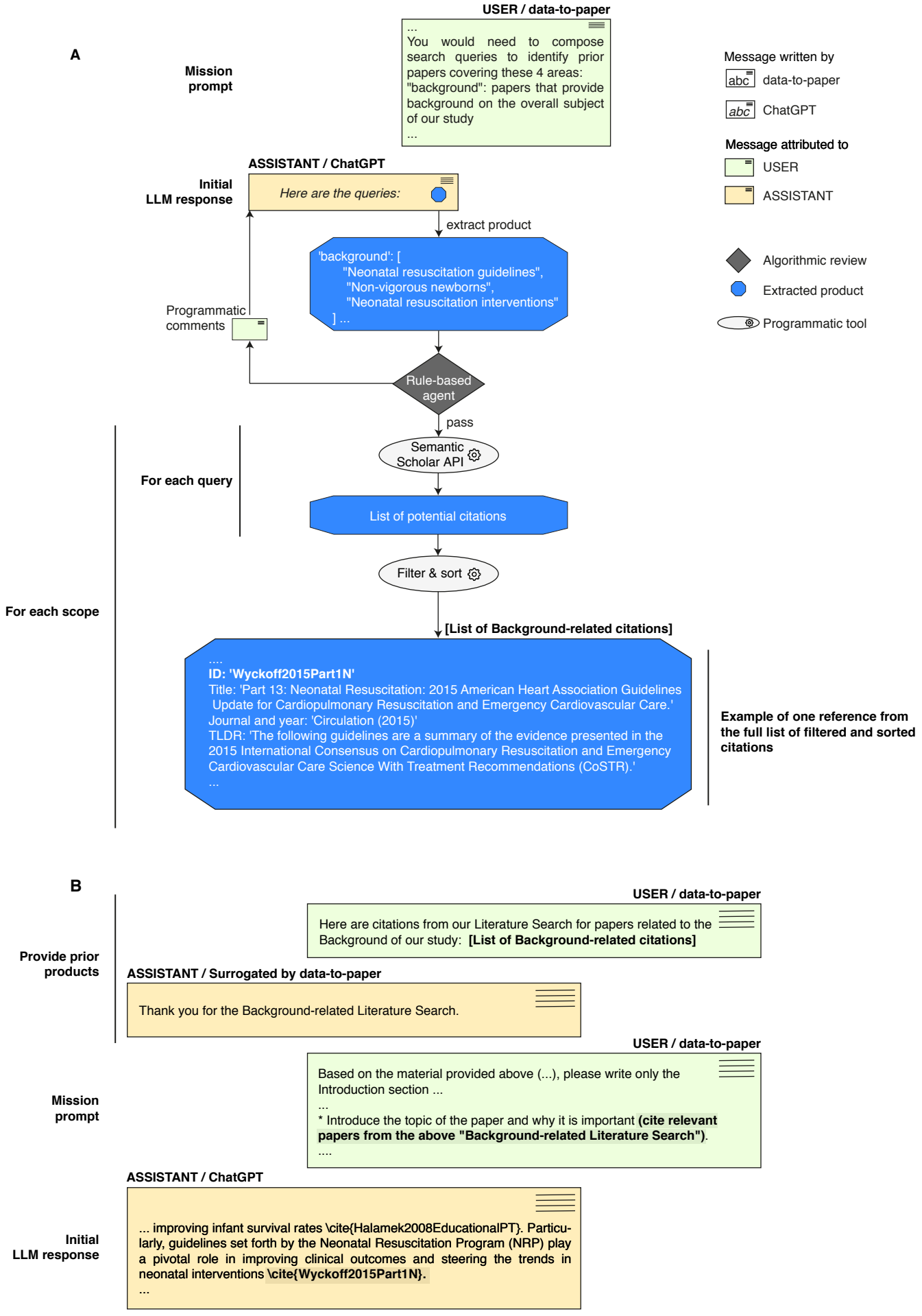

Figure S7. An external citation search API is used to retrieve citations based on LLM-created search queries. Instead of using the LLM to directly retrieve citations from memory (which can lead to hallucinated citations), we use it to devise literature search queries, which data-to-paper uses to perform a programmatic literature search using an external API[30]. **A.** "Mission prompt" requests literature search queries related to the "Title & abstract draft" on four different areas: "Background", "Dataset", "Results" and "Methods" in a "Dict[str, List[str]]" format ("Mission prompt", green box). The LLM

performer agent provides these queries in an accordingly-formatted message (“Initial LLM response”, orange box), from which they are extracted by data-to-paper (blue octagon) and passed to a rule-based review (dark diamond; Table S7). Once the queries pass rule-based review, each query is used to retrieve papers through a call to the Semantic Scholar Academic Graph API[30] (light ellipse with a gear icon). The returned queries are then programmatically filtered and sorted (light ellipse with gear icon; Methods; Table S9), resulting in the final research product (blue octagon). **B.** An example of the “Introduction” writing step, showing how previously obtained citation lists are provided as part of prior products (bold text, “Provided prior products”, green box). The performer agent is then requested to cite relevant papers from the provided list (bold and highlighted text, “Mission prompt”), which it does by using the appropriate citation BibTeX ID in the text (bold and highlighted text, bold and highlighted text, “Initial LLM response”).

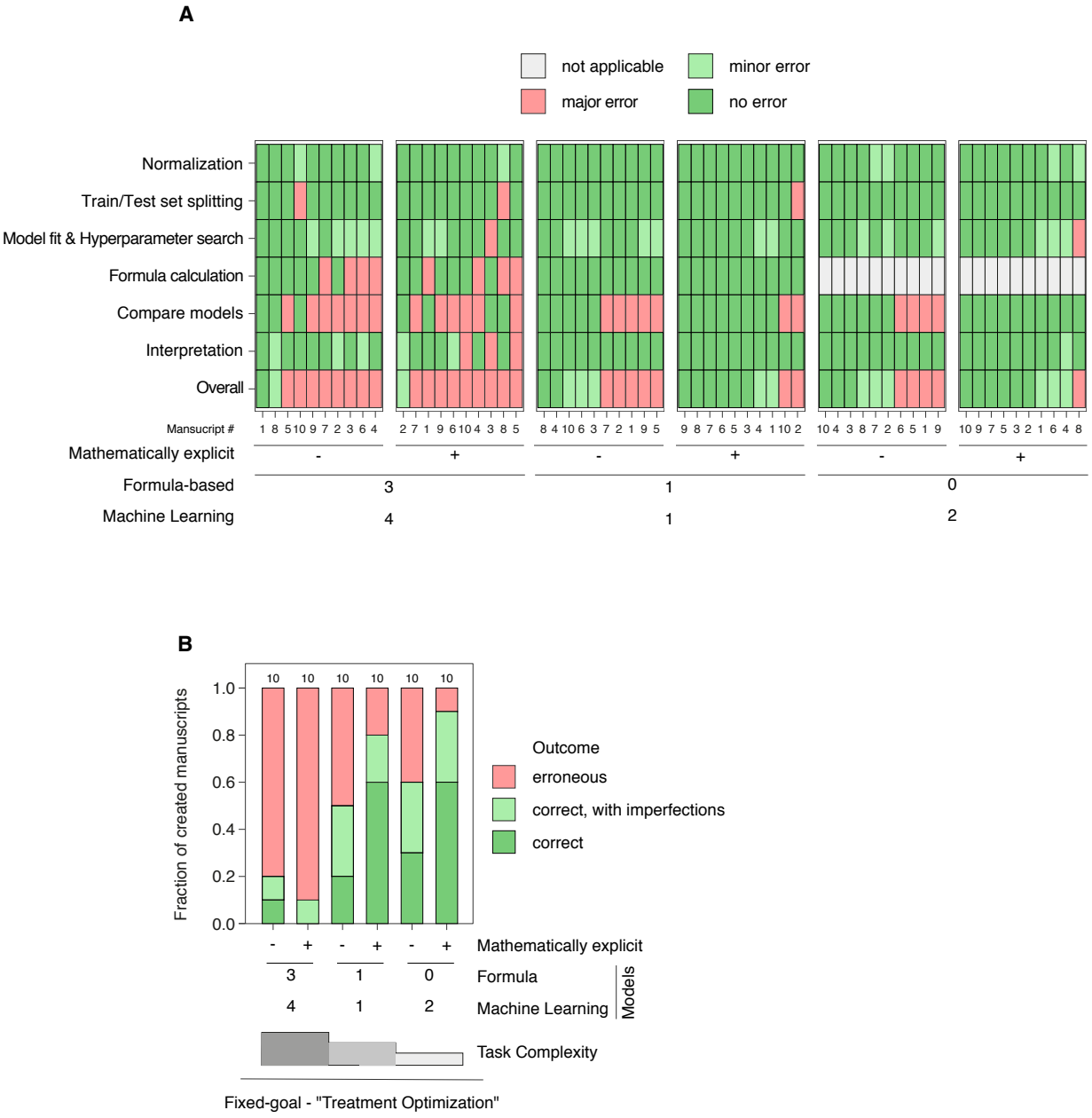

Figure S8. Increasing goal complexity leads to erroneous analysis by data-to-paper. A. Finding and quantifying data-to-paper failure modes by evaluation of different analysis and interpretation tasks (rows) for all data-to-paper "Treatment Optimization" runs (columns), for different research goals, varying in the number of Machine Learning and Formula models they require to create, as well as in whether they provide mathematically explicit explanations (see corresponding research goals in Supplementary Data Descriptions Eai, Ea, Ebi, Eb, Eci, Ec). Tasks are scored as correct (green), correct with imperfections (light green) and erroneous papers (red; corresponding to text highlighting in Supplementary Manuscripts Eai1-10, Ea1-10, Ebi1-10, Eb1-10, Eci1-10, Ec1-10). Within each goal, the 10 created manuscripts are ordered according to their number of erroneous tasks (original manuscript numbers are indicated). The "Overall" row indicates an error in any of the analysis or interpretation tasks. **B.** An overall summary of the fraction of correct, correct with imperfections and erroneous papers for each of the 6 goals.

Table S1. List of data-to-paper research steps with their Provided prior products, Performer system prompt, Performer mission prompt, created products, Reviewer system prompt, Reviewer mission prompt and Max review iterations or Max code revisions if applicable.

List of research steps

Prompts color scheme

Prompts are color-coded according to the following scheme:

General mission description
Detailed mission instructions
Product formatting instructions
One shot examples
Code output formatting guidelines
Reference to prior products

Data exploration

LLM	gpt-4
Provided prior products	Data (only provided as data file for code), General description of dataset, Data file description
Performer system prompt	You are a brilliant data scientist. You are writing a Python code to analyze data.
Performer mission prompt	As part of a data-exploration phase, please write a complete short Python code for getting a first sense of the data. Your code should create an output text file named "data_exploration.txt", which should contain a summary of the data. The output file should be self-contained; any results you choose to save to this file should be accompanied with a short header. The output file should be formatted as follows: ```output # Data Size <Measure of the scale of our data (e.g., number of rows, number of columns)> # Summary Statistics <Summary statistics of all or key variables> # Categorical Variables <As applicable, list here categorical values and their most common values> # Missing Values <Counts of missing, unknown, or undefined values> <As applicable, counts of special numeric values that stand for unknown/undefined if any (check in the "Description of the Dataset" above for any)> # <other summary you deem relevant, if any> <summary> ``` If needed, you can use the following packages which are already installed: ('pandas', 'numpy', 'scipy') Do not provide a sketch or pseudocode; write a complete runnable code. Do not create any graphics, figures or any plots. Do not send any presumed output examples.
Product	Data exploration – code: Python code Data exploration – output: Numerical data
Reviewer system prompt	You are a brilliant data scientist. You are writing a Python code to analyze data.
Reviewer mission prompt	I ran your code. Here is the content of the output file(s) that the code created: ``` <Data exploration code - output> ``` Please follow these two steps:

	(1) Check the code and the output for any issues, and return a bullet-point response addressing these points: * Are there any unexpected NaN values in the output. * Can results be understood from the output file? In particular, do we have a short label for each result? * Are there any results that are missing. Check that under each header in the output file there is a corresponding meaningful result. * Any other issues you find. (2) Based on your assessment above, return a Python Dict[str, str] mapping the issues you have noted above (dict keys) to specific suggested corrections/improvements in the code (dict values). For example: { "The result of the average of variable ... is missing": "Add the missing calculation of ... to the code.", "The average of the variable <xxx> is `Nan`": "Remove missing values in the calculation." } Try to be as specific as possible when describing the issues and proposed fixes. Include in the dict as many issues as you find. If there are no issues, and the code and tables are just perfect and need no corrections or enhancements, then return an empty dict: {} Important: * Do not return the revised code, only the issues and suggested fixes. * If there are no critical issues, then return an empty dict: `{}`. * Do not create positive issues that require no change in the code. In particular, do not write {"No issues found": "No corrections or improvements are needed."}, return an empty dict instead.
Max code revisions	5

Data exploration explanation

LLM	gpt-4
Provided prior products	General description of dataset, Data file description, Data exploration - code
Performer system prompt	You are a scientist who needs to write explanation of the Data Exploration code.
Performer mission prompt	Please return a triple-backtick Latex Block explaining what the code above does. Do not provide a line-by-line explanation, rather provide a high-level explanation of the code in a language suitable for a Methods section of a research paper. Focus on analysis steps. There is no need to explain trivial parts, like reading/writing a file, etc. Also explain what does the code write into the "data_exploration.txt" file. Your explanation should be written in LaTeX, and should be enclosed within a LaTeX Code Block, like this: ```latex \section{Code Explanation} <your code explanation here> ``` Remember to enclose your explanation within a LaTeX Code Block, so that I can easily copy-paste it!
Product	Data exploration – code explanation: LaTex text
Max review iterations	N/A

Research goal

LLM	gpt-3.5-turbo-0613, with temperature=1
Provided prior products	General description of dataset, Data file description, Data exploration - code output, Data exploration - code explanation
Performer system prompt	You are a helpful scientist.
Performer mission prompt	Please suggest a research goal and an hypothesis that can be studied using only the provided dataset. The goal and hypothesis should be interesting and novel. Guidelines: * Try to avoid trivial hypotheses (like just testing for simple linear associations). Instead, you could perhaps explore more complex associations and relationships, like testing for moderation effects or interactions between variables. * Make sure that your suggested hypothesis can be studied using only the provided dataset, without requiring any additional data. In particular, pay attention to using only data available based on the provided headers of our data files (see "Description of the Original Dataset", above). * Avoid goals and hypotheses that involve ethic issues like sociodemographic (Income, Education, etc.) and psychological (Mental Health) variables. Note though that you can, and should, still use these as confounding variables if needed. * Do not suggest methodology. Just the goal and an hypothesis. INSTRUCTIONS FOR FORMATTING YOUR RESPONSE: Please return the goal and hypothesis enclosed within triple-backticks, like this: ``` Research Goal: <your research goal here> Hypothesis: <your hypothesis here> ```
Product	Research goal: Free text
Reviewer system prompt	You are a scientific reviewer for a scientist who needs to suggest research goal and hypothesis.
Reviewer mission prompt	Here is the research goal and hypothesis: *<Research goal product>* Please provide constructive bullet-point feedback on the above research goal and hypothesis. Specifically: * If the hypothesis cannot be tested using only the provided dataset (without requiring additional data), suggest how to modify the hypothesis to better fit the dataset. * If the hypothesis is not interesting and novel, suggest how to modify it to make it more interesting.

	* If the hypothesis is broad or convoluted, suggest how best to focus it on a single well defined question. Do not provide positive feedback; if these conditions are all satisfied, just respond with: "The research goal does not require any changes". If you feel that the initial goal and hypothesis satisfy the above conditions, respond solely with "The research goal does not require any changes".
Max review iterations	1

Literature search I

LLM	gpt-3.5-turbo-0613
Provided prior products	General description of dataset, Data file description, Research goal
Performer system prompt	You are a scientist who needs to write literature search queries.
Performer mission prompt	Please write literature-search queries that we can use to search for papers related to our study. You would need to compose search queries to identify prior papers covering these 2 areas: "dataset": papers that use the same or similar datasets as in our study "questions": papers that ask questions similar to our study Return your answer as a `Dict[str, List[str]]`, where the keys are the 2 areas noted above, and the values are lists of query string. Each individual query should be a string with up to 5-10 words. For example, for a study reporting waning of the efficacy of the covid-19 BNT162b2 vaccine based on analysis of the "United Kingdom National Core Data (UK-NCD)", the queries could be: { "dataset": ['The UK-NCD dataset', 'covid-19 vaccine efficacy dataset'] "questions": ['covid-19 vaccine efficacy over time', 'covid-19 vaccine waning'] }
Product	Literature search I – queries: Structured text Literature search I – citations: Citations
Max review iterations	N/A

Similar citation search

LLM	gpt-4
Provided prior products	General description of dataset, Data file description, Research goal, Literature search I - citations (Dataset and Question scopes)
Performer system prompt	You are a scientist who needs to find most similar papers.
Performer mission prompt	From the literature search above, list up to 5 key papers whose results are most similar/overlapping with our research goal and hypothesis. Return your response as a Python Dict[str, str], where the keys are bibtex ids of the papers, and the values are the titles of the papers. For example: { "Smith2020TheAB": "A title of a paper most overlapping with our goal and hypothesis", "Jones2021AssortedCD": "Another title of a paper that is similar to our goal and hypothesis", }
Product	Similar papers: Citations
Max review iterations	N/A

Goal validation

LLM	gpt-4
Provided prior products	General description of dataset, Data file description, Research goal, Similar papers
Performer system prompt	You are a scientist who needs to check research goal and hypothesis.
Performer mission prompt	Given the related papers listed above, please follow these 3 steps: (1) Provide a bullet-point list of potential similarities between our goal and hypothesis, and the related papers listed above. (2) Determine in what ways, if any, our stated goal and hypothesis are distinct from the related papers listed above. (3) Given your assessment above, choose one of the following two options: a. Our goal and hypothesis offer a significant novelty compared to existing literature, and will likely lead to interesting and novel findings {'choice': 'OK'}. b. Our goal and hypothesis have overlap with existing literature, and I can suggest ways to revise them to make them more novel {'choice': 'REVISE'}. Your response for this part should be formatted as a Python dictionary mapping 'choice' to either 'OK' or 'REVISE'. Namely, return either: {'choice': 'OK'} or {'choice': 'REVISE'}
Product	Goal validation: Binary decision
Max review iterations	N/A

Hypothesis testing plan

LLM	gpt-3.5-turbo-0613
Provided prior products	General description of dataset, Data file description, Data exploration - code, Data exploration - code output, Research goal
Performer system prompt	You are a scientist who needs to write hypothesis testing plan.
Performer mission prompt	We would like to test the specified hypotheses using the provided dataset. Please follow these two steps: (1) Return a bullet-point review of relevant statistical issues. Read the "Description of the Original Dataset" and the "Data Exploration Code and Output" provided above, and then for each of the following generic statistical issues determine if they are relevant for our case and whether they should be accounted for: * multiple comparisons. * confounding variables (see available variables in the dataset that we can adjust for). * dependencies between data points. * missing data points. * any other relevant statistical issues. (2) Create a Python Dict[str, str], mapping each hypothesis (dict key) to the statistical test that would be most adequate for testing it (dict value). The keys of this dictionary should briefly describe each of our hypotheses. The values of this dictionary should specify the most adequate statistical test for each hypothesis, and describe how it should be performed while accounting for any issues you have outlined above as relevant. For each of our hypotheses, suggest a *single* statistical test. If there are several possible ways to test a given hypothesis, specify only *one* statistical test (the simplest one). Your response for this part should be formatted as a Python dictionary, like this: { "xxx is associated with yyy and zzz": "linear regression with xxx as the independent variable and yyy and zzz as the dependent variables while adjusting for aaa, bbb, ccc", "the association between xxx and yyy is moderated by zzz": "repeat the above linear regression, while adding the interaction term between yyy and zzz", } These of course are just examples. Your actual response should be based on the goal and hypotheses that we have specified above (see the "Research Goal" above). Note how in the example shown the different hypotheses are connected to each other, building towards a single study goal. Remember to return a valid Python dictionary Dict[str, str].
Product	Hypothesis testing plan: Structured text
Max review iterations	0

Data analysis code

LLM	gpt-4
Provided prior products	Data (only provided as data file for code), General description of dataset, Data file description, Data exploration - code output, Research goal, Hypothesis testing plan
Performer system prompt	You are a brilliant data scientist. You are writing a Python code to analyze data.
Performer mission prompt	Write a complete Python code to analyze the data and create dataframes as basis for scientific Tables for our paper. The code must have the following sections (with these exact capitalized headers): `# IMPORT` `import pickle` You can also import here any other packages you need from the following list: ('pandas', 'numpy', 'scipy', 'statsmodels', 'sklearn', 'pickle') `# LOAD DATA` Load the data from the original data files described above (see "Description of the Original Dataset"). `# DATASET PREPARATIONS` * Join dataframes as needed. * Dealing with missing, unknown, or undefined values, or with special numeric values that stand for unknown/undefined (check in the "Description of the Original Dataset" for any such values, and consider also the "Output of the Data Exploration Code"). * Create new columns as needed. * Remove records based on exclusion/inclusion criteria (to match study goal, if applicable). * Standardization of numeric values with different units into same-unit values. If no dataset preparations are needed, write below this header: `# No dataset preparations are needed.` `# DESCRIPTIVE STATISTICS` * In light of our study goals and the hypothesis testing plan (see above "Research Goal" and "Hypothesis Testing Plan"), decide whether and which descriptive statistics are needed to be included in the paper and create a relevant table. For example: `## Table 0: "Descriptive statistics of height and age stratified by sex"` Write here the code to create a descriptive statistics dataframe `df0` and save it using: `df0.to_pickle('table_0.pkl')` If no descriptive statistics are needed, write: `# No descriptive statistics table is needed.` # PREPROCESSING Perform any preprocessing steps needed to further prepare the data for the analysis. For example, as applicable: * Standardization and normalization of numeric values (as needed). * Creating dummy variables for categorical variables (as needed). * Any other data preprocessing you deem relevant.

If no preprocessing is needed, write:
`# No preprocessing is needed, because <your reasons here>.`

ANALYSIS
Considering our "Research Goal" and "Hypothesis Testing Plan", decide on 1-3 tables (in addition to the above descriptive statistics, if any) we should create for our scientific paper. Typically, we should have at least one table for each hypothesis test.

For each such scientific table:
[a] Write a comment with a suggested table's caption.
Choose a caption that clearly describes the table's content and its purpose.
For example:
`## Table 1: "Test of association between age and risk of death, accounting for sex and race"`
Avoid generic captions such as `## Table 1: "Results of analysis"`.

[b] Perform analysis
- Perform appropriate analysis and/or statistical tests (see above our "Hypothesis Testing Plan").
- The statistical analysis should account for any relevant confounding variables, as applicable.
- Note that you may need to perform more than one test for each hypothesis.
- Try using inherent functionality and syntax provided in functions from the available Python packages (above) and avoid, as possible, manually
implementing generically available functionality.
For example, to include interactions in regression analysis (if applicable), use the "x * y" string syntax in statsmodels formulas.

[c] Create and save a dataframe for a scientific table
* Create a dataframe containing the data needed for the table (`df1`, `df2`, etc).
* Only include information that is relevant and suitable for inclusion in a scientific table.
* Nominal values should be accompanied by a measure of uncertainty (CI or STD and p-value).
* Exclude data not important to the research goal, or that are too technical.
* Make sure you do not repeat the same data in multiple tables.
* The table should have labels for the both the columns and the index (rows):
- Do not invent new names; just keep the original variable names from the dataset.
- As applicable, also keep unmodified any attr names from statistical test results.

Overall, the section should have the following structure:

ANALYSIS
Table 1: <your chosen table name here>
<write here the code to analyze the data and create a dataframe df1 for the table 1>
df1.to_pickle('table_1.pkl')

Table 2: <your chosen table name here>
etc, up to 3 tables.

SAVE ADDITIONAL RESULTS
At the end of the code, after completing the tables, create a dict containing any additional results you deem important to include in the scientific
paper, and save it to a pkl file 'additional_results.pkl'.

	For example: `additional_results = { 'Total number of observations': <xxx>, 'accuracy of regression model': <xxx>, # etc, any other results and important parameters that are not included in the tables } with open('additional_results.pkl', 'wb') as f: pickle.dump(additional_results, f) ` Avoid the following: Do not provide a sketch or pseudocode; write a complete runnable code including all '# HEADERS' sections. Do not create any graphics, figures or any plots. Do not send any presumed output examples. Avoid convoluted or indirect methods of data extraction and manipulation; Where possible, use direct attribute access for clarity and simplicity. Where possible, access dataframes using string-based column/index names, rather than integer-based column/index positions.
Product	Data analysis – code: Python code Data analysis – tables: Numerical data Data analysis – other results: Numerical data
Reviewer system prompt	You are a brilliant data scientist. You are writing a Python code to analyze data.
Reviewer mission prompt	(1) Check your Python code and return a bullet-point response addressing these points (as applicable): * DATASET PREPARATIONS: - Missing values. If applicable, did we deal with missing, unknown, or undefined values, or with special numeric values that stand for unknown/undefined (check the "Description of the Original Dataset" and "Output of the Data Exploration Code" for any such missing values)? - Units. If applicable, did we correctly standardize numeric values with different units into same-unit values? - Are we restricting the analysis to the correct data (based on the study goal)? * DESCRIPTIVE STATISTICS: If applicable: - did we correctly report descriptive statistics? Does the choice of variables for such statistics make sense for our study? - Is descriptive analysis done on the correct data (for example, before any data normalization steps)? * PREPROCESSING: Review the description of the data files (see above "Description of the Original Dataset") and the data exploration output (see above "Output of the Data Exploration Code"), then check the code for any data preprocessing steps that the code performs but are not needed, or that are needed but are not performed. * ANALYSIS:

	As applicable, check for any data analysis issues, including: - Analysis that should be performed on the preprocessed data is mistakenly performed on the original data. - Incorrect choice of statistical test. - Imperfect implementation of statistical tests. - Did we correctly chose the variables that best represent the tested hypothesis? - Are we accounting for relevant confounding variables (consult the "Description of the Original Dataset")? - In linear regression, if interactions terms are included: * did we remember to include the main effects? * did we use the `*` operator in statsmodels formula as recommended (as applicable, better use the `formula = "y ~ a * b"` string notation instead of trying to manually multiply the variables) - Any other statistical analysis issues. (2) Check the created pkl tables (provided above) and return a bullet-point response addressing these points: * Sensible numeric values: Check each numeric value in the tables and make sure it is sensible. For example: - If the table reports the mean of a variable, is the mean value sensible? - If the table reports CI, are the CI values flanking the mean? - Do values have correct signs? - Do you see any values that are not sensible (too large, too small)? * Measures of uncertainty: If the table reports nominal values (like for regression coefs), does it also report their measures of uncertainty (like p-value, CI, or STD, as applicable)? * Missing data in a table: Are we missing key variables in a given table? * Any other issues you find. (3) Based on your assessment above, return a Python Dict[str, str] mapping the issues you have noted above (dict keys) to specific suggested corrections/improvements in the code (dict values). For example: { "The model does not adequately account for confounding variables": "revise the code to add the following confounding variables ...", "A table is missing": "revise the code to add the following new table '<your suggested table caption>'", "Table <n> reports nominal values without measures of uncertainty": "revise the code to add STD and p-value.", } Try to be as specific as possible when describing the issues and proposed fixes. Include in the dict as many issues as you find. If you are sure that there are no issues, and the code and tables need no revision, then return an empty dict: `{}`.
Max code revisions	3

Data analysis code explanation

LLM	gpt-3.5-turbo-0613
Provided prior products	General description of dataset, Data file description, Data analysis - code
Performer system prompt	You are a scientist who needs to write explanation of the Data Analysis code.
Performer mission prompt	Please return a triple-backtick Latex Block explaining what the code above does. Do not provide a line-by-line explanation, rather provide a high-level explanation of the code in a language suitable for a Methods section of a research paper. Focus on analysis steps. There is no need to explain trivial parts, like reading/writing a file, etc. Also explain what does the code write into the "additional_results.pkl" file. Your explanation should be written in LaTeX, and should be enclosed within a LaTeX Code Block, like this: ```latex \section{Code Explanation} <your code explanation here> ``` Remember to enclose your explanation within a LaTeX Code Block, so that I can easily copy-paste it!
Product	Data analysis – code explanation: LaTex text
Max review iterations	N/A

Table design

LLM	gpt-4
Provided prior products	General description of dataset, Data file description, Research goal, Data analysis - code, *Data analysis - tables*
Performer system prompt	You are a brilliant data scientist. You are writing a Python code to analyze data.
Performer mission prompt	I would like to create latex tables for our scientific paper from the dataframes created in the code above ("table_?.pkl" files). I would like to convert these dataframes to latex tables, using the following 4 custom functions that I wrote: def to_latex_with_note(df, filename: str, caption: str, label: str, note: str = None, legend: Dict[str, str] = None, **kwargs): """ Converts a DataFrame to a LaTeX table with optional note and legend added below the table. Parameters: - df, filename, caption, label: as in `df.to_latex`. - note (optional): Additional note below the table. - legend (optional): Dictionary mapping abbreviations to full names. - **kwargs: Additional arguments for `df.to_latex`. Returns: - None: Outputs LaTeX file. """ def format_p_value(x): returns "{:.3g}".format(x) if x >= 1e-06 else "<1e-06" def is_str_in_df(df: pd.DataFrame, s: str): return any(s in level for level in getattr(df.index, 'levels', [df.index]) + getattr(df.columns, 'levels', [df.columns])) AbbrToNameDef = Dict[Any, Tuple[Optional[str], Optional[str]]] def split_mapping(abbrs_to_names_and_definitions: AbbrToNameDef): abbrs_to_names = {abbr: name for abbr, (name, definition) in abbrs_to_names_and_definitions.items() if name is not None} names_to_definitions = {name or abbr: definition for abbr, (name, definition) in abbrs_to_names_and_definitions.items() if definition is not None} return abbrs_to_names, names_to_definitions Please write a complete Python code that uses the above functions to convert our dataframes to latex tables suitable for our scientific paper. Follow these instructions: Rename column and row names: You should provide a new name to any column or row label that is abbreviated or technical, or that is otherwise not self-explanatory.

Full definitions: You should provide an optional full definition for any name (or new name) that satisfies any of the following:
- Remains abbreviated, or not self-explanatory, even after renaming
- Is an ordinal/categorical value that requires clarification of the meaning of each value.
- Contains possibly unclear notation, like '*' or ':'
- Is a numeric value that has units, that need to be specified.

To avoid re-naming mistakes, I strongly suggest you define for each table a dictionary, `mapping: AbbrToNameDef`, which maps any original column and
row labels that are abbreviated or not self-explanatory to an optional new name, and an optional definition.
If different tables share several common labels, then you can build these table-specific mappings from a `shared_mapping`. See example below.

Overall, the code must have the following structure:

```
# IMPORT
import pandas as pd
from my_utils import to_latex_with_note, format_p_value, is_str_in_df, split_mapping, AbbrToNameDef

# PREPARATION FOR ALL TABLES

< As applicable, define a shared mapping for labels that are common to all tables. For example: >

shared_mapping: AbbrToNameDef = {
 'AvgAge': ('Avg. Age', 'Average age, years'),
 'BT': ('Body Temperature', '1: Normal, 2: High, 3: Very High'),
 'W': ('Weight', 'Participant weight, kg'),
 'MRSA': (None, 'Infected with Methicillin-resistant Staphylococcus aureus, 1: Yes, 0: No'),
 ...: (..., ...),
}
< This is of course just an example. Consult with the "Description of the Original Dataset" and the "Data Analysis Code" for choosing the common
labels and their appropriate scientific names and definitions. >

# TABLE 0:
df = pd.read_pickle('table_0.pkl')

# FORMAT VALUES <include this sub-section only as applicable>
< Rename technical values to scientifically-suitable values. For example: >
df['MRSA'] = df['MRSA'].apply(lambda x: 'Yes' if x == 1 else 'No')

< If the table has P-values from statistical tests, format them with `format_p_value`. For example: >
df['PV'] = df['PV'].apply(format_p_value)

# RENAME ROWS AND COLUMNS <include this sub-section only as applicable>
< Rename any abbreviated or not self-explanatory table labels to scientifically-suitable names. >
< Use the `shared_mapping` if applicable. For example: >
mapping = {k: v for k, v in shared_mapping.items() if is_str_in_df(df, k)}
mapping |= {
```

	'PV': ('P-value', None), 'CI': (None, '95% Confidence Interval'), 'Sex_Age': ('Age * Sex', 'Interaction term between Age and Sex'), } abbrs_to_names, legend = split_mapping(mapping) df = df.rename(columns=abbrs_to_names, index=abbrs_to_names) # Save as latex: to_latex_with_note(df, 'table_1.tex', caption="<choose a caption suitable for a table in a scientific paper>", label='table:<chosen table label>', note="<If needed, add a note to provide any additional information that is not captured in the caption>", legend=legend) # TABLE <?>: < etc, all 'table_?.pkl' files > ``` Avoid the following: Do not provide a sketch or pseudocode; write a complete runnable code including all '# HEADERS' sections. Do not create any graphics, figures or any plots. Do not send any presumed output examples.
Product	Tables design – code: Python code Tables design – tables: LaTex text
Reviewer system prompt	Not required - LLM review is not performed for the “Table design” step, as its output is simply a style conversion of the already-created tables, which can be thoroughly inspected by the rule-based reviewer.
Reviewer mission prompt	
Max code revisions	3

Title and abstract draft

LLM	gpt-3.5-turbo-0613
Provided prior products	General description of dataset, Data analysis - code, Data analysis - other results, Tables design - tables
Performer system prompt	You are a data-scientist with experience writing accurate scientific research papers. You will write a scientific article for the journal Nature Communications, following the instructions below: 1. Write the article section by section: Abstract, Introduction, Results, Discussion, and Methods. 2. Write every section of the article in scientific language, in `.tex` format. 3. Write the article in a way that is fully consistent with the scientific results we have.
Performer mission prompt	Based on the material provided above ("Overall Description of the Dataset", "Data Analysis Code", "Tables of the Paper", "Additional Results (additional_results.pkl)"), please write only the title and abstract for a research paper for a Nature Communications article. Do not write any other parts! The Title should: * be short and meaningful. * convey the main message, focusing on discovery not on methodology nor on the data source. * not include punctuation marks, such as ":,;" characters. The Abstract should provide a concise, interesting to read, single-paragraph summary of the paper, with the following structure: * short statement of the subject and its importance. * description of the research gap/question/motivation. * short, non-technical, description of the dataset used and a non-technical explanation of the methodology. * summary of each of the main results. It should summarize each key result which is evident from the tables, but without referring to specific numeric values from the tables. * statement of limitations and implications. Write in tex format, escaping any math or symbols that needs tex escapes. The title and abstract for a research paper should be enclosed within triple-backtick "latex" code block, like this: ```latex \title{<your latex-formatted paper title here>} \begin{abstract} <your latex-formatted abstract here> \end{abstract} ```
Product	Title & abstract draft: LaTex text
Reviewer system prompt	You are a reviewer for a scientist who is writing a scientific paper about their data analysis results. Your job is to provide constructive bullet-point feedback. We will write each section of the research paper separately. If you feel that the paper section does not need further improvements, you should reply only with:

	"The title and abstract for a research paper does not require any changes".
Reviewer mission prompt	Please provide a bullet-point list of constructive feedback on the above Title and Abstract for my paper. Do not provide positive feedback, only provide actionable instructions for improvements in bullet points. In particular, make sure that the section is correctly grounded in the information provided above. If you find any inconsistencies or discrepancies, please mention them explicitly in your feedback. The Title should: * be short and meaningful. * convey the main message, focusing on discovery not on methodology nor on the data source. * not include punctuation marks, such as ":,;" characters. The Abstract should provide a concise, interesting to read, single-paragraph summary of the paper, with the following structure: * short statement of the subject and its importance. * description of the research gap/question/motivation. * short, non-technical, description of the dataset used and a non-technical explanation of the methodology. * summary of each of the main results. It should summarize each key result which is evident from the tables, but without referring to specific numeric values from the tables. * statement of limitations and implications. You should only provide feedback on the Title and Abstract. Do not provide feedback on other sections or other parts of the paper, like LaTex Tables or Python code, provided above. If you don't see any flaws, respond solely with "The title and abstract for a research paper does not require any changes". IMPORTANT: You should EITHER provide bullet-point feedback, or respond solely with "The title and abstract for a research paper does not require any changes"; If you chose to provide bullet-point feedback then DO NOT include "The title and abstract for a research paper does not require any changes".
Max review iterations	1

Literature search II

LLM	gpt-3.5-turbo-0613
Provided prior products	General description of dataset, Data file description, Research goal, Hypothesis testing plan, Title and abstract draft
Performer system prompt	You are a scientist who needs to write literature search queries.
Performer mission prompt	Please write literature-search queries that we can use to search for papers related to our study. You would need to compose search queries to identify prior papers covering these 4 areas: "background": papers that provide background on the overall subject of our study "dataset": papers that use the same or similar datasets as in our study "methods": papers that use the same or similar methods as in our study "results": papers that report results similar to our study Return your answer as a `Dict[str, List[str]]`, where the keys are the 4 areas noted above, and the values are lists of query string. Each individual query should be a string with up to 5-10 words. For example, for a study reporting waning of the efficacy of the covid-19 BNT162b2 vaccine based on analysis of the "United Kingdom National Core Data (UK-NCD)", the queries could be: { "background": ['SARS-CoV2 spread', 'covid-19 global impact', 'covid-19 vaccine'] "dataset": ['The UK-NCD dataset', 'covid-19 vaccine efficacy dataset'] "methods": ['covid-19 vaccine efficacy analysis', 'kaplan-meier survival analysis'] "results": ['covid-19 vaccine efficacy', 'covid-19 vaccine efficacy over time', 'covid-19 vaccine waning'] }
Product	Literature search II – queries: Structured text Literature search II – citations: Citations
Max review iterations	N/A

Results

LLM	gpt-3.5-turbo-0613
Provided prior products	General description of dataset, Data file description, Data analysis code, Data analysis other results, Latex tables design tables, Title and abstract draft
Performer system prompt	You are a data-scientist with experience writing accurate scientific research papers. You will [...] with the scientific results we have.
Performer mission prompt	Based on the material provided above ("Title and Abstract", "Description of the Original Dataset (with hypertargets)", "Data Analysis Code", "Tables of the Paper with hypertargets", "Additional Results (additional_results.pkl) with hypertargets"), please write only the `Results` section for a Nature Communications article. Do not write any other parts! Use the following guidelines when writing the Results: * Include 3-4 paragraphs, each focusing on one of the Tables: You should typically have a separate paragraph describing each of the Tables. In each such paragraph, indicate the motivation/question for the analysis, the methodology, and only then describe the results. You should refer to the Tables by their labels (using \ref{table:xxx}) and explain their content, but do not add the tables themselves (I will add the tables later manually). * Story-like flow: It is often nice to have a story-like flow between the paragraphs, so that the reader can follow the analysis process with emphasis on the reasoning/motivation behind each analysis step. For example, the first sentence of each paragraph can be a story-guiding sentences like: "First, to understand whether xxx, we conducted a simple analysis of ..."; "Then, to test yyy, we performed a ..."; "Finally, to further verify the effect of zzz, we tested whether ...". * Conclude with a summary of the results: You can summarize the results at the end, with a sentence like: "In summary, these results show ...", or "Taken together, these results suggest ...". IMPORTANT NOTE: Your summary SHOULD NOT include a discussion of conclusions, implications, limitations, or of future work. (These will be added later as part the Discussion section, not the Results section). * Numeric values: - Sources: You can extract numeric values from the above provided sources: "Tables of the Paper with hypertargets", "Additional Results (additional_results.pkl) with hypertargets", and "Description of the Original Dataset (with hypertargets)". All numeric values in these sources have a \hypertarget with a unique label. - Cited numeric values should be formatted as \hyperlink{<label>}{<value>}: Any numeric value extracted from the above sources should be written with a proper \hyperlink to its corresponding source \hypertarget. - Dependent values should be calculated using the \num command. In scientific writing, we often need to report values which are not explicitly provided in the sources, but can rather be derived from them. For

example: changing units, calculating differences, transforming regression coefficients into odds ratios, etc (see examples below).

To derive such dependent values, please use the \num{<formula>, "explanation"} command.
The <formula> contains a calculation, which will be automatically replaced with its result upon pdf compilation.
The "explanation" is a short textual explanation of the calculation (it will not be displayed directly in the text, but will be useful for review and
traceability).

- Toy example for citing and calculating numeric values:

Suppose our provided source data includes:
```
No-treatment response: \hypertarget{Z1a}{0.65}
With-treatment response: \hypertarget{Z2a}{0.87}

Treatment regression:
coef = \hypertarget{Z3a}{0.17}, STD = \hypertarget{Z3b}{0.072}, pvalue =
<\hypertarget{Z3c}{1e-6}
```

Then, here are some examples of proper ways to report these provided source values:
```
The no-treatment control group had a response of \hyperlink{Z1a}{0.65} while the with-treatment group had a response of \hyperlink{Z2a}{0.87}.

The regression coefficient for the treatment was \hyperlink{Z3a}{0.17} with a standard deviation of \hyperlink{Z3b}{0.072} (P-value: <
\hyperlink{Z3c}{1e-6}).
```

And are some examples of proper ways to calculate dependent values, using the \num command:
```
The difference in response was \num{\hyperlink{Z2a}{0.87} - \hyperlink{Z1a}{0.65}, "Difference between responses with and without treatment"}.

The treatment odds ratio was \num{exp(\hyperlink{Z3a}{0.17}), "Translating the treatment regression coefficient to odds ratio"} (CI:
\num{exp(\hyperlink{Z3a}{0.17} - 1.96 * \hyperlink{Z3b}{0.072}), "low CI for treatment odds ratio, assuming normality"},
\num{exp(\hyperlink{Z3a}{0.17} + 1.96 * \hyperlink{Z3b}{0.072}), "high CI for treatment odds ratio, assuming normality"}).
```

* Accuracy:
Make sure that you are only mentioning details that are explicitly found within the Tables and Numerical Values.

* Unknown values:
If we need to include a numeric value that is not explicitly given in the Tables or "Additional Results (additional_results.pkl) with hypertargets", and cannot be derived

	from them using the \num command, then indicate `[unknown]` instead of the numeric value. For example: ``` The no-treatment response was \hyperlink{Z1a}{0.65} (STD: [unknown]). ``` Write in tex format, escaping any math or symbols that needs tex escapes. The `Results` section should be enclosed within triple-backtick "latex" code block, like this: ```latex \section{<section name>} <your latex-formatted writing here> ```
Product	Results: LaTex text
Reviewer system prompt	You are a reviewer for a scientist who is writing a scientific paper about their data analysis results. Your job is to provide constructive bullet-point feedback. We will write each section of the research paper separately. If you feel that the paper section does not need further improvements, you should reply only with: "The Results section does not require any changes".
Reviewer mission prompt	Please provide a bullet-point list of constructive feedback on the above Results for my paper. Do not provide positive feedback, only provide actionable instructions for improvements in bullet points. In particular, make sure that the section is correctly grounded in the information provided above. If you find any inconsistencies or discrepancies, please mention them explicitly in your feedback. Specifically, pay attention to: whether the Results section contains only information that is explicitly extracted from the "Tables of the Paper" and "Additional Results (additional_results.pkl)" provided above. Compare the numbers in the Results section with the numbers in the Tables and Numerical Values and explicitly mention any discrepancies that need to be fixed. Do not suggest adding missing information, or stating whats missing from the Tables and Numerical Values, only suggest changes that are relevant to the Results section itself and that are supported by the given Tables and Numerical Values. Do not suggest changes to the Results section that may require data not available in the the Tables and Numerical Values. You should only provide feedback on the Results. Do not provide feedback on other sections or other parts of the paper, like LaTex Tables or Python code, provided above. If you don't see any flaws, respond solely with "The Results section does not require any changes". IMPORTANT: You should EITHER provide bullet-point feedback, or respond solely with "The Results section does not require any changes"; If you chose to

	provide bullet-point feedback then DO NOT include "The Results section does not require any changes".
Max review iterations	1

Title and abstract

LLM	gpt-3.5-turbo-0613
Provided prior products	General description of dataset, Title and abstract draft, Literature search II - citations (Background, Dataset and Results scopes), Results
Performer system prompt	You are a data-scientist with experience writing accurate scientific research papers. You will write a scientific article for the journal Nature Communications, following the instructions below: 1. Write the article section by section: Abstract, Introduction, Results, Discussion, and Methods. 2. Write every section of the article in scientific language, in `.tex` format. 3. Write the article in a way that is fully consistent with the scientific results we have.
Performer mission prompt	Bases on the material provided above ("Overall Description of the Dataset", "Results Section of the Paper", "Background-related Literature Search", "Dataset-related Literature Search", "Results-related Literature Search", "Title and Abstract"), please help me improve the title and abstract for a Nature Communications research paper. The Title should: * be short and meaningful. * convey the main message, focusing on discovery not on methodology nor on the data source. * not include punctuation marks, such as ":,;" characters. The Abstract should provide a concise, interesting to read, single-paragraph summary of the paper, with the following structure: * short statement of the subject and its importance. * description of the research gap/question/motivation. * short, non-technical, description of the dataset used and a non-technical explanation of the methodology. * summary of each of the main results. It should summarize each key result which is evident from the tables, but without referring to specific numeric values from the tables. * statement of limitations and implications. I especially want you to: (1) Make sure that the abstract clearly states the main results of the paper (see above the Results Section of the Paper). (2) Make sure that the abstract correctly defines the literature gap/question/motivation (see above Literature Searches for list of related papers). Write in tex format, escaping any math or symbols that needs tex escapes. The title and abstract for a research paper should be enclosed within triple-backtick "latex" code block, like this: ```latex \title{<your latex-formatted paper title here>} \begin{abstract} <your latex-formatted abstract here> \end{abstract}```
Product	Title & abstract: LaTex text
Max review iterations	0

Methods

LLM	gpt-3.5-turbo-0613
Provided prior products	General description of dataset, Data file description, Research goal, Data analysis - code, Results, Title and abstract
Performer system prompt	You are a data-scientist with experience writing accurate scientific research papers. You will write a scientific article for the journal Nature Communications, following the instructions below: 1. Write the article section by section: Abstract, Introduction, Results, Discussion, and Methods. 2. Write every section of the article in scientific language, in `.tex` format. 3. Write the article in a way that is fully consistent with the scientific results we have.
Performer mission prompt	Based on the material provided above ("Description of the Original Dataset", "Research Goal", "Data Analysis Code", "Title and Abstract"), please write only the Methods section for a Nature Communications article. Do not write any other parts! The Methods section should be enclosed within triple-backtick "latex" code block and have 3 subsections, as follows: ```latex \section{Methods} \subsection{Data Source} - Describe our data sources (see above "Description of the Original Dataset") \subsection{Data Preprocessing} - Describe preprocessing of the data done by the Python code (see above "Data Analysis Code"). - Do not include preprocessing steps that were not performed by the code. - Do not include preprocessing steps that were performed by the code, but were not used as basis for further analysis affecting the result output. \subsection{Data Analysis} - Describe each of the specific analysis steps performed by the Python code to yield the results. - Do not be over technical. - Do not enumerate the steps as a list; instead, describe the steps in a narrative form. ``` Throughout the Methods section, do NOT include any of the following: - Missing steps not done by the code. - Specific version of software packages, file names, column names. - Names of package functions (e.g., do not say "We used sklearn.linear_model.LinearRegression", say instead "We used a linear regression model") - URLs, links or references. Remember to enclose the Methods section within triple-backtick "latex" code block.
Product	Methods: LaTex text
Max review iterations	0

Introduction

LLM	gpt-4
Provided prior products	General description of dataset, Literature search II - citations (Background, Dataset, Methods and Results scopes), Results, Title and abstract, Methods
Performer system prompt	You are a data-scientist with experience writing accurate scientific research papers. You will write a scientific article for the journal Nature Communications, following the instructions below: 1. Write the article section by section: Abstract, Introduction, Results, Discussion, and Methods. 2. Write every section of the article in scientific language, in `.tex` format. 3. Write the article in a way that is fully consistent with the scientific results we have.
Performer mission prompt	Based on the material provided above ("Overall Description of the Dataset", "Title and Abstract", "Background-related Literature Search", "Results-related Literature Search", "Dataset-related Literature Search", "Methods-related Literature Search", "Methods Section of the Paper", "Results Section of the Paper"), please write only the Introduction section for a Nature Communications article. Do not write any other parts! The introduction should be interesting and pique your reader's interest. It should be written while citing relevant papers from the Literature Searches above. Specifically, the introduction should follow the following multi-paragraph structure: * Introduce the topic of the paper and why it is important (cite relevant papers from the above "Background-related Literature Search"). * Explain what was already done and known on the topic, and what is then the research gap/question (cite relevant papers from the above "Results-related Literature Search"). If there is only a minor gap, you can use language such as "Yet, it is still unclear ...", "However, less is known about ...", etc. * State how the current paper addresses this gap/question (cite relevant papers from the above "Dataset-related Literature Search" and "Results-related Literature Search"). * Outline the methodological procedure and briefly state the main findings (cite relevant papers from the above "Methods-related Literature Search") Note: each of these paragraphs should be 5-6 sentence long. Do not just write short paragraphs with less than 5 sentences! Citations should be added in the following format: \cite{paper_id}. Do not add a \section{References} section, I will add it later manually. Note that it is not advisable to write about limitations, implications, or impact in the introduction. Write in tex format, escaping any math or symbols that needs tex escapes. The Introduction section should be enclosed within triple-backtick "latex" code block, like this: ```latex \section{<section name>}

	<your latex-formatted writing here> ```
Product	Introduction: LaTex text
Reviewer system prompt	You are a reviewer for a scientist who is writing a scientific paper about their data analysis results. Your job is to provide constructive bullet-point feedback. We will write each section of the research paper separately. If you feel that the paper section does not need further improvements, you should reply only with: "The Introduction section does not require any changes".
Reviewer mission prompt	Please provide a bullet-point list of constructive feedback on the above Introduction for my paper. Do not provide positive feedback, only provide actionable instructions for improvements in bullet points. In particular, make sure that the section is correctly grounded in the information provided above. If you find any inconsistencies or discrepancies, please mention them explicitly in your feedback. Also, please suggest if you see any specific additional citations that are adequate to include (from the Literature Searches above). You should only provide feedback on the Introduction. Do not provide feedback on other sections or other parts of the paper, like LaTex Tables or Python code, provided above. If you don't see any flaws, respond solely with "The Introduction section does not require any changes". IMPORTANT: You should EITHER provide bullet-point feedback, or respond solely with "The Introduction section does not require any changes"; If you chose to provide bullet-point feedback then DO NOT include "The Introduction section does not require any changes".
Max review iterations	1

Discussion

LLM	gpt-4
Provided prior products	General description of dataset, Literature search II - (Background, and Results scopes), Results, Title and abstract, Methods, Introduction
Performer system prompt	You are a data-scientist with experience writing accurate scientific research papers. You will write a scientific article for the journal Nature Communications, following the instructions below: 1. Write the article section by section: Abstract, Introduction, Results, Discussion, and Methods. 2. Write every section of the article in scientific language, in `.tex` format. 3. Write the article in a way that is fully consistent with the scientific results we have.
Performer mission prompt	Based on the material provided above ("Overall Description of the Dataset", "Title and Abstract", "Background-related Literature Search", "Results-related Literature Search", "Introduction Section of the Paper", "Methods Section of the Paper", "Results Section of the Paper"), please write only the Discussion section for a Nature Communications article. Do not write any other parts! The Discussion section should follow the following structure: * Recap the subject of the study (cite relevant papers from the above "Background-related Literature Search"). * Recap our methodology (see "Methods" section above) and the main results (see "Results Section of the Paper" above), and compare them to the results from prior literature (see above "Results-related Literature Search"). * Discuss the limitations of the study. * End with a concluding paragraph summarizing the main results, their implications and impact, and future directions. Citations should be added in the following format: \cite{paper_id}. Do not add a \section{References} section, I will add it later manually. Write in tex format, escaping any math or symbols that needs tex escapes. The Discussion section should be enclosed within triple-backtick "latex" code block, like this: ```latex \section{<section name>} <your latex-formatted writing here> ```
Product	Discussion: LaTex product
Reviewer system prompt	You are a reviewer for a scientist who is writing a scientific paper about their data analysis results. Your job is to provide constructive bullet-point feedback. We will write each section of the research paper separately. If you feel that the paper section does not need further improvements, you should reply only with: "The Discussion section does not require any changes".
Reviewer mission prompt	Please provide a bullet-point list of constructive feedback on the above Discussion for my paper. Do not provide positive feedback, only provide actionable instructions for improvements in bullet points.

	In particular, make sure that the section is correctly grounded in the information provided above. If you find any inconsistencies or discrepancies, please mention them explicitly in your feedback. Also, please suggest if you see any specific additional citations that are adequate to include (from the Literature Searches above). You should only provide feedback on the Discussion. Do not provide feedback on other sections or other parts of the paper, like LaTex Tables or Python code, provided above. If you don't see any flaws, respond solely with "The Discussion section does not require any changes". IMPORTANT: You should EITHER provide bullet-point feedback, or respond solely with "The Discussion section does not require any changes"; If you chose to provide bullet-point feedback then DO NOT include "The Discussion section does not require any changes".
Max review iterations	1

Table S2. Research Goals generated by gpt-3.5-turbo and Llama 2-70B. Green highlighting indicates that a factor is a dataset feature, red highlighting indicates that this information is not a dataset feature. Results are summarized in Fig. S3A.

1 ChatGPT	Research Goal: To investigate the association between physical activity and the likelihood of having high blood pressure, and whether this association is moderated by age. Hypothesis: There is a negative association between physical activity and the likelihood of having high blood pressure (Hypertension), and this association is moderated by age, such that the association is stronger among younger individuals compared to older individuals.
2 ChatGPT	Research Goal: To investigate the association between physical activity and the incidence of diabetes among adults in the dataset. Hypothesis: There is a negative relationship between physical activity and the risk of diabetes, such that individuals who engage in regular physical activity are less likely to have diabetes compared to those who do not engage in regular physical activity. This relationship will be moderated by age, such that the protective effect of physical activity against diabetes will be stronger among younger adults compared to older adults.
3 ChatGPT	Research Goal: To investigate the relationship between physical activity and diabetes among different age groups, while controlling for potential confounding variables. Additionally, to explore potential moderation effects of gender and education on this relationship. Hypothesis: Higher levels of physical activity are associated with a lower prevalence of diabetes, and this association is stronger among older individuals compared to younger individuals. Moreover, the association between physical activity and diabetes may be moderated by gender and education, with females and individuals with higher education experiencing a stronger protective effect of physical activity against diabetes.

4 ChatGPT	Research Goal: To investigate the relationship between physical activity and diabetes status, and whether this relationship is moderated by age. Hypothesis: There is a positive association between physical activity and diabetes status, such that individuals who engage in regular physical activity are less likely to have diabetes compared to those who are physically inactive. Furthermore, we hypothesize that this association will be stronger among younger individuals compared to older individuals,indicating a moderation effect of age on the relationship between physical activity and diabetes status.
5 ChatGPT	Research Goal: To examine the association between physical activity and the prevalence of diabetes, while considering the potential moderating effect of BMI. Hypothesis: There is a significant interaction effect between physical activity and BMI on the prevalence of diabetes, such that the protective effect of physical activity is stronger among individuals with lower BMI compared to those with higher BMI.
6 ChatGPT	Research Goal: To examine the relationship between fruit and vegetable consumption and the risk of diabetes, taking into account the potential interaction effect of gender. Hypothesis: Higher levels of fruit and vegetable consumption will be associated with a decreased risk of diabetes, and this association will be stronger among females compared to males.
7 ChatGPT	Research Goal: To examine the association between physical activity and the risk of diabetes, while considering the potential moderating effect of age. Hypothesis: The relationship between physical activity and the risk of diabetes will be stronger among younger individuals compared to older individuals.

8 ChatGPT	Research Goal: To examine the association between lifestyle factors and the presence of diabetes among individuals in the dataset. Hypothesis: There is a significant association between physical activity and the presence of diabetes, such that individuals who engage in regular physical activity are less likely to have diabetes compared to those who are physically inactive.
9 ChatGPT	Research Goal: Investigate the association between physical activity and the likelihood of having high blood pressure, and whether this association is moderated by the presence of diabetes. Hypothesis: Individuals with diabetes who engage in regular physical activity are less likely to have high blood pressure compared to individuals with diabetes who do not engage in regular physical activity. Furthermore, this association is stronger among individuals with diabetes who have a higher body mass index (BMI).
10 ChatGPT	Research Goal: To examine the association between fruit and vegetable consumption and the risk of high blood pressure, while considering the potential moderating effect of education level. Hypothesis: Higher levels of fruit and vegetable consumption will be associated with a lower risk of high blood pressure, and this association will be stronger among individuals with higher education levels.
1 Llama	Research Goal: To evaluate the effect of lower education level on glycemic control and quality of life among patients with diabetes, while considering potentially confounding factors and examining plausible mediation routes. Hypothesis: Individuals with lower education levels will exhibit inferior glycemic control and decreased quality of life relative to those with higher education levels. Furthermore, we posit that these differences will be partly attributable to variations in healthcare accessibility (including preventive care utilization, medical visit frequency, and insurance coverage) and harmful behavior patterns (principally smoking prevalence and heavy alcohol consumption). By delving into these causal pathways, the study's findings may

	contribute to the development of evidence-based strategies aimed at mitigating health disparities in disadvantaged diabetic populations.
2 Llama	Research Goal: *** Investigate the impact of social isolation on cognitive decline in older adults, taking into account potential confounding factors and identifying specific cognitive domains affected. *** Hypothesis: *** Among older adults, social isolation will be associated with accelerated cognitive decline in memory and executive functions, controlling for relevant confounders such as socioeconomic status, health conditions, lifestyle choices, urban vs. rural residency, mobility limitations, sensory impairments, and technology usage patterns. Specifically, we predict that older adults who are socially isolated, particularly those living alone and facing barriers to social engagement, will exhibit steeper declines in cognitive performance over time relative to their less isolated counterparts. Moreover, we propose that the relation between social isolation and cognitive decline will be mediated by reduced cognitive stimulation and decreased social support. **physical activity levels. Furthermore,** we expect to identify distinct cognitive profiles among older adults based on their social isolation status and explore how these profiles differ in terms of cognitive performance and decline over time. **/sprinkle throughout document**
3 Llama	Research Goal: To examine the relationship between social support and reduced symptoms of depression in individuals with diabetes, while considering the role of frequent walking or jogging. Hypothesis: Individuals with diabetes who receive social support from family, friends, and community resources will exhibit lower levels of depressive symptoms, specifically Beck Depression Inventory scores, after a six-month period, compared to those who do not receive such support. We predict that the positive effect of social support on reducing symptoms of depression will be most pronounced among those who frequently walk or jog. Furthermore, we expect that the combination of social support and frequent walking or jogging will produce a synergistic effect, leading to greater reductions in depressive symptoms compared to either intervention alone.

4 Llama	Research Goal: To examine the relationship between social support and mental health outcomes among adults with type 2 diabetes, and to identify potential moderators and mediators of this association. Hypothesis: Higher levels of social support will be associated with lower symptoms of depression and anxiety among adults with type 2 diabetes. We expect that this relationship will be strongest among individuals with lower levels of education, as they may face greater barriers in accessing healthcare resources and social support networks. Additionally, we will explore whether the association between social support and mental health outcomes varies by duration of diabetes diagnosis, income, gender, and comorbidities. We also test possible interaction terms between social support and these variables to uncover nuanced patterns in the data. Furthermore, we propose that increased access to emotional support, tangible aid, and informational support mediates the relationship between social support and mental health outcomes.
5 Llama	Research Goal: To investigate the relationship between social support and mental health outcomes in individuals with type 2 diabetes. Hypothesis: Individuals with type 2 diabetes who receive high levels of emotional support will have better mental health outcomes, as measured by General Health (GenHlth) and Mental Health (MentHlth), compared to those who receive low levels of emotional support. Additionally, we predict that tangible assistance will have a significant impact on MentHlth, but not GenHlth. We will also explore whether the relationship between social support and mental health outcomes varies by age, sex, and income.
6 Llama	Research Goal: To investigate the relationship between social isolation and cognitive decline in older adults with diabetes, specifically examining the role of glycemic control and potential mediating factors. Hypothesis: Older adults with diabetes who are socially isolated will experience greater cognitive decline in memory and executive function over a two-year period compared to those who are not socially isolated, after controlling for potential confounding variables. Furthermore, we predict that the association between social isolation and cognitive decline will be strongest among

	individuals with poor glycemic control, defined as HbA1c levels above 7%. We will also examine whether depression, anxiety, and sleep quality serve as mediators of the relationship between social isolation and cognitive decline.
7 Llama	Research Goal: To investigate the relationship between healthcare accessibility and health outcomes among individuals with diabetes, with a focus on understanding the temporal dynamics and potential mediating factors that influence this association. Hypothesis: We hypothesize that increased access to healthcare will lead to improved health outcomes over time among individuals with diabetes, and that this relationship will be influenced by various mediating factors, including health literacy, patient activation, and social support. Specifically, we expect that individuals with diabetes who have access to healthcare will have better HbA1c control, blood pressure management, lipid profiles, and body mass index compared to those without access, and that these differences will persist over time. Furthermore, we anticipate that the association between healthcare accessibility and health outcomes will vary across different age groups, with younger adults facing distinct barriers to healthcare access that may affect their health outcomes differently than older adults. By examining these relationships and identifying key modifiable factors,our study aims to provide actionable insights into optimizing healthcare delivery and improving health outcomes for individuals with diabetes.
8 Llama	Research Goal: ======= Our goal is to investigate the impact of improved cardiovascular health on healthcare utilization among patients with diabetes, specifically examining whether achieving optimal cardiovascular health metrics (e.g., normal blood pressure, cholesterol levels, and body mass index) reduces the likelihood of physician visits and, consequently, healthcare expenditure. This knowledge gap is significant since diabetes management involves long-term monitoring and treatment, and understanding the effectiveness of various interventions can help policymakers allocate resources efficiently and improve patient outcomes. Hypothesis: ===== We hypothesize that, after adjusting for relevant confounders (age, sex, education, income, smoking

	status, and other appropriate variables), a 1 mmHg decrease in systolic blood pressure will correspond to a 2% reduction in the total number of physician visits in the past year among patients with diabetes. Additionally, we expect that achieving optimal cardiovascular health markers (e.g., a combination of normal blood pressure, cholesterol levels, and body mass index) will be associated with a significantly lower likelihood of having at least one physician visit in the past year, indicating improved health outcomes and potentially reduced healthcare expenditure. This study's results can contribute to the development of evidence-based policies and clinical guidelines tailored to populations with diabetes, ultimately improving patient wellbeing while containing healthcare costs.
9 Llama	Research Goal: ======= To examine the relationship between social support from family members and mental health outcomes among individuals with diabetes, while controlling for potential confounding variables. Hypothesis: ====== We hypothesize that individuals with diabetes who report higher levels of social support from family members will exhibit better mental health outcomes, including reduced symptoms of stress, anxiety, and depression, compared to those who receive less social support from family members. Furthermore, we expect that an increase in social support from family members over time will correspond with improvements in mental health outcomes.
10 Llama	Research Goal: To investigate the relationship between specific types of social support and mental health outcomes among adults with Type 2 diabetes, while controlling for potential confounders and exploring potential mediating mechanisms. Hypothesis: We hypothesize that higher levels of emotional support, practical assistance, and informational support will be associated with improved mental health outcomes (lower levels of depression, anxiety, and stress) in adults with Type 2 diabetes. Furthermore, we expect that these relationships will be moderated by factors such as age, sex, and disease duration, and that certain psychological processes (e.g., coping strategies, self-esteem) and behavioral factors (e.g., medication adherence, physical activity) will play a mediating role

	in the relationship between social support and mental health outcomes.

Table S3. Remaining Topics and findings of papers produced for “Health Indicators” dataset (A4-5) and “Social Network” dataset (B4-5).

A4	**Topic:** Diabetes & physical activity **Title:** “Physical Activity, BMI, and Age: Impacts on Diabetes Risk in a National Study” **Conclusion:** “[...] lower physical activity levels, higher BMI, and older age were associated with an increased likelihood of developing diabetes. [...] revealed significant interaction effects between physical activity and BMI, as well as physical activity and age, [...]”
A5	**Topic:** Physical activity & chronic diseases in diabetic population **Title:** “Insights into the Association between Physical Activity and Chronic Health Conditions in Individuals with Diabetes” **Conclusion:** “[...] high blood pressure, high cholesterol, and coronary heart disease. [...] significant negative associations between physical activity and these chronic health conditions, [...]”
B4	**Topic:** State size, party affiliation & in- and outgoing twitter interactions **Title:** “Patterns and Influential Factors in Twitter Interactions among U.S. Congress Members” **Conclusion:** “While party affiliation shows some relationship with interaction patterns, the influence of state representation is less pronounced”
B5	**Topic:** State size, party affiliation & twitter interactions **Title:** “Understanding Twitter Dynamics and Influence among Members of the US Congress” **Conclusion:** “[...] highlight differences in Twitter engagement between Party and Chamber, [...] suggesting that the size of State representation plays a role in fostering online engagement among Congress members.”

Table S4. Key features of data analysis of papers created based on the “Health Indicators” and “Social Network” datasets.

	Paper A1	**Paper A2**	**Paper A3**	**Paper A4**	**Paper A5**
Descriptive statistics	Stratified by Diabetes	Stratified by Diabetes	Stratified 2x2 by Fruit & Vegetable consumption	Stratified by Diabetes	Stratified by Diabetes
Focus group	-	Diabetes=1	-	-	Diabetes=1
Statistical test	2x Logistic regressions	1x Linear regression	1x Logistic regression	1x Logistic regression	3x Logistic regressions
Confounding variables	Yes	Yes	Yes	Yes	Yes
Interactions	Yes	No	No	Yes	No
Number of tables	3	2	2	2	4

	Paper B1	**Paper B2**	**Paper B3**	**Paper B4**	**Paper B5**
Descriptive statistics	Interactions as edges stratified by chamber	-	-	-	Interactions per node stratified by chamber and party
Focus group	Edges	Nodes	Edges	Nodes	Nodes
Statistical test	1x Chi-Square	1x Chi-Square 1x Anova	2x Chi-Square	2x Linear regression	1x Anova
Confounding variables	No	No	No	Yes	Yes
Created variables	-	-	-	State representative count, In-Degree, Out-degree	State representative count, In-Degree, Out-degree
Number of tables	2	2	2	2	2

Table S5. Key result statements from data-to-paper created "Treatment Policy" papers, together with the used statistical tests (in parenthesis).

Paper	Policy effect on treatment	Policy effect on clinical outcome
Original	"Nearly two thirds of infants in the pre-guideline cohort received endotracheal suctioning with recovery of meconium compared to less than a third of infants in the post-guideline cohort **(*p<0.01*)**." (*Chi-square*)	"Though a higher proportion of the pre-guideline cohort were admitted to the NICU for respiratory issues compared to the post-guideline cohort, **the two groups did not differ significantl**y with regard to morbidity and therapies." (*t-test*)
D1	"[...] **significant changes in interventions** [...] including a decrease in the use of endotracheal suction." *(Chi-square)*	"[...] **no significant differences** in neonatal outcomes were observed between the pre and post guideline groups." *(ANOVA)*
D2	"Following the policy change, [...] **decrease in the application of positive pressure ventilation (PPV)** [...]" *(Linear regression)*	"[...] the policy change **was associated with potential increases in the length of stay** in the neonatal intensive care unit" (*Linear regression*)
D3	"[...] resulted in **significant changes in therapies**, with a decrease in endotracheal suctioning and an increase in the recovery of meconium." *(Chi-square)*	"[...] did **not lead to measurable improvements in neonatal outcomes**, as assessed by APGAR scores, length of Neonatal Intensive Care Unit stay, and SNAPPE-II scores." (*Linear regression*)
D4	"[...] we observed **a significant reduction in the use of endotracheal suctioning** in adherence to the new guidelines." (*Chi-square*)	"[...] **no significant differences** were found in the length of stay or Apgar scores between the pre-and post-guideline periods." (*t-test*)
D5	"[...] revealed a **significant shift in the use of endotracheal suction** following the revised guidelines, [...]" (*Chi-square*)	"[...] **no significant difference in APGAR5 scores** was observed." (*t-test*)
D6	"[...] revealed **significant changes in treatment approaches** following policy revisions." (*Chi-square*)	"[...] **no statistically significant differences in neonatal outcomes** were observed between the pre-and post-guideline implementation groups." (*Mann-Whitney U test*)
D7	"[...] were associated with **a significant decrease in the use of endotracheal suction**, without a notable impact on the usage of positive pressure ventilation." (*Chi-square*)	"[...] **no significant differences in neonatal outcomes**, including Neonatal Intensive Care Unit (NICU) length of stay and Apgar scores at 1 and 5 minutes." (*Mann-Whitney U test*)
D8	"[...] **revealed significant changes in NICU therapies** following the policy change, specifically a marked decrease in endotracheal suction." (*Chi-square*)	"[...] neonatal outcomes **did not exhibit statistically significant differences.**" (*t-test*)
D9	"[...] led to **a significant decrease** in the use of endotracheal suction [...] use of positive pressure ventilation (PPV) did not show a significant change." (*Logistic regression*)	"[...] **no statistically significant differences** in the length of stay in the NICU or the Apgar score at 1 minute between the pre- and post-guideline cohorts." (*Linear regression*)
D10	"[...] **a significant decrease in the use of endotracheal suction**, with a trend towards decreased usage of positive pressure ventilation." (*Chi-square*)	"[...] **no significant differences** in length of stay or APGAR scores between the two groups." (*Linear regression*)

Table S6. Accuracy of analysis and interpretation of the effect of policy change on treatment and clinical outcome, as well as accuracy of overall conclusions, in each of the 10 data-to-paper “Treatment Policy” research papers (Supplementary Manuscripts D1-10).

	Analysis and statistics		**Results interpretation**		**Conclusions**
Paper	**treat-ment**	**clinical outcome**	**treat-ment**	**clinical outcome**	
D1	✓	✓	**?** [d]	✓	✓
D2	✓ [a]	✓	✓ [a]	✖ [e]	✖
D3	✓ [b]	✓	✓	✓	✓
D4	✓	✓	✓	✓	✓
D5	✓	✓	✓	✓	✓
D6	✓	✓	✓	✓	✓
D7	✓	✓	✓	✓	✓
D8	✓	✓	✓	✓	✓
D9	✓	✓ [c]	✓	✓	✓
D10	✓	✓	✓	✓	✓

✓, ✖, **?**: correct / incorrect / borderline

a: By correcting for confounding factors, found a barely significant association of a variable (PPV) with policy change, which was not reported in the original study.

b: Data analysis of treatment change also tested for a change in variables that should not be accounted as treatment (like Breastfeeding).

c: Final analysis is correct, but reviewing the run conversation shows wrong intermediate attempts.

d: While 3 treatment variables were associated with policy change, the results section reported 2 of those as insignificant.

e: Citing p-value of a model’s "intercept" as if it was the p-value for the effect of the policy change on an outcome variable.

Table S7. List of rule-based product checks and auto-corrections.

Product type / Product	**Requested formatting**	**Checks**
Type: Free text	Single text block, enclosed within triple-backticks	- The response must include exactly one triple-backtick block from which the product is extracted. Feedback is provided for no blocks, multiple blocks, or incomplete blocks.
Research goal		No additional checks
Type: Structured text and binary decision	An evaluable Python object	- The response must include a textual representation of a Python object, extracted by its flanking characters ([], or {}). - The extracted textual representation must be an evaluable Python object. - The resulting Python object must be of the requested type (as defined below for each research step).
Literature search I − queries	A Python Dict[str, List[str]]	- The dictionary keys must be: 'dataset', 'questions'. - The dictionary values must be a list of strings each with a maximum of 10 words.
Goal validation	A Python Dict[str, str]	- The dictionary must include a single key: 'choice'. - The dictionary values must be either "OK" or "REVISE".
Hypothesis testing plan	A Python Dict[str, str]	- The dictionary must have at most 3 items. * A dictionary key which starts with the word `hypothesis`, is modified to remove this word.
Literature search II − queries	A Python Dict[str, List[str]]	- The dictionary keys must be: 'background', 'dataset', 'methods', 'results'. - The dictionary values must be a list of strings each with a maximum of 10 words.
Type: LaTex text	Single block of LaTex text enclosed within triple-backticks	- The response must include exactly one triple-backtick block from which the product is extracted. Feedback is extracted for no blocks, multiple blocks, or incomplete blocks. * Any citations with incorrect format are removed. * Any "floating citations", not enclosed with 'cite{}' are fixed to the correct format. * Any unescaped characters are getting escaped. - The extracted LaTex text does not include unwanted LaTex commands (`verb`, `\begin{figure}`, and `cite` which is only allowed in sections with citations). - The citations in the LaTex text that have the correct format are from the extracted list of citations given in the context messages. - The extracted LaTex text includes only the requested section. - The extracted LaTex text is compiled with no errors.
Data exploration − code explanation	```latex \section{Code Explanation} <your code explanation here> ```	No additional checks
Data analysis − code explanation	```latex \section{Code Explanation} <your code explanation here> ```	No additional checks
Title & abstract draft	```latex \title{<your latex-formatted paper title here>} \begin{abstract} <your latex-formatted abstract here> \end{abstract} ```	- The title does not include the ":" symbol. (soft rule - feedback is only provided once) - The abstract is written as a single paragraph ('\n' appears at most once in the abstract string). - The abstract does not include any URL addresses. - The abstract does not include any of the following phrases: 'Acknowledgments', 'Data Availability', 'Author Contributions', 'Competing Interests', 'Additional Information', 'References', 'Supplementary'. - The abstract does not include subsections.

Results	```latex \section{Results} <your latex-formatted writing here> ```	- The section must not include numeric values that were not provided in the conversation context (Methods). - The section does not include any of the following phrases: 'Acknowledgments', 'Data Availability', 'Author Contributions', 'Competing Interests', 'Additional Information', 'References', 'Supplementary', 'In conclusions', 'Future research', 'Future work', 'Future studies', 'Future directions', 'Limitations'. - The section does not include any URL addresses. - The section does not include any unknown result (marked by any of the following options '[unknown', '<unknown', '[insert', '<insert', '[missing', '<missing', '[to be', '<to be', 'xx', 'xxx'), requested to returned as '[unknown]' in the mission prompt). - The section must include specific references for each of the tables. - The section must not include subsections.
Title & abstract	```latex \title{<your latex-formatted paper title here>} \begin{abstract} <your latex-formatted abstract here> \end{abstract} ```	Same as for "Title * abstract draft" (above)
Methods	```latex \section{Methods} <your latex-formatted writing here> \subsection{Data Source} <your latex-formatted writing here> \subsection{Data Preprocessing} <your latex-formatted writing here> \subsection{Data Analysis} <your latex-formatted writing here> ```	- The section must include the requested subsections: 'Data Source', 'Data Preprocessing' and 'Data Analysis'. - The section must not include specific software versions. - The section does not include any URL addresses. - The section must not include any of the following phrases: 'Acknowledgments', 'Data Availability', 'Author Contributions', 'Competing Interests', 'Additional Information', 'References', 'Supplementary'.
Introduction	```latex \section{Introduction} <your latex-formatted writing here> ```	- The section must not include any URL addresses. - The section must not include any of the following phrases: 'Acknowledgments', 'Data Availability', 'Author Contributions', 'Competing Interests', 'Additional Information', 'References', 'Supplementary'. - The section must not include subsections.
Discussion	```latex \section{Discussion} <your latex-formatted writing here> ```	- The section must not include any URL addresses. - The section must not include any of the following phrases: 'Acknowledgments', 'Data Availability', 'Author Contributions', 'Competing Interests', 'Additional Information', 'References', 'Supplementary'. - The section must not include subsections.

Type: Python code	Single block of code enclosed within a triple-backtick	**Static checks:** - The response must include exactly one triple-backtick code block from which the code product is extracted. Feedback is provided for no blocks, multiple blocks, or incomplete blocks. **Runtime checks:** - Run encountered any syntax error, builtin Python runtime error or module runtime error. - Run encountered warnings of specified types (DeprecationWarning, ResourceWarning, and others). - Run does not complete within a user-specified duration. - Run attempts to open files that are not part of the numeric products provided. - Run attempts writing into files that are not part of the requested output files. - Run attempts to import potentially problematic modules, such as ‘os’, ‘sys’, etc. - Run calls any of a list of unallowed functions, including ‘print’, ‘input’, ‘eval’, etc. - Run does not contain ‘__name == '__main__‘
Data exploration - code	Code should include the headers: # Data Size # Summary Statistics # Categorical Variables # Missing Values	- Code must include all the specified headers as comments.
Data analysis - code	Code should include the headers: # IMPORT # LOAD DATA # DATASET PREPARATIONS # DESCRIPTIVE STATISTICS # PREPROCESSING # ANALYSIS # SAVE ADDITIONAL RESULTS	- Code must include all the specified headers as comments.
Table design - code	Code should include the headers: # IMPORT # PREPARATION FOR ALL TABLES # TABLE 1 # TABLE 2 etc (for each table created in the "Data analysis" step.	- Code must include all the specified headers as comments. - Code must contain the imports of: to_latex_with_note, format_p_value, is_str_in_df and split_mapping.
Type: Numerical data		- The code must create the requested output files (see below for each step). - The code must not create any other files.
Data exploration − code output	**"data_exploration.txt"** An output text file. Must contain the following headers: # Data Size # Summary Statistics # Categorical Variables # Missing Values	- The created output file contains all the specified headers. - The created output file is not too large (less than 2500 tokens).

Data analysis − tables	**“table_?.pkl”** At least 2 output files, each containing a single dataframe representing a table for the paper.	- The code must create at least one file "table_?.pkl". - Each “table_?.pkl” file must contain a single dataframe. - Dataframe index must not be a numeric range (to make sure all rows are properly labeled). - Dataframe values must be either strings, numeric, bool, or tuple (to be able to convert them into LaTex in the “Table design” step). PValue objects are also allowed (Methods, Supplementary Table 5). - Dataframes must not include the same non-integer numeric value (to avoid overlap in table content). - Dataframes must not report ‘min’, ’max’, ‘mean’, ‘std’ and quantiles (this improper scientific presentation often occurs as ChatGPT code uses the pd.describe method). - Dataframe must not have any NaN values. - Dataframe must not exceed specified size (max of 10 columns and 20 rows).
Data analysis − other results	**“additional_results.pkl”** An output file containing a Python dict, representing analysis results needed for the paper in addition to the results provided in the tables.	- The code must create a file “additional_results.pkl”. - The file must contain a single object of type Dict[str, Any].
Tables design − tables	**“table_?.tex”** For each "table_?.pkl" file created in the "Data analysis" step, the Table design code must create a "table_?.tex" file, providing a scientifically-formatted LaTex representation of the table.	- A "table_?.tex" must be created for each "table_?.pkl" provided. - Each "table_?.tex" file compiles to Latex without errors. - The width of the compiled table must be within text margins. - The table must not have a column with the same int value (such as often happens in tables that list a “count” column from a pd.describe method). - The table must not have any PValue objects that were not converted to strings (to make sure ChatGPT builds into the code conversion of small p-values into “<1e-6” string; Methods, Supplementary Table 5). - The table has a valid caption. - The table has a valid label in the format “table:<tag>” (to allow citing the tables from the result section). - If the table has a footnote, the footnote must be different from the caption. - Table row and column headers do not contain certain unallowed characters, such as underscores. - The table has a legend with keys matching any label of rows or columns that has characteristics of an abbreviation (such as names that include more than two uppercase characters; and names that contain ‘., ‘:’, ‘_’ punctuation symbols). - The legend does not include keys that are not part of the column or row labels.

Table S8. Package-specific guardrails and p-value tracking.

Package	Guardrails
pandas	● Override dataframe's runtime *KeyError* to not only list the wrongly used key but also provide a list of all available keys. ● Set dataframe's custom string representation with float formatted to two decimal points. ● Override dataframe's *repr* function to avoid omitting columns. ● Raise runtime error when using unallowed dataframe methods, such as *to_html* and *to_json*. ● Override dataframe's *to_latex* function to add chars escaping in table and caption. ● Do not allow changing the original values of a Series of a dataframe that was read from an external file. Instead issue a rule-based message requesting to create a new series with a sensible name to avoid coding mistakes.
statsmodels	● P-value tracking*. ● Do not allow p-values that are either NaN or 1, which indicate erroneous or ill-defined statistical tests. ● Prevent running the *fit* method of a given instance more than once (multiple calls to this function return the same result object, leading to analysis mistakes when not handled properly by ChatGPT code). ● Allow access only to the structured summary of statistical tests (the *summary2* method), and not the textual summary (the *summary* method). This guardrail prevents ChatGPT errors related to parsing of the textual output of *summary*. ● Require performing at least one statistical test. ● Require incorporating a p-value in one of the saved dataframes. ● Require using string formula notation to run statistical models. ● Check for singularity of the covariance matrix to avoid multicollinearity problems.
scipy	● P-value tracking*. ● Do not allow p-values that are either NaN or 1, which indicate erroneous or ill-defined statistical tests. ● Require performing at least one statistical test. ● Require incorporating a p-value in one of the saved dataframes. ● Prevent unpacking or iterating over results objects. Instead, we only allow direct access to the output variables by name to avoid mistakes in accessing the variables in their correct order.
scikit-learn	● P-value tracking*. ● Do not allow p-values that are either NaN or 1, which indicate erroneous or ill-defined statistical tests. ● Limit the grid size to 30 when using *GridSearchCV*. Limit the number of iterations to 30 when using *RandomizedSearchCV*. ● Override *random_state* parameter for *RandomForestRegressor*, *ElasticNet* and *MLPRegressor* classes to automatically set it, if not provided, to assure reproducibility of the analysis results.

- Limit the size of the *MLPRegressor* hidden layers to 2 layers and 50 neurons per layer to avoid too long runtimes.

* P-value tracking:
Rationale: We implement a method to track p-values in order to give rule-based feedback requiring the LLM-created code to correctly format too small p-values as “<1e-6” (or any other user-chosen limit). Without this guardrail, ChatGPT often creates tables with the raw p-values as output by statistical test functions, which could be extremely close to 0 (or even equal to 0 due to floating point rounding).
Implementation: All statistical functions of the package that output p-values are monkeypatched such that p-values are converted from floats to a custom *PValue* class. Objects of this class behave like a float, yet they can be distinguished based on their type. In the “Data analysis” coding step, when the LLM-created code calls such a monkeypatched function, it gets *PValue* objects, which it may subsequently incorporate into the dataframe tables that the code creates. Then, in the “Table design” coding step, the LLM is instructed to use four custom functions that we wrote (see “Performer mission prompt”, “Table design”, Supplementary Table 1): (a) *to_csv_with_note()* which converts a dataframe to LaTex; (b) *format_p_value()* which converts numeric values to strings, such that values lower than 1e-6 is converted to “<1e-6”; (c) *is_str_in_df()* which allows checking if a given string is found in the dataframe’s columns or index; (d) *split_mapping()* which allows creating the table’s legend and a mapping from abbreviations to full names. Importantly, in addition to their normal functionality, as advertised to the LLM, these functions also check and issue a rule-based feedback message either when the code applies *format_p_value* on non-*PValue* arguments, or when the code calls *to_csv_with_note* with a dataframe containing unformatted *PValue* objects. This way we provide rule-based feedback on code that left some p-values unformatted, as well as on code that erroneously apply *format_p_value* on numeric values that are not p-values.

Table S9. Literature search parameters.

Research step	Scope	Number of citations presented	Citation influence threshold	Sorted by
Literature search I	Dataset	12	2	Search rank
	Question	12	2	Search rank
Literature search II	Dataset	12	2	Search rank
	Method	6	10	Search rank
	Background	12	5	Embedding similarity
	Result	12	1	Embedding similarity

Supplementary Data Description.
Below are the human-provided products for each of the 4 datasets A-E, as well as the research goals for datasets D and E.

A. Health Indicators dataset

General description of the dataset

The dataset includes diabetes related factors extracted from the CDC's Behavioral Risk Factor Surveillance System (BRFSS), year 2015. The original BRFSS, from which this dataset is derived, is a health-related telephone survey that is collected annually by the CDC. Each year, the survey collects responses from over 400,000 Americans on health-related risk behaviors, chronic health conditions, and the use of preventative services. These features are either questions directly asked of participants, or calculated variables based on individual participant responses.

Data file description

"diabetes_binary_health_indicators_BRFSS2015.csv"
The csv file is a clean dataset of 253,680 responses (rows) and 22 features (columns).
All rows with missing values were removed from the original dataset; the current file contains no missing values.

The columns in the dataset are:

#1 `Diabetes_binary`: (int, bool) Diabetes (0=no, 1=yes)
#2 `HighBP`: (int, bool) High Blood Pressure (0=no, 1=yes)
#3 `HighChol`: (int, bool) High Cholesterol (0=no, 1=yes)
#4 `CholCheck`: (int, bool) Cholesterol check in 5 years (0=no, 1=yes)
#5 `BMI`: (int, numerical) Body Mass Index
#6 `Smoker`: (int, bool) (0=no, 1=yes)
#7 `Stroke`: (int, bool) Stroke (0=no, 1=yes)
#8 `HeartDiseaseorAttack': (int, bool) coronary heart disease (CHD) or myocardial infarction (MI), (0=no, 1=yes)
#9 `PhysActivity`: (int, bool) Physical Activity in past 30 days (0=no, 1=yes)
#10 `Fruits`: (int, bool) Consume one fruit or more each day (0=no, 1=yes)
#11 `Veggies`: (int, bool) Consume one Vegetable or more each day (0=no, 1=yes)
#12 `HvyAlcoholConsump` (int, bool) Heavy drinkers (0=no, 1=yes)
#13 `AnyHealthcare` (int, bool) Have any kind of health care coverage (0=no, 1=yes)
#14 `NoDocbcCost` (int, bool) Was there a time in the past 12 months when you needed to see a doctor but could not because of cost? (0=no, 1=yes)
#15 `GenHlth` (int, ordinal) self-reported health (1=excellent, 2=very good, 3=good, 4=fair, 5=poor)
#16 `MentHlth` (int, ordinal) How many days during the past 30 days was your mental health not good? (1-30 days)
#17 `PhysHlth` (int, ordinal) Hor how many days during the past 30 days was your physical health not good? (1-30 days)
#18 `DiffWalk` (int, bool) Do you have serious difficulty walking or climbing stairs? (0=no, 1=yes)

#19 `Sex` (int, categorical) Sex (0=female, 1=male)
#20 `Age` (int, ordinal) Age, 13-level age category in intervals of 5 years (1=18-24, 2=25-29, ..., 12=75-79, 13=80 or older)
#21 `Education` (int, ordinal) Education level on a scale of 1-6 (1=Never attended school, 2=Elementary, 3=Some high school, 4=High school, 5=Some college, 6=College)
#22 `Income` (int, ordinal) Income scale on a scale of 1-8 (1=<=10K, 2=<=15K, 3=<=20K, 4=<=25K, 5=<=35K, 6=<=50K, 7=<=75K, 8=>75K)

Here are the first few lines of the file:
```output
Diabetes_binary,HighBP,HighChol,CholCheck,BMI,Smoker,Stroke,HeartDiseaseorAttack,PhysActivity,Fruits,Veggies,HvyAlcoholConsump,AnyHealthcare,NoDocbcCost,GenHlth,MentHlth,PhysHlth,DiffWalk,Sex,Age,Education,Income
0,1,1,1,40,1,0,0,0,0,1,0,1,0,5,18,15,1,0,9,4,3
0,0,0,0,25,1,0,0,1,0,0,0,0,1,3,0,0,0,0,7,6,1
0,1,1,1,28,0,0,0,0,1,0,0,1,1,5,30,30,1,0,9,4,8

```

B. Social Network dataset

General description of the dataset

* Rationale:
The dataset maps US Congress's Twitter interactions into a directed graph with social interactions (edges) among Congress members (nodes). Each member (node) is further characterized by three attributes: Represented State, Political Party, and Chamber, allowing analysis of the adjacency matrix structure, graph metrics and likelihood of interactions across these attributes.

* Data Collection and Network Construction:
Twitter data of members of the 117th US Congress, from both the House and the Senate, were harvested for a 4-month period, February 9 to June 9, 2022 (using the Twitter API). Members with fewer than 100 tweets were excluded from the network.

- `Nodes`. Nodes represent Congress members. Each node is designated an integer node ID (0, 1, 2, ...) which corresponds to a row in `congress_members.csv`, providing the member's Represented State, Political Party, and Chamber.

- `Edges`. A directed edge from node i to node j indicates that member i engaged with member j on Twitter at least once during the 4-month data-collection period. An engagement is defined as a tweet by member i that mentions member j's handle, or as retweets, quote tweets, or replies of i to a tweet by member j.

* Data analysis guidelines:
- Your analysis code should NOT create tables that include names of Congress members, or their Twitter handles.
- Your analysis code should NOT create tables that include names of States, or their two-letter abbreviations. The code may of course do statistical analysis of *properties* related to States, but should not single out specific states.

Data file description

"congress_members.csv"
A csv file of members of the 117th Congress, including their Twitter handles, Represented State, Party, and Chamber.
Data source: `https://pressgallery.house.gov/member-data/members-official-twitter-handles`.
Rows are ordered according to the node ID, starting at 0.

Fields:

`Handle`: Twitter handle (without `@`)
`State`: Categorical; Two-letter state abbreviation; including also: "DC", "PR", "VI", "AS", "GU", "MP".
`Party`: Categorical; Party affiliation ("D", "R", or "I")
`Chamber`: Categorical; The member's chamber ("House", "Senate")

Here are the first few lines of the file:
```output
Handle,State,Party,Chamber
SenatorBaldwin,WI,D,Senate
SenJohnBarrasso,WY,R,Senate
SenatorBennet,CO,D,Senate

```

“congress_edges.dat”
This file provides the interaction network between members of the 115th US Congress on Twitter.
Download and adapted from: `https://snap.stanford.edu/data/congress-twitter`

Each line contains two integers (i, j), indicating a directed edge from node ID i to node ID j, compatible with nx.read_edgelist('congress_edges.dat', create_using=nx.DiGraph()). An i->j edge indicates that Congress member i had at least one tweet engaging with Congress member j during the 4-month collection period.

C. Infection dataset

General description of the dataset

General description
In this prospective, multicentre cohort performed between August 2020 and March 2022, we recruited hospital employees from ten acute/nonacute healthcare networks in Eastern/Northern Switzerland, consisting of 2,595 participants (median follow-up 171 days).
The study comprises infections with the delta and the omicron variant.
We determined immune status in September 2021 based on serology and previous SARS-CoV-2 infections/vaccinations:
Group N (no immunity); Group V (twice vaccinated, uninfected); Group I (infected, unvaccinated); Group H (hybrid: infected and ≥1 vaccination).

Participants were asked to get tested for SARS-CoV-2 in case of compatible symptoms, according to national recommendations.
SARS-CoV-2 was detected by polymerase chain reaction (PCR) or rapid antigen diagnostic (RAD) test, depending on the participating institutions.

"TimeToInfection.csv":
Each healthworker (same `ID`) is represented in several rows corresponding to different time intervals since the start of the study.
For each healthworker and time interval, the file contains information on vaccination status and whether an infection occurred during the interval.

"Symptoms.csv":
Lists only healthworkers infected during the study (those that have an infection event in "TimeToInfection.csv").
Each infected healthworker is represented by a single row in the dataset, indicating the number of symptoms they experienced, and the viral variant causing the infection.

Data file description

The dataset consists of 2 data files:

File 1: "TimeToInfection.csv"
Collected data per worker per time interval.
Each healthworker (same `ID`) is represented in several time intervals (several rows per healthworker).
The file contains 16 columns:

Data in the file "TimeToInfection.csv" is organised in time intervals, from day_interval_start to day_interval_stop.
Missing data is shown as "" for not indicated or not relevant (e.g. which vaccine for the non-vaccinated group).
It is very important to note, that per healthworker (=ID number), several rows (time intervals) can exist,
and the length of the intervals can vary (difference between day_interval_start and day_interval_stop).
This can lead to biased results if not taken into account, e.g. when running a statistical comparison between two columns.

It can also lead to biases when merging the two files, which therefore should be avoided. The file contains 16 columns:
`ID` (int) [*]: Unique Identifier, same in both files
`group` Categorical (str) [*]: Vaccination group: "N" (no immunity), "V" (twice vaccinated, uninfected), "I" (infected, unvaccinated), "H" (hybrid: infected and ≥1 vaccination)
`age` Continuous (float) [*]: age in years ("" for not indicated)
`sex` Categorical (str) [*]: "female", "male" (or "" for not indicated)
`BMI` Categorical (str) [*]: "o30" for >30, "u30" for under 30 ("" for not indicated)
`patient_contact` Categorical (int): Having contact with patients during this interval (1=yes, 0=no, or "" for not indicated)
`using_FFP2_mask` Categorical (int) [*]: Always using protective respiratory masks during work (1=yes, 0=no, or "" for not indicated)
`negative_swab` Categorical (int): Documentation of ≥1 negative test in the previous month (1=yes, 0=no)
`booster` Categorical (int): Receipt of booster vaccination (1=yes, 0=no, or "" for not indicated)
`positive_household` Categorical (int), SARS-CoV-2 infection of a household contact within the same month (1=yes, 0=no)
`months_since_immunisation` Continuous (float), time since the most recent immunization event (infection or vaccination) measured from study start date.
Negative values indicate that infection/vaccination took place after the starting date of the study.
Empty "" for not vaccinated and not infected.
`time_dose1_to_dose_2` Continuous (float), time interval between first and second vaccine dose (months). Empty ("") when not vaccinated twice at the beginning of the time interval.
`vaccinetype` Categorical (str): "Moderna", "Pfizer_BioNTech" or "" for not vaccinated.
`day_interval_start` Continuous (int) interval start day (number of days since start of study)
`day_interval_stop` Continuous (int) interval stop day (number of days since start of study)
`infection_event` Categorical (int): If an infection occured during this time interval (1=yes, 0=no)
Notes / cautions:
- Columns 1-5 and 7, marked with [*], are the same as in the file "Symptoms.csv".
- Empty data "" is used to indicates= missing ot not relevant (e.g. which vaccine for the non-vaccinated group).
- For each healthworker (unique `ID` number), different number of rows (time intervals) can exist.
- For each healthworker, the specific time intervals (`day_interval_start`, `day_interval_stop`) can be different.

File 2: "Symptoms.csv"
Data of infected workers. Each infected worker is represented by a row in the dataset (764 rows).
For each infected worker (line), the dataset contains the following columns:

`ID` (int): Unique Identifier, same in both files

`group` Categorical (str): Vaccination status at the time of infection onset: "N" (no immunity), "V" (twice vaccinated, uninfected), "I" (infected, unvaccinated), "H" (hybrid: infected and ≥1 vaccination)
`age` Continuous (int): age in years
`sex` Categorical (str): "female", "male" (or "" for not indicated)
`BMI` Categorical (str): "o30" for >30, "u30" for under 30
`comorbidity` Categorical (int): any comorbidity pre-existed (1=yes, 0=no)
`using_FFP2_mask` Categorical (int): Always using protective respiratory masks during work (1=yes, 0=no)
`months_until_reinfection` (int): time until next infection in months
`variant` Categorical (str): "delta" or "omicron" (or "" for not indicated)
`booster_over7_days_before` Categorical (int): If a booster was given more than 7 days before the infection event (1=yes, 0=no)
`symptom_number` Continuous (int): Number of symptoms which occurred after the infection

Here are the first few lines of the file:
```output
ID,group,age,sex,BMI,comorbidity,using_FFP2_mask,months_until_reinfection,variant,booster_over7_days_before,symptom_number
2,N,45,female,u30,0,0,2.5,delta,0,11
3,V,58,female,u30,1,0,4.2,omicron,0,6
7,V,32,female,u30,0,1,4.5,omicron,1,5
```

D. Treatment Policy dataset

General description of the dataset

A change in Neonatal Resuscitation Program (NRP) guidelines occurred in 2015:
Pre-2015: Intubation and endotracheal suction was mandatory for all meconium-stained non-vigorous infants
Post-2015: Intubation and endotracheal suction was no longer mandatory; preference for less aggressive interventions based on response to initial resuscitation.

This single-center retrospective study compared Neonatal Intensive Care Unit (NICU) therapies and clinical outcomes of non-vigorous newborns for 117 deliveries pre-guideline implementation versus 106 deliveries post-guideline implementation.

Inclusion criteria included: birth through Meconium-Stained Amniotic Fluid (MSAF) of any consistency, gestational age of 35–42 weeks, and admission to the institution's NICU. Infants were excluded if there were major congenital malformations/anomalies present at birth.

File descriptions

"meconium_nicu_dataset_preprocessed_short.csv"
The dataset contains 44 columns:

`PrePost` (0=Pre, 1=Post) Delivery pre or post the new 2015 policy
`AGE` (int, in years) Maternal age
`GRAVIDA` (int) Gravidity
`PARA` (int) Parity
`HypertensiveDisorders` (1=Yes, 0=No) Gestational hypertensive disorder
`MaternalDiabetes` (1=Yes, 0=No) Gestational diabetes
`ModeDelivery` (Categorical) "VAGINAL" or "CS" (C. Section)
`FetalDistress` (1=Yes, 0=No)
`ProlongedRupture` (1=Yes, 0=No) Prolonged Rupture of Membranes
`Chorioamnionitis` (1=Yes, 0=No)
`Sepsis` (Categorical) Neonatal blood culture ("NO CULTURES", "NEG CULTURES", "POS CULTURES")
`GestationalAge` (float, numerical). in weeks.
`Gender` (Categorical) "M"/ "F"
`BirthWeight` (float, in KG)
`APGAR1` (int, 1-10) 1 minute APGAR score
`APGAR5` (int, 1-10) 5 minute APGAR score
`MeconiumConsistency` (categorical) "THICK" / "THIN"
`PPV` (1=Yes, 0=No) Positive Pressure Ventilation
`EndotrachealSuction` (1=Yes, 0=No) Whether endotracheal suctioning was performed
`MeconiumRecovered` (1=Yes, 0=No)
`CardiopulmonaryResuscitation` (1=Yes, 0=No)
`ReasonAdmission` (categorical) Neonate ICU admission reason. ("OTHER", "RESP" or "CHORIOAMNIONITIS")
`RespiratoryReasonAdmission` (1=Yes, 0=No)
`RespiratoryDistressSyndrome` (1=Yes, 0=No)

`TransientTachypnea` (1=Yes, 0=No)
`MeconiumAspirationSyndrome` (1=Yes, 0=No)
`OxygenTherapy` (1=Yes, 0=No)
`MechanicalVentilation` (1=Yes, 0=No)
`Surfactant` (1=Yes, 0=No) Surfactant inactivation
`Pneumothorax` (1=Yes, 0=No)
`AntibioticsDuration` (float, in days) Neonate treatment duration
`Breastfeeding` (1=Yes, 0=No) Breastfed at NICU
`LengthStay` (float, in days) Length of stay at NICU
`SNAPPE_II_SCORE` (int) 0–20 (mild), 21–40 (moderate), 41- (severe)

Here are the first few lines of the file:
```output
PrePost,AGE,GRAVIDA,PARA,HypertensiveDisorders,MaternalDiabetes,ModeDelivery,
FetalDistress,ProlongedRupture,Chorioamnionitis,Sepsis,GestationalAge,Gender,
BirthWeight,APGAR1,APGAR5,MeconiumConsistency,PPV,EndotrachealSuction,Meconiu
mRecovered,CardiopulmonaryResuscitation,ReasonAdmission,RespiratoryReasonAdmi
ssion,RespiratoryDistressSyndrome,TransientTachypnea,MeconiumAspirationSyndro
me,OxygenTherapy,MechanicalVentilation,Surfactant,Pneumothorax,AntibioticsDur
ation,Breastfeeding,LengthStay,SNAPPE_II_SCORE
1,30,1,1,0,1,CS,1,0,1,NEG
CULTURES,36.6,M,2.65,0,3,THICK,1,1,1,1,RESP,1,0,0,1,0,1,0,0,7,0,9,25
1,32,1,1,0,1,VAGINAL,0,0,1,NEG
CULTURES,39.1,M,4.58,1,4,THIN,1,1,1,0,OTHER,0,0,0,0,0,1,0,0,2,1,14,18
1,34,1,1,0,0,VAGINAL,0,0,0,NEG
CULTURES,38.4,M,3.98,7,9,THICK,0,1,1,0,RESP,1,1,0,0,0,1,0,0,10,0,28,16

```

Human-provided Research goal

Research goal:
Examining the impact of guideline change on neonatal treatment and outcomes.

Hypothesis:
- Change in treatment policy lead to change in treatments.
- The change in treatment policy improved neonatal outcome, measured by duration of stay, apgar scores, etc.

E. Treatment Optimization dataset

General description of the dataset

Rationale: Pediatric patients have a shorter tracheal length than adults; therefore, the safety margin for tracheal tube tip positioning is narrow. Indeed, the tracheal tube tip is misplaced in 35%–50% of pediatric patients and can cause hypoxia, atelectasis, hypercarbia, pneumothorax, and even death.
Therefore, in pediatric patients who require mechanical ventilation, it is crucial to determine the Optimal Tracheal Tube Depth (defined here as `OTTD`, not an official term).

Note: For brevity, we introduce the term `OTTD` to refer to the "optimal tracheal tube depth". This is not an official term that can be found in the literature.

Existing methods: The gold standard to determine OTTD is by chest X-ray, which is time-consuming and requires radiation exposure.
Alternatively, formula-based models on patient features such as age and height are used to determine OTTD, but with limited success.

The provided dataset focus on patients aged 0-7 year old who received post-operative mechanical ventilation after undergoing surgery at Samsung Medical Center between January 2015 and December 2018.
For each of these patients, the dataset provides the OTTD determined by chest X-ray as well as features extracted from patient electronic health records.

File descriptions

"tracheal_tube_insertion.csv"
The csv file is a clean dataset of 969 rows (patients) and 6 columns:
Tube:
#1 `tube` - "tube ID", internal diameter of the tube (mm) [Included only for the formula-based model; Do not use as a machine-learning model feature]
Model features:
#2 `sex` - patient sex (0=female, 1=male)
#3 `age_c` - patient age (years, rounded to half years)
#4 `ht` - patient height (cm)
#5 `wt` - patient weight (kg)
Target:
#6 `tube_depth_G` - Optimal tracheal tube depth as determined by chest X-ray (in cm)

Here are the first few lines of the file:
```output
tube,sex,age_c,ht,wt,tube_depth_G
3.5,0,0,62.8,6.2,9.7
4,1,0,69,9.1,11
3,1,0,52,3.7,8.6

```

Human-provided Research goal

We formulated 6 different research goals that differ in the breadth of requested analysis and in the provision of mathematically explicit instruction.

Ea: 4 Machine-Learning and 3 formula-based models

```
## Research Goal:

To construct and test 4 different machine-learning models and 3 different
formula-based models for the optimal tracheal tube depth (defined here as
`OTTD`, not an official term).

### ML MODELS:
Using the provided features (age, sex, height, weight), your analysis code
should create and evaluate the following 4 machine learning models for
predicting the OTTD:

- Random Forest (RF)
- Elastic Net (EN)
- Support Vector Machine (SVM)
- Neural Network (NN)

Important: It is necessary to hyper-parameter tune each of the models.

### FORMULA-BASED MODELS:
Your analysis code should compute the following 3 formula-based models for
the OTTD:

- Height Formula-based Model:
OTTD = height [cm] / 10 + 5 cm

- Age Formula-based Model:
optimal tube depth is provided for each age group:
0 <= age [years] < 0.5: OTTD = 9 cm
0.5 <= age [years] < 1: OTTD = 10 cm
1 < age [years] < 2: OTTD = 11 cm
2 < age [years]: OTTD = 12 cm + (age [years]) * 0.5 cm / year

- ID Formula-based Model:
OTTD (in cm) = 3 * (tube ID [mm]) * cm/mm

## Hypotheses:

- Each of the 4 machine learning models will have significantly better
predictive power than each of the formula-based models (as measured by their
squared residuals (prediction - target)**2 on the same test set).
```

Eb: 1 Machine-Learning and 1 formula-based model

```
## Research Goal:

To construct and test 1 machine-learning model and 1 formula-based model for
the optimal tracheal tube depth (defined here as `OTTD`, not an official
term).

### ML MODEL:
Using the provided features (age, sex, height, weight), your analysis code
should create and evaluate the following 1 machine learning model for
predicting the OTTD:

- Random Forest (RF)

Important: It is necessary to hyper-parameter tune the model.

### FORMULA-BASED MODEL:
Your analysis code should compute the following 1 formula-based model for the
OTTD:

- Height Formula-based Model:
OTTD = height [cm] / 10 + 5 cm

## Hypothesis:

- The machine-learning model will have a significantly better predictive
power than the formula-based model (as measured by their squared residuals
(prediction - target)**2 on the same test set).
```

Ec: 2 Machine-Learning models

```
## Research Goal:

To construct and test 2 different machine-learning models for the optimal
tracheal tube depth (defined here as `OTTD`, not an official term).

### ML MODELS:
Using the provided features (age, sex, height, weight), your analysis code
should create and evaluate the following 2 machine learning models for
predicting the OTTD:

- Random Forest (RF)
- Elastic Net (EN)

Important: It is necessary to hyper-parameter tune each of the models.

## Hypothesis:
- The two machine-learning models will significantly differ in their
predictive power (as measured by their squared residuals (prediction -
target)**2 on the same test set).
```

Eai, Ebi and Eci:

We provided the same goals as above except with the omission of the explicit mathematical specification of the distance formula. Namely exactly the same goals, just deleting the specification: “`(prediction - target)**2`”.

Supplementary Dataset A. "Health Indicators" dataset.

Supplementary Dataset B. "Social network" dataset.

Supplementary Dataset C. "Infection" dataset.

Supplementary Dataset D. "Treatment Policy" dataset.

Supplementary Dataset E. "Treatment Optimization" dataset.

Supplementary Manuscripts A1-5, B1-5, C1-4, D1-10, Do, Ea1-10, Eb1-10, Ec1-10, Eai1-10, Ebi1-10, Eci1-10, Eo. All manuscripts produced by data-to-paper based on the "Health Indicators" dataset (A1-5), the "Social Network" dataset (B1-5), the "Infection" dataset (C1-4), the "Treatment Policy" dataset (Do (open-goal), D1-10) and the "Treatment Optimization" dataset with its respective goals (Eo (open-goal), Ea1-10, Eb1-10, Ec1-10, Eai1-10, Ebi1-10, Eci1-10). Text has been manually highlighted as follows: green for good practice, e.g. putting findings into context or good representations of results, and correctly used numerical values in the text; yellow for atypical, but not erroneous practice, orange for minor mistakes which do not affect the overall message of the paper, and red for fundamental mistakes, affecting the results or conclusions of the paper. Of note, given that manuscripts C1-4 all had major errors in the data analysis code, which affected the accuracy of following steps, they were only highlighted with red highlighting.

Supplementary Runs A1-5, B1-5, C1-4, D1-10, Do, Ea1-10, Eb1-10, Ec1-10, Eai1-10, Ebi1-10, Eci1-10, Eo. Run files are color-coded terminal output files of the conversations of each of the created papers (html format). They include all data-to-paper research steps, each represented by one or two (in case of a review step) distinct LMM conversations (step starts are marked with 'Starting conversation', name of the conversation, purple). Each message in a conversation starts with a header, which includes (a) message number in the conversation (square brackets); (b) message attribution (Methods; in capital letters: SYSTEM, USER, ASSISTANT, SURROGATE or COMMENTER); (c) casting agent (agent name enclosed in curly brackets used for system testing); (d) conversation name (preceded by `->` symbol). Triple-backtick text within messages is highlighted (brighter for text, or Python code syntax pigmentation). Messages are classified and color-coded as follows: (a) SYSTEM and USER messages in green; (b) SURROGATE messages in turquoise (Methods); (c) COMMENTER messages in blue, which are meant only as comments and are not sent to ChatGPT; (d) ASSISTANT message in bright turquoise, including a header listing all prior messages included in the conversation as sent to the API[32]. Of note, to save space, messages that have already been presented before appear as a single line with "[...]". In addition to the conversation messages, the file also contains the following alerts (in red): (a) API[32] calls including the LLM model and number of tokens; (b) indications of conversation message deletions; (c) API[32] failed responses; (d) model bumping; (e) check of numerical values comparison; and (in blue) (f) Citation retrieved from Semantic Scholar Academic Graph API[30], presenting the information about a successful retrieval of citations or a summary of the information retrieved for each of the citations.

Supplementary Coding Runs. Run files of evaluation of coding capabilities of different LLMs (Fig. S3B). File names indicate the underlying LLM. Same annotation as for Supplementary Runs A-E.

Supplementary Manuscripts Ch and Eh1-3. Human co-piloted example manuscripts. These manuscripts were created in open-goal modality with the “Infection” dataset[27] or in fixed-goal modality with “Treatment Optimization” dataset[29] (Supplementary Data Description C, Ea; Supplementary Dataset C, E; Methods).

Supplementary Video. Short demonstration of data-chaining feature of data-to-paper manuscripts, showcasing the clickable numeric values from Results section to upstream Data analysis code.