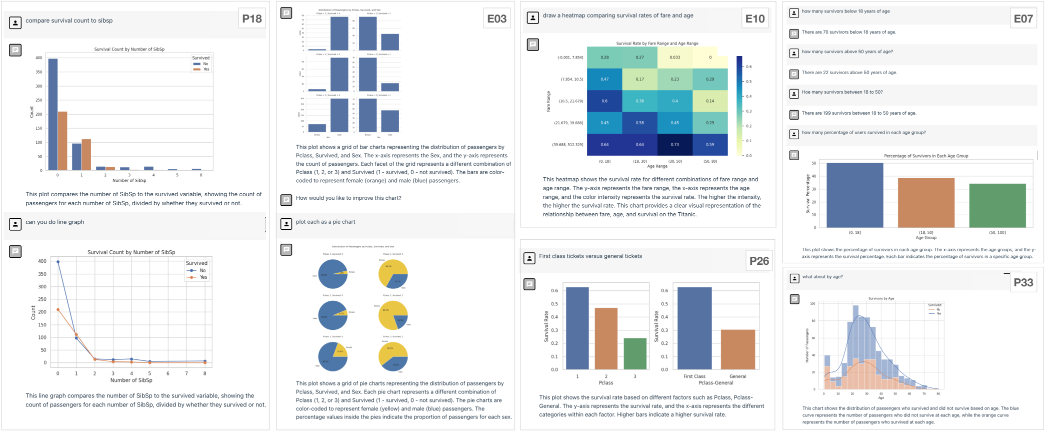
Data has Entered the Chat: How Data Workers Conduct Exploratory Visual Analytic Conversations with GenAI Agents

MATT-HEUN HONG, University of North Carolina, USA
ANAMARIA CRISAN, University of Waterloo, CANADA



P18

compare survival count to sibsp

This plot compares the number of SibSp to the survived variable, showing the count of passengers for each number of SibSp, divided by whether they survived or not.

can you do line graph

This line graph compares the number of SibSp to the survived variable, showing the count of passengers for each number of SibSp, divided by whether they survived or not.

E03

This plot shows a grid of bar charts representing the distribution of passengers by Pclass, Survived, and Sex. The x-axis represents the Sex, and the y-axis represents the count of passengers. Each facet of the grid represents a different combination of Pclass (1, 2, or 3) and Survived (1 - survived, 0 - not survived). The bars are color-coded to represent female (orange) and male (blue) passengers.

How would you like to improve this chart?

plot each as a pie chart

This plot shows a grid of pie charts representing the distribution of passengers by Pclass, Survived, and Sex. Each pie chart represents a different combination of Pclass (1, 2, or 3) and Survived (1 - survived, 0 - not survived). The pie charts are color-coded to represent female (yellow) and male (blue) passengers. The percentage values inside the pies indicate the proportion of passengers for each sex.

E10

draw a heatmap comparing survival rates of fare and age

This heatmap shows the survival rate for different combinations of fare range and age range. The y-axis represents the fare range, the x-axis represents the age range, and the color intensity represents the survival rate. The higher the intensity, the higher the survival rate. This chart provides a clear visual representation of the relationship between fare, age, and survival on the Titanic.

P26

First class tickets versus general tickets

This plot shows the survival rate based on different factors such as Pclass, Pclass-General. The y-axis represents the survival rate, and the x-axis represents the different categories within each factor. Higher bars indicate a higher survival rate.

E07

how many survivors below 18 years of age

There are 70 survivors below 18 years of age.

how many survivors above 50 years of age?

There are 22 survivors above 50 years of age.

Hoe many survivors between 18 to 50?

There are 199 survivors between 18 to 50 years of age.

how many percentage of users survived in each age group?

This plot shows the percentage of survivors in each age group. The x-axis represents the age groups, and the y-axis represents the survival percentage. Each bar indicates the percentage of survivors in a specific age group.

P33

what about by age?

This chart shows the distribution of passengers who survived and did not survive based on age. The blue curve represents the number of passengers who did not survive at each age, while the orange curve represents the number of passengers who survived at each age.

Fig. 1. **Excerpts from participants' exploratory visual analytic conversations, including sequential utterances, derived from our interview (E#) and crowd-sourced (P#) study participants.**

We investigate the potential of leveraging the code-generating capabilities of Large Language Models (LLMs) to support exploratory visual analysis (EVA) via conversational user interfaces (CUIs). We developed a technology probe that was deployed through two studies with a total of 50 data workers to explore the structure and flow of visual analytic conversations during EVA. We analyzed conversations from both studies using thematic analysis and derived a state transition diagram summarizing the conversational flow between four states of participant utterances (**`Analytic Tasks`**, **`Editing Operations`**, **`Elaborations and Enrichments`**, and **`Directive Commands`**) and two states of Generative AI (GenAI) agent responses (visualization, text). We describe the capabilities and limitations of GenAI agents according to each state and transitions between states as three co-occurring loops: analysis elaboration, refinement, and explanation. We discuss our findings as future research trajectories to improve the experiences of data workers using GenAI.
Code & Data: https://osf.io/6wxpa

CCS Concepts: • **Human-centered computing → Empirical studies in visualization**; **Natural language interfaces**.

Additional Key Words and Phrases: Visual Analysis, Conversational Interfaces, Generative AI, User Study

---

Authors' addresses: Matt-Heun Hong, University of North Carolina, USA, mhh@cs.unc.edu; Anamaria Crisan, University of Waterloo, CANADA, ana.crisan@uwaterloo.ca.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
2160-6455/2025/1-ART1 $15.00
https://doi.org/10.1145/3744750

ACM Reference Format:
Matt-Heun Hong and Anamaria Crisan. 2025. Data has Entered the Chat: How Data Workers Conduct Exploratory Visual Analytic Conversations with GenAI Agents. *ACM Trans. Interact. Intell. Syst.* 1, 1, Article 1 (January 2025), 40 pages. https://doi.org/10.1145/3744750

1 INTRODUCTION

Data workers are leveraging automation to support their analysis and visualization of data [9, 74, 79]. However, existing approaches for automation tend to target the model-building phases, with fewer methods that support the preliminary, but essential, exploratory visual analysis (EVA) phases [10]. A unique challenge of EVA is that analysis outcomes need not be concretely defined at the outset and are typically resolved through multiple, often parallel, iterative cycles of information foraging and sensemaking [35, 36, 58, 61, 63, 71]. Such a branching and refining analysis structure can be difficult to automate, and, technical attempts to do so have been met with criticism of brittleness and rigidity[51, 64]. Moreover, EVA can require extensive domain knowledge, which, to date, has been difficult to represent and be used by automated techniques[5, 51]. The code-generating capabilities of Large Language Models (LLMs) [4, 18, 56], and for data analysis and visualization specifically [8, 12, 47], present an potent, but under explored, avenue to automating EVA.

Our research examines the capabilities of LLM-generated code to support EVA via a conversational user interfaces (CUI). Conversational interfaces that support visual analytics can be standalone applications, like Iris [16] or ChartChat [28], or embedded within notebook environments [22]. ***Our goals are to examine the structure of these conversations between a data worker and a Generative AI (GenAI) conversational agent and to compare our observations to pertinent and well-established tasks for visual analytics.*** Prior research has conducted limited explorations of visual analytic conversation for EVA and has reported predominantly on responses to singular utterances. With our goal of studying EVA conversations in mind, we considered the following research questions:

- **RQ1:** How do GenAI conversational agents compare to, and fit, into data workers existing EVA workflows?
- **RQ2:** What is the structure of EVA conversations and how do they evolve?
  - **RQ2A**: What types of utterances do data workers pose?
  - **RQ2B:** How do GenAI conversational agents respond to utterances?

To answer these questions, we conducted two studies with data workers, aided by AI Threads, a technology probe that we developed (Section 3). The first user study conducts semi-structured interviews with ten participants that examine their perception of GenAI agents and how they could impact their current analysis practices (RQ1; Section 5). The second study builds on the results of the first by conducting a broader survey with an additional 40 participants to surface the structure of the EVA conversations (RQ2; Section 6). Through these studies and their findings, our research makes the following contributions: **First, we provide an examination of the utility of GenAI conversational agents for EVA from the lens of data workers.** Our findings show that the primary role of GenAI is to speed up tasks, and that this affordance can help data workers discover new insights even on datasets they are familiar with. **Second, we derived a state transition model that model EVA conversational flow** between four types of participant utterances (`Analytic Tasks`, `Editing Operations`, `Elaborations and Enhancements`,`Directive Commands`) and two types of GenAI agent responses (visualization and text). From this transition model, we describe three common loops (analysis elaboration, refinement, and explanation) that co-occur throughout EVA conversations. **Finally, we provide an annotated dataset of visual analytic conversations of 502 utterance-response pairs.** These can be used to further examine and probe ways that GenAI Agents can support EVA conversations with data workers.

2 RELATED WORK

We cover related research examining data workers' perspectives on automation, techniques for supporting analytic conversational interfaces, and visual natural language interfaces (V-NLIs).

2.1 Examining Perspectives toward Automation in Data Work

Data workers and their practices have been a subject of study across the HCI, CSCW, and Visualization research communities [10, 53, 75]. As the tooling ecosystem for data work evolved, increased scrutiny was directed towards understanding the use of automation in data work [9, 15, 74, 75, 79]. Wang *et. al.* [75] use a technology probe to center the conversation with 20 data workers and understand their attitudes toward automation. Both Crisan *et al.* [9] and Xin *et. al.* [79] interviewed 29 and 16 participants, respectively, to investigate the contemporary usage of tools that automate data work. They also summarize the degree of human intervention desired, or required, across the activities of data work. Drozdal *et al.* [15] interviewed 20 data workers to capture their perspectives on transparency and trust in light of automation. The evolving capabilities of GenAI agents have introduced new avenues for automation in data work. Li *et al.* [40] explored how 18 data workers would collaborate with generative AI agents for data storytelling. Gu *et. al.* also use a technology probe in an interview with 22 data workers to understand how they use AI assistance in data analysis [23]. In a separate study, they also use a Wizard-of-Oz (WoZ) protocol to investigate how participants respond and rate different kinds of suggestions to support analysis planning [22]. Tory and Setlur [71] similarly used a WoZ design to examine how 42 data analysts, a category of data workers [10], use CUIs for preliminary data exploration. Their follow-on research [63], also explores different modalities for interfacing with AI assistants. Both studies [63, 71] were conducted before the popularization of GenAI agents, but anticipate their affordances and role in data work.

Our research deepens the investigation of GenAI-supported EDA by gathering data worker perspectives to bridge ideas about automated analytic assistants and CUIs from past work.

2.2 Conversational Agents for Visual Data Analysis

Conversational user interfaces (CUIs) facilitate an interactive dialogue between the analyst and an automated agent [50]. CUIs have been explored in a variety of domains, from mental health, to education, finance, and hospitality. We focus on the subset of prior research that has examined CUIs specifically for data analysis and visualizations. CUIs for use in data analysis have explored a variety of techniques for mapping utterances to analytic outputs [34, 63]. Iris [16] implements a domain-specific language to map participant utterances to pre-specified analytic functions. Their interfaces support a single-threaded conversation and can produce both text and visual responses. Meaning that the entire conversation history is one continuous sequence of questions and answers. Most notably, Iris can incorporate sequential utterances to compose the agent's responses. Other research explores more question-and-answer scenarios with visualizations. Kim *et. al.* [37] implement a pipeline that leverages natural language templates to facilitate a conversation for explaining visualizations. Their research similarly emphasizes the importance of taking sequential utterances into account to produce visualizations. Recently, Maddigan *et. al* [47] created Chat2Vis and explored the ability of different generative AI agents to generate visualizations in response to user utterances. However, their research did not consider sequential utterances, only singleton question-answer, or evaluation with users. Prior research has also relied on the use of technology probes and human wizards [28, 63, 71] to investigate the expectations and evolution of visual analytic conversations and the interplay between text and visual responses.

Our research further probes the capabilities of CUIs to serve as tools for automating data analysis. We extend prior work by analyzing and describing the branching structure of EVA conversations (Section 3.1), noting the potential of multi-threaded CUIs.

2.3 Generative AI for Visualization & Visual Natural Language Interfaces

Visual Natural Language Interfaces (V-NLIs) are another subset of Natural Language Interfaces, which broadly includes visual analytic CUIs [64]. V-NLIs share many similar methods with CUIs to create appropriate mappings from utterances to dataset attributes, data transformation, and visual encodings, but V-NLIs do not necessarily support conversational turn-taking. Initial approaches, such as DataTone [21], Eviza [62], and FlowSense [80] used parse trees, grammar, rules, or heuristics to resolve these mappings. SNOWY [65] uses templates and parse trees to suggest follow-on utterances based on the present visualization the user generated. These prior works also uncovered the challenges of managing ambiguous utterances, as well as, capturing and modeling the semantics, intent, and pragmatics of these conversations [29]. The Orko system [67] reified these channels with a non-tabular datasets. More recent approaches have tried to learn these mappings directly from the data using machine learning approaches, including Deep Neural Networks. Examples for such systems include, Advisor [42], ncNet [46], Talk2Data [26], and Urania [25]. DNN approaches vary in the specific architecture and labeled training corpora they use to learn mapping functions. While many techniques have been explored, they are still considered too brittle to effectively respond to diverse analytic conversations [47, 64]. =

The use of generative AI for data analysis and visualization is relatively recent but actively explored. Recent tools such as Chat2Vis [47], LIDA [12], and Data Formulator [73] provide preliminary evidence that suggests LLMs outperform other approaches for understanding a natural language utterance and producing an appropriate visualization. Li *et. al.* [39] conducted a recent analysis to further demonstrate that LLMs are capable of generating code for data specifications. However, these systems primarily explore just the generation of a single visualization at a time, not a conversation, and, moreover, have limited user testing. LLMs have also been explored for visualization tasks that are adjacent to data analysis, such as automatic chart summarization [33, 70], chart-question answering [13, 48], and data-driven story generation [27].

Our research explores the capabilities of GenAI agents to support visual analytic conversation. We surface and describe a variety of human utterances and agent responses.

3 AI THREADS: AN EVA CONVERSATIONAL AGENT TECHNOLOGY PROBE

In this section, we describe the design and implementation of AI Threads, a technology probe [31] that we use to conduct our research. Following the approaches of prior work [71, 74], we use AI Threads and a common dataset to conduct two studies, one to understand the role of GenAI (Section 5) and a second to further elaborate on the structure of EVA conversations (Section 6). The primary purpose of AI Threads is to serve as an interface for conducting EVA with a GenAI conversational agent, while also collecting, storing, and replaying user conversations. For transparency and reproducibility, we provide the code [1], additional details of AI Threads's implementation in Appendix B, and examples of its capabilities in Appendix C.

3.1 Supporting a Branching Dialogue Structure

Conversational User Interfaces (CUIs) can take on a variety of designs that support different conversational structures, such as reflection, learning/instructional, and counseling [50, 52]. The current majority of commercial and research CUIs use a single-threaded conversation-centric design, meaning that humans and AI agents take turns and the conversation proceeds sequentially in a single chat window pane (e.g., as in Iris [16]). However, prior research has shown that *exploratory analytic processes, both with and without a conversational agent, have a branching structure.* For example, studies of computational notebooks show that data scientists create 'analysis branches'

---

[1] https://osf.io/6wxpa

to organize and structure their analysis. [35, 36, 58, 61]. A small set of studies [28, 63, 71] have suggested this is also the case for visual analytic conversations with chatbots. Tory and Setlur demonstrated, via a pair of Wizard of Oz (WoZ) studies [63, 71], that it was common for study participants to operate within a common context when refining or elaborating on a particular analysis, but would then shift context entirely when seeking to pose a new question – the result was a branching conversational structure. They showed the need for, but did not develop, alternatives for the common single-threaded designs.

AI Threads was developed to support a branching analysis structure. It does so in a simple way by giving users the option to open a thread and continue to conversation in a side panel. While this is a slight departure from present single-threaded chatbot interfaces, it still follows common multi-threaded conversational interface design patterns (e.g., Slack). In Section 3.2, we discuss the interface elements, including the addition of the `Thread` panel, which was designed to support branching conversations. In Section 3.3, we discuss how we also take advantage of this branching conversation structure to manage GenAI agent context. As multi-threaded designs for CUIs have been long explored [17], and are an intuitive extension to support visual analytic conversations, we do not extensively validate this design choice. However, we did assess it in both studies and did not find any significant issues.

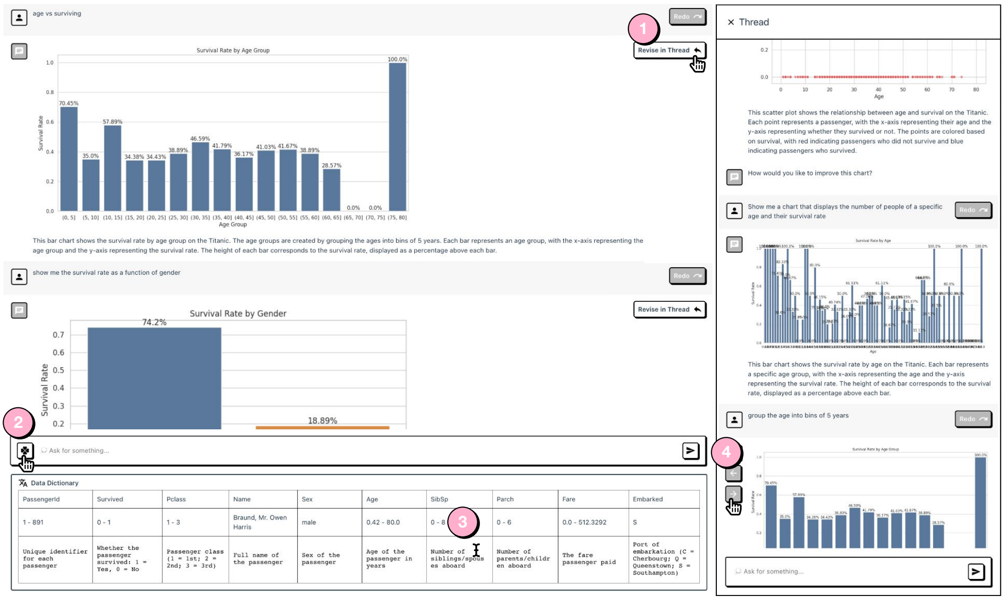
3.2 Technology Probe Design

An overview of AI Threads is shown in Figure 2. In Figure 3 we provide an overview of the pipeline that AI Threads uses to coordinate between the interface, data, and GenAI agent.

Participants upload a dataset to AI Threads (not shown) and are directed to the `Main Chat` area to begin their analysis and converse with the agent. The agent's response can be either text or static visualizations with an accompanying explanatory caption. The `Main Chat` area includes an utterance input panel, a `Data Dictionary` panel, which provides an AI-generated and user-modifiable description of the dataset's attributes, and a panel that retains the diarized chat history.

To converse with the agent, participants can type an utterance or select ✤, which will prompt the model to *"Show me something interesting"*. The agent responds in the `Main Chat`. To further refine a visualization or elaborate on analysis (e.g., create an analysis branch), the user can click on the Revise in Thread ↰ button that triggers the `Thread` panel. For example, they may choose to modify attribute bindings to different channels (by adding or removing them), modify the data transformations (e.g., the bin width in a histogram), add highlights or annotations, and add reference or regression lines among other operations [38, 60]. The conversational agent was programmed to encourage users to employ a `Thread` by including the text "*My responses may not be perfect! Please open a Thread to improve this chart*", with each visualization response. The `Thread` panel also contains a conversation history and utterance input area. Once a user completes their refinements in the `Thread`, the visualization in the `Main Chat` is also updated and becomes part of the main analytic context without the intermediary refinement utterances. The content of a `Thread` persists even after the user closes the panel, allowing them to continue their analysis or share with others how their analysis evolved.

Responding to Errors. If the agent produces an error, or, if the user is not satisfied with the response, they can click the Redo ⤼ button to automatically re-trigger another attempt. Both the current and prior responses are retained and the user can scroll through them using ← or → arrows. Users can opt to redo a response at any point during the conversation, and this does not delete utterances or responses that follow it in the conversation. Importantly, only the most recent version of the response is retained and passed as context to the GenAI agent. Typing 'Undo' will remove the most recent utterance from the chat history so that it is no longer visible to the user either, in addition to removing it from the context passed to the GenAI agent.



age vs surviving

This bar chart shows the survival rate by age group on the Titanic. The age groups are created by grouping the ages into bins of 5 years. Each bar represents an age group, with the x-axis representing the age group and the y-axis representing the survival rate. The height of each bar corresponds to the survival rate, displayed as a percentage above each bar.

show me the survival rate as a function of gender

Data Dictionary

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked
1 - 891	0 - 1	1 - 3	Braund, Mr. Owen Harris	male	0.42 - 80.0	0 - 8	0 - 6	0.0 - 512.3292	S
Unique identifier for each passenger	Whether the passenger survived: 1 = Yes, 0 = No	Passenger class (1 = 1st; 2 = 2nd; 3 = 3rd)	Full name of the passenger	Sex of the passenger	Age of the passenger in years	Number of siblings/spouses aboard	Number of parents/children aboard	The fare passenger paid	Port of embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

This scatter plot shows the relationship between age and survival on the Titanic. Each point represents a passenger, with the x-axis representing their age and the y-axis representing whether they survived or not. The points are colored based on survival, with red indicating passengers who did not survive and blue indicating passengers who survived.

How would you like to improve this chart?

Show me a chart that displays the number of people of a specific age and their survival rate

This bar chart shows the survival rate by age on the Titanic. Each bar represents a specific age group, with the x-axis representing the age and the y-axis representing the survival rate. The height of each bar corresponds to the survival rate, displayed as a percentage above each bar.

group the age into bins of 5 years

Fig. 2. **AI Threads interface.** The `Main Chat` area contains the conversation history diarized by the user's and chatbot's responses. To modify a visualization in collaboration with the agent the user can 1) open the `Thread` panel (right). The `Thread` panel retains a history of modifications that a user can return to and further refine at any time. Closing a `Thread` updates the visualization displayed in the `Main Chat` conversation with the latest version in the `Thread` panel. When in the `Main Chat` area, the user can 2) pose their utterance in the input panel. The `Main Chat` also includes an 3) editable and persistent `Data Dictionary`, which describes the dataset. When agent outputs are re-generated after clicking a redo button, the user can 4) click the arrow keys to revert to previous generative outputs. This figure is a snapshot from P13 analysis (Section 6).

3.3 Implementation

An overview of AI Threads' workflow for responding to user utterances is shown in Figure 3. We implemented AI Threads using Svelte on the frontend and Python REST API on the backend to manage the code execution. AI Threads retains a record of the entire analytic conversation, including whether the user's utterances were part of the `Main Chat` or `Thread`, the code that was generated, and the response that was produced. These records can be reloaded into AI Threads .

3.3.1 *Managing Datasets.*

Passing an entire dataset to a GenAI agent has obvious limitations. Instead, AI Threads passes a data dictionary to the agent, which is a high-level structured description of the dataset including a definition of its attributes. Data dictionaries are commonly used to pass information between human workers [59] and we incorporate this idea into the system prompts to the agent and from AI Threads. Through the `Data Dictionary` pane, users can modify this information. Generating such structured summaries of datasets and using them to augment the analysis of data tables is one of several possible approaches [69]; however, this is also an active area of research, whose outputs will impact the design of CUIs supporting EVA. In theory, using structured summaries, instead of passing some or all of the dataset, could allow an agent to manage datasets of different sizes, and would be limited primarily by the number of dataset attributes and the size of the resulting data dictionary that an LLM can handle as part of its token limits. However, as token limits pertain to both a single prompt and the entire conversational history, the notion of how large a dataset conversational agents can scale to is a dynamic value. Answering this question fully is beyond the scope, but an interesting area of future work.

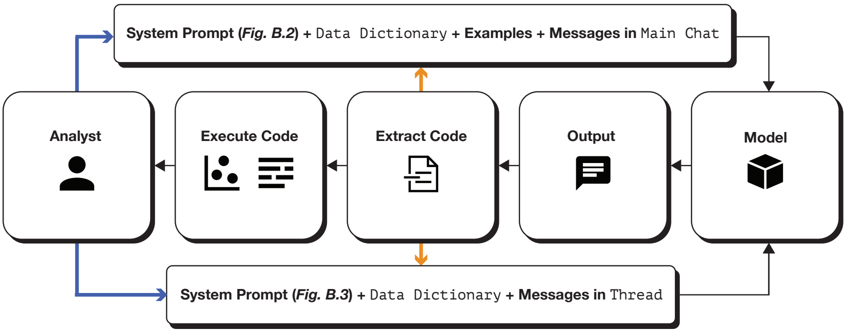

System Prompt (Fig. B.2) + Data Dictionary + Examples + Messages in Main Chat
Analyst
Execute Code
Extract Code
Output
Model
System Prompt (Fig. B.3) + Data Dictionary + Messages in Thread

Fig. 3. **Overview of AI Threads's code generation pipeline.** AI Threads prompts a GenAI agent to generate **executable python code** in response to a **user utterance**. The conversational context passed to the GenAI agent differs depending on whether the utterance is posed in the `Main Chat` (figure top) or `Thread` (bottom).

3.3.2 Managing Conversational Context. Context refers to the history of the conversation. In single-threaded interfaces, this refers to the entire dialogue, limited by the context allowance of the model [76]. While context lengths for LLMs are constantly extending, this does not necessarily improve the model's performance [43]. Borrowing ideas from Tory and Setlur [71], AI Threads, uses information on whether the conversation is occurring in the `Main Chat` or `Thread` panel to prioritize conversational context. When a user opens a thread, AI Threads provides the GenAI agent with a different set of instructions and conversation history compared to the `Main Chat` (see Figure 3, top). Within a `Thread`, an agent is instructed to *revise* an existing piece of code based upon the user's subsequent utterances (Figure 3, bottom). Using conversations and examples from a previously published studies (see Appendix C) we found that this approach improves incremental updates to charts, compared to using a single context thread. As our goal was to generally understand the structure of visual analytic conversations with GenAI, we did not exhaustively consider and compare alternative approaches. More extensive explorations for context management specific to data analysis are outside the scope of our research, but could be an area of future work.

3.3.3 Prompt Engineering. In developing AI Threads, we experimented with different prompts to encourage the GenAI agent to produce code from which we can render and display a data visualization. We did this because ChatGPT, and even Code Interpreter [1], have a preference for producing text outputs over code that generates visualizations. Moreover, the sensitivity of model behavior to multiple prompts is well established. We explored several techniques for generating and evaluating multiple, but related, prompts and comparing their results [3, 81]. Using established analytic tasks from prior works (See Appendix C), we primarily assessed whether the agent consistently produced executable code that, when appropriate, displayed a visualization. Finally, we also compared to Code Interpreter [1] to ensure response consistency. For transparency, the final prompt and instructions we developed are described in Appendix B.

3.4 AI Threads Study Configuration

Prior research exploring new technologies for facilitating automation in data analysis have used a common dataset [71, 75] to conduct studies, without loss of generality. We apply the same strategy and the same `titanic` [20] dataset in both our studies. The primary reason for this choice is that sought to assess how people interact with GenAI agents for EVA – not necessarily how effectively agents perform different analytic tasks and datasets, which should be established via a benchmark. Since, the `titanic` dataset is well-studied, and likely part of an LLM's training data, we hypothesized that agent responses would be higher quality (e.g., more likely to generate executable code, and appropriate responses). We wanted to exploit this over-performance to study the evolution of EVA conversations under more ideal, and less error-prone, scenarios. We could reasonably expect that GenAI performance would degrade on novel datasets. However, recent research also finds that while LLMs do perform well on their training datasets, they are also non-trivially performant on novel datasets [6]. People are also generally knowledgable of the Titanic sinking event, if not the specific dataset, meaning analysts can ask the agent questions that cannot be answered with the dataset (our studies show this does occur). Prior research has not examined a blending between an agent's general knowledge and a user's data. Finally, it remains unclear how commercial LLMs, which we use here, retain data. While a user can upload their own data to AI Threads, concerns about data privacy proved this to be a non-starter. Moreover, at this time, the performance of open-source LLMs for code generation significantly lags behind commercial tools – this may change in the future. We discuss potential limitations of this dataset choice in Section 7.2.

4 OVERALL APPROACH

We explored the structure and evolution of exploratory visual analytic (EVA) conversations involving GenAI by conducting a study with two complementary parts. Both parts rely on AI Threads to scaffold reflection, discussion, and overall impressions. The first part (Section 5) involved semi-structured interviews with data workers to understand the context and perceived role of GenAI in their exploration processes. It also provided a close assessment of AI Threads's design choices. The second part (Section 6) analyzed a broader sample of EVA conversations involving AI Threads, defining dialogue structures by classifying and analyzing participant utterances and agent responses. Collectively, these two studies situate the context of use for GenAI agents within the evolving and branching structure of EVA.

5 STUDY I: SEMI-STRUCTURED INTERVIEWS WITH DATA WORKERS

The main goal of this first study was to examine data workers' perceptions toward the capabilities of GenAI agents for data analysis and visualization (**RQ1**). The perspective of these experts can inform the ways that GenAI conversational agents can be situated in EVA activities.

5.1 Participants & Recruitment

Defining Data Workers: Prior research has shown that data workers have a myriad of roles and skills [10, 53, 75]. Some data workers are capable programmers while others use interactive tools to conduct their analysis. We aimed to recruit participants from across this spectrum to have a broad understanding of how GenAI may be used within EVA phases of data work. Rather than restrict participants to having specific roles (e.g., 'Data Scientist', 'Machine Learning Engineering'), our inclusion criteria focused on participants' routine use of either programming languages (e.g., R, Python, D3) or specialized interactive tools (e.g., Tableau, PowerBI, Excel) to conduct data work. The use of GenAI in present workflows was not a requirement, as there a number of factors (e.g., corporate policy, concerns about privacy, personal preference etc.) that would influence or significantly limit its use.

Recruitment and Demographic Overview: To recruit participants, we distributed an intake survey, which asked participants about their expertise and technical backgrounds, via social media. We received a total of 34 responses; of those, 10 participants met the inclusion criteria for our study. An overview of participants is in Table 1; the participants were predominantly male (n=9) and three said they had used generative AI models to conduct data analysis (primarily via Code Interpreter).

Table 1. Overview of Participants from Study I

Participant	**Role**	**Analysis Tools**	LLM Use
E01	Data Visualization Expert; Business Analyst	Excel and/or Google Sheets; Tableau	No
E02	Engineering Manager	Excel and/or Google Sheets; Tableau	Yes
E03	Visualization Software Engineer	Python and/or R; D3	No
E04	Data Scientist / Engineer	Python and/or R; Tableau	Yes
E05	Technical Writer	Excel and/or Google Sheerts	No
E06	Data Visualization Experts; Business Analyst	Excel and/or Google Sheets; Tableau; SQL; Dataiku;Spotfire; Smartsheet	Yes
E07	Business Analyst	Excel and/or Google Sheets	No
E08	Machine Learning Scientist / Engineer	Python and/or R; D3	No
E09	Engineering Manager	Excel and/or Google Sheets	No
E10	Data Visualization Expert; Business Analyst	Excel and/or Google Sheets; Tableau	Yes

5.2 Session Protocol

5.2.1 Intake. We gathered participants' consent to participate in the study and then asked them to elaborate on their background, role, tool use, and perception of generative AI agents. All participants were aware of GenAI agents and their ability to conduct naturalistic conversations, even if they did not know about the code generation capabilities of these agents or had used them to analyze data.

5.2.2 Dataset Analysis. As part of the intake, participants were also presented with a video showing an overview of AI Threads's capabilities. They were then asked to use AI Threads to analyze the canonical Titanic dataset (see Section 3.4), to identify and visualize factors contributing to overall survival. For practical purposes, participants were given the choice to analyze the dataset before the session or during – this decision was made to be reasonably accommodating to busy professional schedules. When participants chose to conduct the analysis during the session, the study administrators refrained from asking any probing questions or making clarifications. We recorded session logs (Section 3.3) for all analyses regardless of when they were conducted.

5.2.3 Discussion. All participants were invited to join an 'on-camera' session to discuss their analysis. For participants who chose to conduct the analysis in session, this meant that we started the meeting recording at this time. The discussion phase focused on the participants' data analysis, but we also asked them to reflect on their current practices and how they may use GenAI. With all participants, we reviewed their analysis and asked probing questions about their utterances and the agent's responses. We asked them to think-out-loud and posed a set of pre-determined probing questions (see Supplemental Materials) asking participants about their analytic intents, their impressions of the agent's responses, the quality of the visualizations, and the overall usability of AI Threads, including errors that arose during their session. Participants were also asked questions to about the overall usability of AI Threads to establish the impact of our design choices, if any, on the study results. After each session, study administrators conducted a debrief to discuss our observations and emergent themes from across sessions.

5.3 Study Data Collection and Analysis

Each session included one study administrator and a note-taker. The study administrator followed a pre-defined script (see Supplemental Materials), but, where appropriate, they would also follow up with improvised probing questions. In addition to administrator notes, we collected recordings, diarized transcripts, and AI Threads name session logs for each participant. We conducted a thematic analysis [54] of the the session logs. We began our thematic analysis with ~30 utterance examples from the session logs, which we attempted to group into similar themes. As we proceeded through the remaining utterances themes expanded in size or emerged organically. Through iterative refinement, we consolidated and organized the utterances and developed descriptive names for each theme. All study documents and analyses are available in the supplementary materials.

5.4 Results

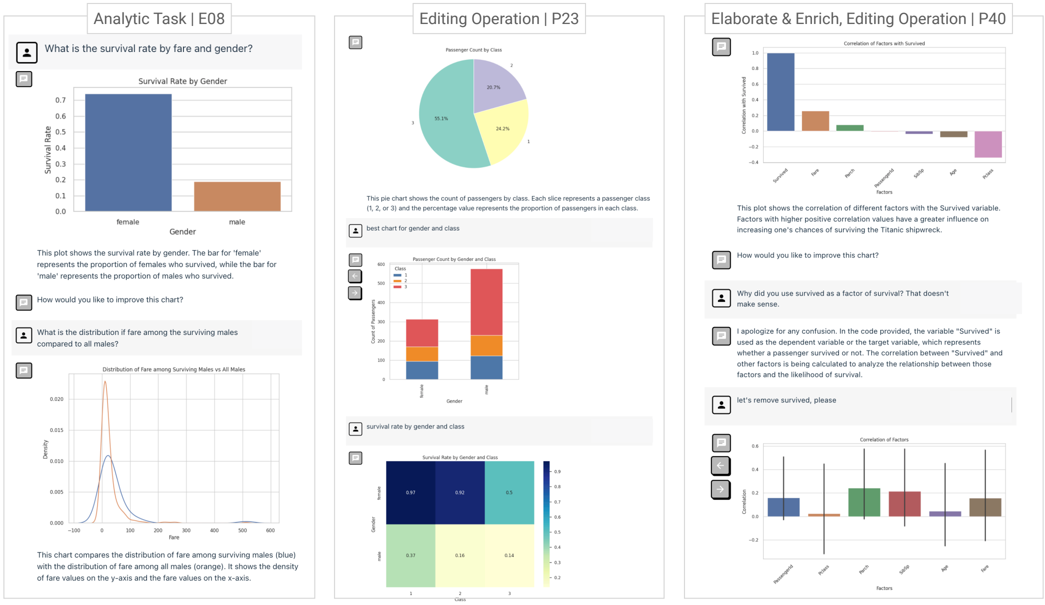
Participants discussed their overall strategies and impressions for interacting with the GenAI agent through AI Threads. They contrasted their experience with tools they presently use, and made further suggestions for improvement.

5.4.1 GenAI Agents can Support Branching EVA Processes. Even though participants all had the historical context for the Titanic event, and in some cases had used the dataset, their first steps were to *"ask simple questions just to [get] orient[ed]'*[E08] and then to *"narrow [their] questions based on responses [they were] getting"*[E06]. Participants found the GenAI agent helped to speed up data orientation and exploration. E06 felt that *"its ability to do basic analysis was really good [and that] you can get [a] quick analysis"*. E07 concurred, stating that *"it would save a lot of time writing the commands [and] I can get a lot of analysis done"*. E05 found that visualizations were especially useful to drive analysis iteration: *"I was trying to get an answer as quickly as I could, which is why a graph is a great way to do that"*. This process of going from broad questions, posed in parallel, to narrowing on more specific questions over time further confirms prior observations that EVA dialogues, even with GenAI agents, continue to follow a branching structure.

While leveraging GenAI to explore these data questions, participants found that the inclusion of threads helped them organize their analysis. E02 found that *"having it [the thread] visually separate from the main thread [...] was really great"* and made following the analysis more *"visually clear"*. E09 felt that *"keeping the context with a certain aspect with[in] the threads is great"*. Reflecting on their use of threads, E08 commented that *"outside of thread was different enough where I wanted to start over [and I] used the thread to modify things [visualizations]"*. This was a very useful feature for E03 who commented that *"the fact that you can do a non-linear refinement of one thing and still keep the context [while you] delve into one particular graph [...] was probably the most useful thing"*. Participants' overall feedback supports multiple, over a single, conversational thread.

5.4.2 GenAI Enables Rapid Iteration. A key feature of how GenAI agents supported EVA was by enabling rapid iteration between exploration, examination, and refinement. Participants did not raise any specific concerns about response latency, and our assessments confirm that the average delay ranged from a few seconds to up to 30 seconds. E08 summarized a common sentiment that GenAI resulted in *"a pretty fast iteration cycle to go from the query to the visual"*. E03 reflects that *"the first time making a dashboard [with D3], I think it took me like 2-3 weeks to get that done. I think having something like this for a first pass would be definitely very useful."* For several participants, this resulted in new insights about the dataset because they could ask questions easily and receive a response quickly. Notably, both E02 and E10, who had analyzed the Titanic dataset, were surprised to discover something new. E02 expressed that they *"learned a lot about the dataset in a fun way, without necessarily doing a lot of work [like] linking and dropping columns and rows."* E10, commented: *"I have played with this data a long time ago, so I knew some of the things, but [...] I didn't realize the survival rate by age was so spread out"* These results align with prior research in automating data science that indicates participants often want the most support at earlier stages of the data workflow where there exists greater ambiguity and need for human and AI interaction [9, 74, 79].

The ease of producing quality visualizations and appropriateness of the results to participant utterances helped rapid iteration. E03 stated that they *"think it's cool in general that you can do this, that you can just take natural language and get a pretty good graph to get what you want"*. As their exploratory analyses evolved, participants felt that keeping track of the visualizations could become difficult. E10 explained that *"if I build, 200 charts in an hour of data exploration [...] I'd love to be able to just see the movie reel of all of those things [I've built]."* Automatically summarizing the analysis is not trivial to resolve and is beyond the scope of this work (see Section 7.1.1).



Analytic Task | E08

What is the survival rate by fare and gender?

This plot shows the survival rate by gender. The bar for 'female' represents the proportion of females who survived, while the bar for 'male' represents the proportion of males who survived.

How would you like to improve this chart?

What is the distribution if fare among the surviving males compared to all males?

This chart compares the distribution of fare among surviving males (blue) with the distribution of fare among all males (orange). It shows the density of fare values on the y-axis and the fare values on the x-axis.

Editing Operation | P23

This pie chart shows the count of passengers by class. Each slice represents a passenger class (1, 2, or 3) and the percentage value represents the proportion of passengers in each class.

best chart for gender and class

survival rate by gender and class

Elaborate & Enrich, Editing Operation | P40

This plot shows the correlation of different factors with the Survived variable. Factors with higher positive correlation values have a greater influence on increasing one's chances of surviving the Titanic shipwreck.

How would you like to improve this chart?

Why did you use survived as a factor of survival? That doesn't make sense.

I apologize for any confusion. In the code provided, the variable "Survived" is used as the dependent variable or the target variable, which represents whether a passenger survived or not. The correlation between "Survived" and other factors is being calculated to analyze the relationship between those factors and the likelihood of survival.

let's remove survived, please

Fig. 4. **Themes and examples of participant utterances from Study I (E#) and Study II (P#).** In Study II (Section 6), we focused on further defining and extending the structure of EVA conversations.

5.4.3 Emergent Conversational Structure. Analyzing the session logs, we observed patterns (themes) of utterances occurring across all participants. One theme was utterances that posed **`analytic tasks`** common to EVA, for example, to aggregate values (*"What is the survival rate by age?"[E10]*), filter values, or compare two values. Another clear theme was utterances that aimed to **`edit visualization`** results, such as adding or removing variables (*"Remove sex from the chart and add Survived[E10]"*), adding annotations (*"add the count of the number of passengers"[E01]*). We also observed utterances with the common theme of **`elaborating on the analysis`**, by trying to retrieve information beyond the dataset (*"Do you know when the Titanic sink[sic]?"[E02]*) or soliciting suggestions(*"what other alternative charts should I consider?"*[E05]). A final theme was utterances in the form of **`directive commands`** (*" ok, use this value to calclunate[sic] the total fare in today's rate"*[E02]). To enable more nuanced conclusions about EVA's conversational structure, Study II (Section 6) involved a larger participant sample analyzed alongside this study.

5.4.4 Limitations and Areas for Improvement. Some participants reported friction in their interactions. For example, E01 was an outlier in finding the chat modality too restricting and wanted more of a multi-modal interaction experience: *"there's a reason why I use a visual tool like Tableau. And that's having a tool that gives me a useful abstraction from manipulating the data to what I'm doing [...]"*. Another participant felt that the agent's response lacked some nuance or did not entirely understand the user's intent. For example, E05 indicated that they wanted the GenAI agent to add visual embellishments without being told to do so: *"let's say my question was what were the top three. It would be nice to highlight them [in the chart]"*. Tied to nuance was the agent's proactivity in responding; some participants felt that the agent should communicate more proactively when encountering errors, or highlight possible analysis paths. E09 felt it would be helpful for the agent to proactively make *"suggestions and recommendations"*. However, the majority of participants did not report or suggest any serious limitations.

6 STUDY II: EXAMINING THE STRUCTURE OF VISUAL ANALYTIC CONVERSATIONS

The goal of this subsequent study was to gain an expanded perspective into the scope of visual analytic conversations with GenAI agents (**RQ2**). This study also uses AI Threads as a technology probe for analyzing and visualizing a dataset. Compared to the prior study, it recruits a larger participant pool and focuses more on the types of participant utterances, the agents' responses, and the cooperative dialogues between humans and AI agents. The results of both studies provide insights into how data workers shape visual analytic conversations (**RQ2A**) in light of agent responses (**RQ2B**) and can be used to improve both the agent's responses in the future.

6.1 Participants & Recruitment

6.1.1 Recruitment. We recruited participants from the Prolific platform[2], aiming to collect a balanced gender sampling. To select for data workers, we pre-screened candidates according to their self-reported technical skills, requiring that they had coding experience and expertise in programming languages or tools commonly used for data work. We also required that they be based in the United States, have an approval rating between 80-100, have completed at least 100 approved prior tasks, and have English language proficiency. All participants completed an intake questionnaire about their backgrounds using the same survey from the prior study. Here, we only report on the results of participants who further met our definition of data worker (Section 5.1). We anticipated approximately 20 minutes to complete the study, for which they were compensated $10.50\hour, which is above the US federal minimum wage. We screened all participants' session logs, removing participants who did not use AI Threads in a manner relevant to the study, for example, by posing a few utterances and then leaving the browser open to extend the time.

6.1.2 Demographics and Skills. We recruited a total of 40 participants (Table 2). Our study had a roughly even breakdown of participants who identified as female (n=19, 47.5%) or male (n=21; 52.5%). A total of 7 (17.5%) participants indicated they were currently students. The average age was 35 ($\sigma_{age} = 14.5$). Participants completed intake questions identical to the previous study (Section 5). From this intake, we know that nearly half (n=17; 42.4%) of participants had a background in either computer science, data science, machine learning, or natural language processing. Moreover, 35 (70%) reported routinely conducting data analysis, primarily with spreadsheet programs (Excel or Google Sheets; n=26; 65%) and/or scripting languages (Python or R; n=9; 22.5%); two participants (P02 and P31) reported previously analyzed the Titanic dataset. Despite the technology's recency, 9 participants (22.5%) reported previously using a GenAI (e.g., Code Interpreter) to carry out data analysis. These self-reported skills overlap with those of data workers in our prior study. Only two participants (P11 and P14) reported background programming experience or other analysis tools; however, this conflicted with their self-reported information on Prolific. After reviewing their sessions, we found they were of good quality and so opted to retain them.

6.2 Session Protocol

We used the same intake survey and dataset analysis as with the previous study (Section 5). Instead of semi-structured interview procedures, participants were asked to respond to a survey that rated the overall user experience, the agent's response, and the quality. Participants were encouraged to conduct a free-form analysis, but we also asked all of them to answer a fixed set of three questions. Participants were also given the option to detail any errors they encountered in the analysis session. We also provided them with the opportunity to provide general feedback. The average session was approximately 24 minutes. The shortest session was 10 minutes and the longest was 65 minutes.

---

[2] https://app.prolific.co/

Table 2. Overview of 40 participants recruited for Study II. In some instances (indicated with *) there was a mismatch between what participant's reported on their profiles and their responses to our intake survey. The majority of participants without programming experience use Excel and/or Google Sheets for analysis.

Factor	Category	Participants	Total
Analysis Tools	Excel and/or Google Sheets	P01, P02, P03, P04, P05, P06, P07, P08, P10, P12, P13, P17, P18, P22, P23, P24, P25, P26, P28, P29, P31, P34, P36, P38, P39, P40	26
	Python and/or R	P08, P13, P15, P17, P18, P19, P26, P28, P30	9
	Stata	P18	1
	None*	P09, P11, P14, P16, P21, P27, P20, P32, P33, P35, P37	11
Programming Experience	No*	P01, P03, P05, P06, P07, P09, P10, P11, P14	9
Prior LLM Use	Yes	P04, P07, P15, P17, P22, P29, P31, P34, P36	9

6.3 Study Data Collection and Analysis

6.3.1 Data Collection. In addition to each participant's session log, we administered a session quality survey that gathered Likert scale and free-form textual responses. Likert scale values included Strongly Disagree, Disagree, Neutral, Agree, and Strongly Agree. As with the prior study, the session logs include a full transcript of the analysis conversation conducted within AI Threads, which constituted the participants' typed utterances as well as the agent responses, including both the code generated for data transformations and visualization specifications and finally a PNG image of the resulting visualization presented to the user (Section 3.3). If participants opted to redo an agent's response, either because there was an error or because they wanted a different visualization, we also retained a record of this action and the agent's prior response(s).

6.3.2 Data Analysis. We summarize all Likert scale questions using simple descriptive statistics; as this was not an experimental study we do not report on the statistical significance of the results. We analyzed the session logs from the previous study (Section 5) concurrently with those of this study (in total n=50 session logs). For agent responses, we tagged whether the response was a visualization, text, or an error. If the response from the agent was a visualization, we indicated what type of visualization it was. We also tagged failures to generate executable code and made comments on other errors we observed. For the participants' utterances, we conduct a thematic analysis [54]. We used the themes from the previous study, while extending them to introduce subcategories. Of the four overarching themes (**Analytic Tasks**, **Editing Operations**, **Elaboration & Enrichments**, and **Directive Commands**), with an additional 22 categories were distributed among them. For analytic tasks, we defined categories by adapting an existing taxonomy of analytic activities [2] and another of analytic intents [66]. For editing operations, we adapt the categories from prior research by Kim *et al.* [38] that described editing transformations between visualizations.

6.4 Data and Session Overview

6.4.1 Overview of Collected Session Logs. Across 50 participants *from both studies* we recorded 502 utterances and the equivalent number of agent responses (1004 utterances & responses in total). The agent responses were primarily visualizations (56% of all agent responses), while the remainder were text responses (38%) or resulted in an error (6%).

6.4.2 Session Quality. The overall session quality will have an impact on participants' responses (e.g., if they felt friction with design choices, or if they encountered errors). Overall, the majority of participants agreed or strongly agreed that interacting with the GenAI agent was easy (77.5%) and enjoyable to use (72.5%). The majority also felt that the agent understood what they were

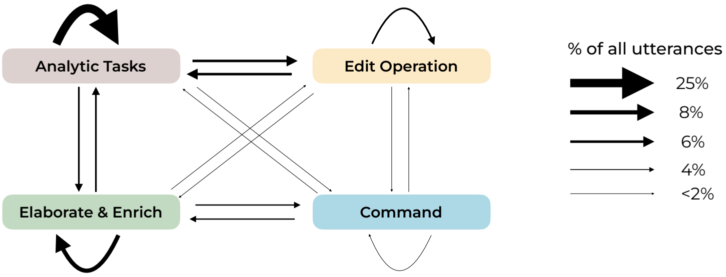


Fig. 5. **Components of EVA conversations and transitions between them.** Participants often posed an analytic task utterance (e.g., to aggregate or filter data) and would commonly follow up by posing another analytic task. However, they also followed up by asking the agent to edit visualizations, elaborate on trends found in visualizations, or direct the agent by giving feedback (e.g., "That's not what I wanted").

asking it (77.0%) and that the visualizations it produced were appropriate for the utterances they inputted (87.5%) and were easy to understand (71.8%). A small minority (3%) expressed negative sentiment towards the `Thread` feature. Overall, these findings suggest that GenAI agents can respond successfully to a variety of user queries and that the introduction of threads into the conversational interface did not have negative influence on the study results.

Summary of Errors: The majority (67.5%) of participants did not report errors when asked or did not find errors that occurred to be onerous. We examined the errors produced in the session logs and found that the majority were failures to generate executable code. This can sometimes be attributed to API request timeouts or issues beyond our control. Other types of errors include returning unrendered code (instead of visualization or text summary; 9 of 502), a failure to produce the right chart when asked for a specific type (2 of 502), or a failure to produce a stylistic change (e.g., make text larger). When an error occurred, participants did use the Redo ↷ option to resolve it. We did not scrutinize all of the generated code for accuracy but did observe instances where the visualizations appeared to show the wrong information (e.g., third class survival rate higher than first class; Figure 6, P35). We also found a small number of instances where the caption or follow-up dialogue accompanying the chart does not its match content. In general, we did not observe significant examples of hallucinations (e.g., incorrect, or evidently fabricated results), but note one interesting exception. We saw a small number of logical reasoning errors that could be attributed to model hallucinations. In Figure 6, P35 asks the agent *"what can you tell me about this chart?"*, the agent's response correctly identifies that first-class passengers were more likely to survive – but the chart erroneously shows the opposite. If the agent had truly considered the chart's data, it should have propagated the error in part in its subsequent responses. We could consider this a special case of hallucination, specific to sequential utterances.

6.5 Results

We break down our analysis of the session logs into three parts. First (Section 6.5.1), we describe the individual components of analytic conversations derived from the thematic analysis (Table 3). Next (Section 6.5.2), we discuss the agent responses. Finally, in Section 6.5.3 we tie everything together into a state transition diagram between participant utterances and agent responses.

6.5.1 The Components of Analytic Conversations. Here, we build on the four primary utterance themes: `analytic tasks`, `editing operations`, `elaborations & enrichments` (E&E), and `directive commands`. Participants used multiple types of utterances throughout their conversations.

Table 3. Themes, Definitions, and Examples for Utterances from Study II.

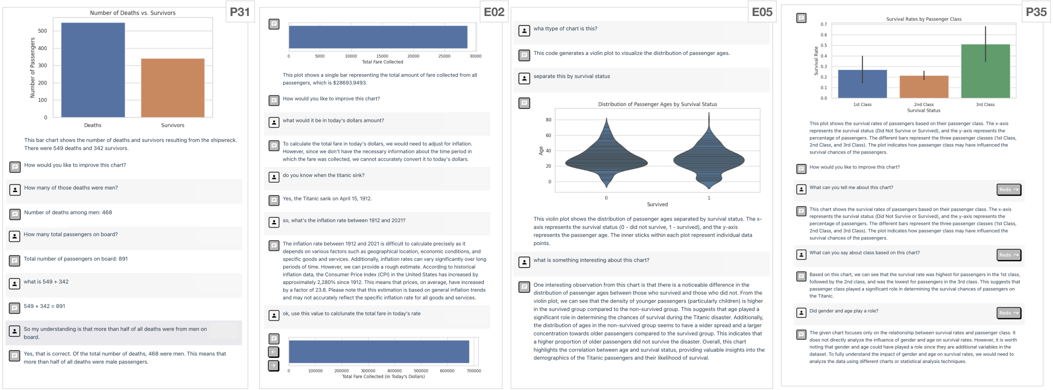
Theme	Definition	Category	Participant Example from Study
`Analytic Task`	Expressions of visual analytic tasks, adapted from Amar *et al.* [2] and Srinivasan*et al.* [65], that concern the dataset	Aggregation	*In total how many passengers survived?*[E07] *Whats the average age of the survivors?*[P22] *What is the survival rate by age?*[E10]
		Comparison	*Compare all the survivors of Cherbough vs. the fare they paid*[P39] *How many children survived versus died?*[P03]
		Correlation	*Does fare relate to survival?*[E10] *generate a chart correlating rate of survival with fare paid*[P30]
		Distribution	*what's the breakdown of passengers by age that survived?*[E02] *What is the distribution if fare among the surviving males compared to all males?*[E08]
		Filter	*How many of the surviving children were girls?*[P11] *How many of those deaths were men?*[P31] *What percentage of people under 18 years old died on the Titanic?*[P25]
		Group	*What percentage of each age group survived?*[P09] *Show me the ages of the people in each class.*[P09]
		Sort	*Order the following from most likely to survive to least likely by gender and class*[P23] *Rank most likely to least likely to survive by Gender and rank*[P27]
		Summary	*What class were the majority of passengers that embarked at Queenstown?*[P23] *What were the oldest age group on the ship?* [P22]
`Editing Operations`	Editing operations to modify the initial chart. Categories are adapted and extended from Kim *et. al.* [38]	Add/Remove Variable	*can you add the number of sibsp as another variable*[E07] *Break those same numbers down further by gender*[P02] *Remove sex from the chart, and add Survived*[E10]
		Annotation	*Alongside the legend, we should add a "Total Survivors: x" count, adding up all the numbers*[P30] *add the count of the number of passengers*[E01]
		Change Encoding	*Can you make the bars side by side instead of stacked?*[E08] *Show it as 2 pie charts (one for Female, one for male)*[E10] *use a different plot for this*[E03]
		Stylistic	*Can you make the numbers more legible? They're blocked by the bars*[P30] *I like to improve this chart by making it bigger and clearer*[P22]
		Transformation	*Can you normalize the Y axis?*[E08] *Can you make it percentages?*[P28]
		Transpose	*Please swap the x and y*[P02] *Rotate the bar so it is horizontal*[E10]
`Elaborate & Enrich`	Contextualizing, utilizing, and expanding on data and analytic insights	Clarification	*What is parch?*[P40] *what type of chart is this?*[E05] *what is being defined within each chart*[P05]
		Explanation	*What variables did you use to determine this?*[P03] *Why did you use survived as a factor of survival? That doesn't make sense.*[P40]
		Information Retrieval	*were there enough lifeboats to save everyone?*[P21] *Do you know when the titanic sink[sic]?*[E02] *Where [sic] there any animals*[P34]
		Probing	*What is something interesting about this chart?*[E05] *What can you say about class based on this chart?*[P35] *which factor figured the most in survival [P38]*
		Suggestions	*What other kinds of charts can you use to best display this data?*[P36] *what other alternative charts should I consider?*[E05]
`Directive Command`	Actions or feedback that chatbot is directed to take	Computation	*ok, use this value to calclunate [sic] the total fare in today's rate*[E02] *what is 549 + 342?*[P31]
		Feedback	*This visualization doesn't makes sense to me. i was more like interested in the count of families*[E09] *It seems a little odd that pclass is fairly low while fare is pretty significant. Seems like higher the pclass, the higher the fare paid, right?*[P40] *that's cool*[P18]
		Compose	*Make a line graph of the correlation between gender and survival*[P20] *I need a pie chart showing passengers by class size.*[P03] *plot pclass, survived, and sex*[E03]

Analytic task utterances were found in 48 (of 50; 96%) participant conversations, E&E utterances in 42 (84%) conversations, editing operations utterances in 32 (64%), and commands in 26 (52%) of conversations. A review of the session logs showed that participants transitioned between different classes of utterances throughout the analysis session (Figure 5); in Section 6.5.3, we elaborate on this observation also taking the GenAI agent's responses into consideration. We now define these different utterance themes, providing examples from participants.

`Analytic tasks (AT)` guided the agent to uncover insights, patterns, or trends within the data. These were the most common type of utterance (47% of total 502 participant utterances) and the most common way that participants began their analyses. We used a prior low-level taxonomy of visual analytic tasks [2] and terminology for visual analytic intents [65] to further annotate categories of this utterance. The majority of utterances expressed a single analytic task, typically an aggregation, and these were often followed up with other types of tasks in subsequent dialogue, either in the `Main Chat` or `Thread`. Utterances containing two analytic tasks most commonly wanted to group or filter the data and then perform an aggregation (e.g.,*"show me a chart of how many people in each age group survived"*[P06]). We observed that the agent was capable of resolving the order of operations correctly to produce the desired result. A small subset of utterances (2%) expressed more than two analytic tasks, for example [P19] asked the agent to show *"what type of class female was most likely to survive in order?"*, which requires filtering the data, grouping it by class, computing survival, and rank ordering the results. There are also interesting examples where the participant appears to ask an analytic question, but is not. For example, P04 asked *"what was the average sex of the people who survived the Titanic? "*, but here they use the word 'average' to mean, 'typical' not the aggregation operation. The agent also had to make decisions about whether to aggregate or show all of the data. For example, P20 asks to *"make a chart of the age of each passenger and whether they died or not"* and it is not clear if the agent should show each passenger, which would produce an unreadable chart or aggregate (it often does the latter, but inconsistently).

`Editing operations (EO)` direct agents to revise a visualization in a specific manner. Editing operations constitute 16% of all utterances. Tory and Setlur [71] describe a conversational state transition model that includes a set of editing operations (select attributes, transform, filter, and encode), however, we observed a greater diversity of operations when reviewing the session logs. Prior research by Kim *et al.* [38] provided more granular terminology for describing transformation and encoding operations, which we apply here. Editing operations were primarily performed within a `Thread` (87%). The most common operation was an explicit request to change the encoding type (36%) (Figure 1, E03; Figure 4, E08) However, some encoding changes also occurred in response to the agent's suggestions for alternative chart types (e.g., *"Can you show me this data in your top choice for graphs?"[E03]* or were vague requests (e.g., *"Please do a different kind of chart to show class.") [P03]*. We did not observe instances where the agent prevented a participant from modifying an encoding or asking follow-up questions for the transition, even when it may have been warranted. For example, some encoding transitions may not be possible without losing information, such as going from a scatter plot to a standard simple bar chart. While participants did not explicitly call out this behavior in their feedback, but, we believe there is room to improve this response.

Another common change was directing the agent to add or remove additional variables (23%) from the visualizations. One side effect of adding/removing variables was to change the encoding type as well, even if this was not expressed in the utterance (Figure 4, P05). Another common category of edits was stylistic changes (21%) (e.g., changing the font size, adding annotations), which were some of the largest sources of errors in the agent's responses. Whether the participants wanted larger font sizes, to move the legend, or to improve the positioning of annotations, the agents rather consistently failed to respond appropriately to these requests. We observed two



P31

This bar chart shows the number of deaths and survivors resulting from the shipwreck. There were 549 deaths and 342 survivors.

How would you like to improve this chart?

How many of those deaths were men?

Number of deaths among men: 468

How many total passengers on board?

Total number of passengers on board: 891

what is 549 + 342

549 + 342 = 891

So my understanding is that more than half of all deaths were from men on board.

Yes, that is correct. Of the total number of deaths, 468 were men. This means that more than half of all deaths were male passengers.

E02

This plot shows a single bar representing the total amount of fare collected from all passengers, which is $28693.9493.

How would you like to improve this chart?

what would it be in today's dollars amount?

To calculate the total fare in today's dollars, we would need to adjust for inflation. However, since we don't have the necessary information about the time period in which the fare was collected, we cannot accurately convert it to today's dollars.

do you know when the titanic sink?

Yes, the Titanic sank on April 15, 1912.

so, what's the inflation rate between 1912 and 2021?

The inflation rate between 1912 and 2021 is difficult to calculate precisely as it depends on various factors such as geographical location, economic conditions, and specific goods and services. Additionally, inflation rates can vary significantly over long periods of time. However, we can provide a rough estimate. According to historical inflation data, the Consumer Price Index (CPI) in the United States has increased by approximately 2,280% since 1912. This means that prices, on average, have increased by a factor of 23.8. Please note that this estimation is based on general inflation trends and may not accurately reflect the specific inflation rate for all goods and services.

ok, use this value to calclunate the total fare in today's rate

E05

wha ttype of chart is this?

This code generates a violin plot to visualize the distribution of passenger ages.

separate this by survival status

This violin plot shows the distribution of passenger ages separated by survival status. The x-axis represents the survival status (0 - did not survive, 1 - survived), and the y-axis represents the passenger age. The inner sticks within each plot represent individual data points.

what is something interesting about this chart?

One interesting observation from this chart is that there is a noticeable difference in the distribution of passenger ages between those who survived and those who did not. From the violin plot, we can see that the density of younger passengers (particularly children) is higher in the survived group compared to the non-survived group. This suggests that age played a significant role in determining the chances of survival during the Titanic disaster. Additionally, the distribution of ages in the non-survived group seems to have a wider spread and a larger concentration towards older passengers compared to the survived group. This indicates that a higher proportion of older passengers did not survive the disaster. Overall, this chart highlights the correlation between age and survival status, providing valuable insights into the demographics of the Titanic passengers and their likelihood of survival.

P35

This plot shows the survival rates of passengers based on their passenger class. The x-axis represents the survival status (Did Not Survive or Survived), and the y-axis represents the percentage of passengers. The different bars represent the three passenger classes (1st Class, 2nd Class, and 3rd Class). The plot indicates how passenger class may have influenced the survival chances of the passengers.

How would you like to improve this chart?

What can you tell me about this chart?

This chart shows the survival rates of passengers based on their passenger class. The x-axis represents the survival status (Did Not Survive or Survived), and the y-axis represents the percentage of passengers. The different bars represent the three passenger classes (1st Class, 2nd Class, and 3rd Class). The plot indicates how passenger class may have influenced the survival chances of the passengers.

What can you say about class based on this chart?

Based on this chart, we can see that the survival rate was highest for passengers in the 1st class, followed by the 2nd class, and was the lowest for passengers in the 3rd class. This suggests that passenger class played a significant role in determining the survival chances of passengers on the Titanic.

Did gender and age play a role?

The given chart focuses only on the relationship between survival rates and passenger class. It does not directly analyze the influence of gender and age on survival rates. However, it is worth noting that gender and age could have played a role since they are additional variables in the dataset. To fully understand the impact of gender and age on survival rates, we would need to analyze the data using different charts or statistical analysis techniques.

Fig. 6. Examples of co-operative dialogues between agents and participants

predominant patterns of responses. The first seemed to be a failure to understand and apply the command. Reviewing the logs, it is also possible to observe some frustration as participants repeatedly request these stylistic changes and then eventually move on. A second source of error was anchoring issues [41], where the visualization changed too much in response to an edit. Future work concerning EVA conversations may be better to achieve these stylistic changes, and likely other editing operations, via alternative interaction techniques instead of by typing utterances. For example, by allowing the user to interactively update a visualization specification rather than continue to rely on textual inputs. The remaining categories (annotation, transformation, and transpose) collectively represent 20% of all edit operations.

`Elaborate and Enrich (E&E)` **utterances broadly capture participants' attempts to contextualize, utilize, or expand upon insights that surfaced from the data or analysis.** E&E utterances constitute 24% of all utterances. The majority (38%) of E&E utterances took the form of probing questions. Participants most commonly use probing questions to ask whether there was anything interesting in the chart. Participant's next most common category (15%) of E&E utterances was followed by asking the agent to ***retrieve information that was not present in the dataset***. For example, participants inquired about *"how many lifeboats did the Titanic have?"*[P21], or if *"you [the agent] know when the titanic sink?[sic]"[E02].* A total 21% of E&E utterances were aimed at clarifying or explaining elements of the data or agent's response. For example, E05 (Figure 6) asks *"what type of chart is this?"*. Explanations went further as well, asking, for example, *"Could you [the agent] explain this chart to me more?"*[P36] or *"what is being defined within each chart?"*[P05].

Some of these E&E questions could be answered with ***a combination of information contained within data and also from other sources***. For example, E02 asked the agent to compute the total fare collected into modern currency, which it does using computations from the dataset and by extracting information from a follow-up response (Figure 6, E02). Another example of this behavior is when a participant asks for chart or analysis suggestions, which requires combining the context of the conversation (present chart type and attributes visualized) with general knowledge that is not contained in the dataset [3] [P09] asks *"what decks were the various classes located on?"* to which the agent responds with a heatmap visualization. There is no column in the dataset corresponding to deck information. However, the first letter of the cabin number (a column in the dataset) does

---

[3] Note that AI Threads does not use web search capabilities

correspond to the deck[4]; the relationship between the cabin and deck information is not present the column description. Upon review, we found that the agent generated code to extract the first letter from the cabin column, it then created a new 'deck' column and finally summarized the data by class to produce the visualization. Through AI Threads, a GenAI agent is instructed to prepend *"I can't answer this question with the data"* whenever it responds to an utterance with information outside of the dataset. However, it does not always do so.

`Directive Commands (DC)` are utterances that articulate an expected response, or attempt to guide the agent. When participants wanted to exert greater control over the agent's responses they posed command utterances (12% of the 502 total). What differentiates these utterances from others is the level of detail in which the participants pose their utterances. Compose utterances often specified specific chart types, used actual attribute names e.g., pclass instead of 'passenger class' or 'fare class'), or directed the agent to change its response type (e.g., "show this a visualization instead"). In a very small number of instances, a participant asked for specific computations (Figure 6, P31 and E02). Another way that participants tried to steer agents' responses was through feedback. For example, P30 tells the agent that *"That graph [the agent produced] isn't readable, it's completely filled with grey lines and the legend goes over the title"*. In response to a failed attempt to modify a bar chart encoding E08 comments that *"the bars are still stacked though"*. However, there were also examples when participants simply thought something was interesting and chose to pay the model a compliment (e.g., *"that's cool"[P18]*, *"Very useful"[E10]*). It was not clear that the agent used either kind of feedback effectively to improve their subsequent responses.

6.5.2 Quality and Diversity of Agent Responses. Prior studies on visual analytic conversations were limited in their capacity to interrogate the interplay between visual and textual responses. The WoZ study by Tory & Setlur [71] only produced visualizations for responses; their follow-on research exploring multiple modalities [63] used pre-defined template responses. The Iris system [16] enabled both automatic textual and visual responses, but offers a limited perspective on the range and diversity of responses. We find that not only can GenAI agents appropriately switch between text and visualization responses, but can even build on the context of each. In Figure 1 we show how E07 defines age categories through several sequential utterances that are then incorporated into a visualization. More examples are shown in Figure 6. ***Overall, we find that GenAI agents are capable of switching relatively seamlessly between producing text and visualization responses through a series of long sequential utterances.***

When the agent produced a visualization (56% of all responses), it produced 12 unique chart types and in 11% of all instances produced multi-panel charts. The most common chart type was a bar chart (77%), followed by a pie chart (13%), and a histogram (11%). Interestingly, it produced variations on different families of charts in response to variances in user utterances. For example, it produced a simple bar chart (77%), but also stacked (6%) and divergent (1%) variants. For representing distributions, it produced histograms (11%), violin plots (3%), boxplots (2%), and density plots (0.5%). To show bivariate relationships it produced scatter plots (3.5%), line plots (2%), and even connected scatter plots. To show more complex multivariate relationships it also produced heatmaps (2.5%). Visualizations were produced both in response to specific requests for charts (e.g., *"show this as heatmap"*) but were more commonly automatically selected by the agent (e.g., *"show me the distribution of age"*, produces a histogram). Overall, participants thought the visualizations produced by the agent were appropriate for their questions. A total of 87% of participants either agreed (65%) or strongly agreed (22%) that the visualizations produced were appropriate for the questions they asked. The majority (80%) also found that the text captions describing the visualization, which

---

[4] https://www.encyclopedia-titanica.org/cabins.html and https://www.kaggle.com/code/ccastleberry/titanic-cabin-features

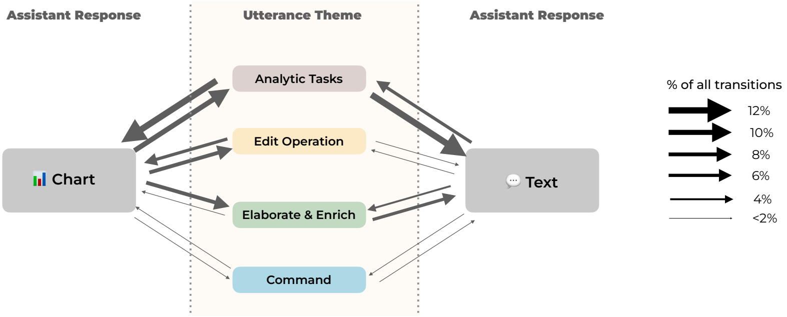


Fig. 7. A state transition diagram summarizing the sequences of participant and agent dialogue.

were also automatically produced, were useful. The use of a `Thread` panel to further elaborate on analysis and visualizations was also perceived to be helpful by participants. P13 was enthusiastic about their experience: *"I really enjoyed the thread feature to edit the graphs and make them a bit easier with more information on what was wanted"*. As we previously discussed (Section **??**), our choice of dataset likely does influence the quality of visualization. As such, this observation serves as more of a check on the session quality (e.g., more than one type of chart; generally valid results).

6.5.3 The Evolution of Visual Analytic Conversations. Having defined the types of utterances and attendant agent responses, we now consider how it all comes together in analytic conversations.

In Figure 7 we present a state transition diagram that shows how conversations evolved between different types of utterances and agents responses. In Figure 5, we present only the transition between the types of utterances. Taking these two results together, **`Analytic tasks`** were not only the most common type of utterance, but, was also how the majority of conversations began. In response to an analytic task, it was nearly equally likely for the agent to respond with a visualization (📊) or text (💬). However, the most common loop in conversations was **`AT`** →📊→**`AT`**. These loops could play out as asking a series of new questions (e.g., switch context) or, more commonly, further elaborating on the analysis (e.g., maintain context). For example, a participant might ask to first aggregate survival by age, view the result, and then go further to filter by a specific gender. We refer to this as the 'analysis elaboration loop', where participants begin with typical broad questions and narrow down over time, which we also observed in the prior study (Section 5). Interestingly, text responses appeared to lead to less follow-up relative to charts, highlighting that visualization responses provide more engagement.

Another common loop is **`AT`**→📊→**`EO`**. We refer to this as the '*refinement loop*', where the result (typically a visualization) is iteratively improved on through text. While the '*refinement loop*' is common, we also know from the prior study (Section **??**) that it is suboptimal – participants do not want to refine a visualization through sequential utterances, they would rather do so interactively through other modalities. In Tory and Setlur [71], they also find evidence for the '*refinement loop*', but with a deeper exploration into its possibilities – we conducted this deeper exploration in our work through our application of Kim *et. al.'s* [38] taxonomy for graph editing. Another common loop was the '*explanatory loop*', transitioning roughly as follows: **`AT`**→📊→**`EE`**→💬. This loop shows how participants engage with information in the visualization, expecting the GenAI agent to be adept at answering questions about a visualization to discuss trends and findings. Importantly, these three loops are not distinct but feed into each other as participants pursued multiple parallel lines of analytic inquiries. ***This observation uniquely highlights nuanced and integrated aspects of EVA conversations that prior research has not covered.*** Prior work often understood one narrow facet of analytic conversations, whereas, we find evidence that, under relatively ideal circumstances, EVA conversations have richer structures. It also reifies that AI Threads can support the branching structure of visual analytic conversations that others have also observed.

The conversation transitioned not only between utterance types, but, parts of the interface. Participants used the multi-threaded affordances of AI Threads to move between the `Main Chat` and one or more `Threads`; a total of 62% of utterances appeared in the `Main Chat` and the remainder were in a `Thread`. There was a strong relationship between the type of utterance and where it occurred in the interface. We observed that 90% of **`edit operation`** utterances occurred in the `Threads`, whereas only 19% of **`analytic task`** utterances do. A total 62% of **`E&E`** occurred in the `Main Chat` and the rest in the `Threads`. The distribution of where utterances were posed shows there was a logic to the structure of these conversations that AI Threads was able to facilitate. As conversations grew longer, we also observed a handful of instances where participants wanted to branch analysis from earlier points in the conversation. For example, in a thread, P36 asks the agent if *"[we can] go back to the old style of chart [from earlier in the dialogue?]"*. The agent failed to fulfill this request. This highlights the temporal relationships of utterances that can be difficult for the agent to resolve. Notably lacking are attempts from the agents to ask the participant for clarification. The dialogue was generally one-way, with agents responding to participants' utterances.

6.6 Summary of Findings

Our analysis of visual analytic conversations with a broader participant pool reveals a rich and intentional dialogue between data workers and GenAI agents. Our findings show how EVA conversations evolve between multiple states of participant utterances and agent responses. We show that under idealized conditions, GenAI agents can appropriately transition between states in a way that is useful to data workers. We also resolve three co-occurring loops in EVA conversations (*'analysis elaboration'*, *'refinement'* and *'explanation'*) and discuss their interplay in EVA conversations.

7 DISCUSSION

Research examining the use of generative AI agents for facilitating data analysis and visualization focuses largely on its technical capabilities and less so on what people actually do. In our research, we leveraged a human-centered approach to examine how people conduct exploratory visual analyses (EVA) with GenAI agents through a multi-threaded Conversational User Interface(CUI). Our research builds on the established understanding that EVA has a branched structure [35, 36, 58, 61, 71], where data workers ask multiple questions in parallel and shift between branches to dig deeper [63, 71]. Using the AI Threads technology probe as a scaffold, we explored to what extent and in what ways these prior observations hold up with GenAI agents. Through a series of two studies examining the context of use and evolution of EVA conversations, we find that humans and GenAI agents are rapidly switching contexts between different data questions and conversational intents. To describe this transitional flow, we defined four types of utterances (**`Analytic Tasks`**, **`Editing Operations`**, **`Elaborations and Enrichments`**, and **`Directive operations`**) and defined three predominately conversational loops (Analytic Elaboration, Refinement, and Explanation). Prior research has only identified some aspects of these EVA loops (notably, refinement [71]), but has not previously shown how these loops co-occur in the stream of conversation. ***As the momentum grows for incorporating GenAI more deeply into analytic workflows, our findings provide a grounded assessment of how data workers and GenAI agents collaborate.***

7.1 Applications of our Findings and Areas of Future Research

7.1.1 Designing Conversational Interfaces for Data Work. Visual analytic conversations with GenAI straddle a unique position between code generation, naturalistic dialogues, and data visualizations. The properties of these conversations can require design considerations unique to data work.

The ways that data workers interface with GenAI agents need to continue evolving. Adding GenAI agents to present tools for data analysis will potentially fail without consideration of the structure of visual analytic conversations. Data workers, regardless of technical capabilities, may wish to switch between posing text utterances and interactively modifying visualizations. Enabling multi-modal refinement actions could help further organize and streamline analytic conversations by reducing the need for sequential text utterances. Finally, whether chat, a notebook, or some other interface (or a combination thereof), the persistent problems of analytic transparency and provenance grow with more complex interactions. Adding appropriate scaffolds to retain and expose the analysis' provenance becomes more complex in light of expanding multi-modal interactions. The issues of tracking visual and analytic data provenance have been explored in other interactive systems and could be incorporated into conversational GenAI interfaces for data work [57]. Moreover, provenance information that captures and summarizes the branching structure of EVA can help conversational agents maintain and switch context appropriately; a potentially impactful area of future work is to further explore the relationship between provenance data and an agent's context.

The response of agents should be further improved to facilitate two-way dialogues. GenAI agents show limited capacity to seek further information from the user, either to clarify their intents or suggest possible actions. In our studies, we found that participants wanted more suggestions from the agent. Agents could offer options in response to analytic task utterances, as the SNOWY does [65] for singular charts, or for alternative analysis paths [22, 23]. We also identified areas where agents could proactively alert participants to destructive changes (e.g., dropping attributes when changing chart types) or help the user form a better mental model when encountering errors. Finally, there are opportunities to make use of different kinds of agent responses, visualization, and charts, to offer more nuances and qualitative insights [68]. Given the diverse components of visual analytic conversations, it may be necessary to use multiple agents, rather than just one, that are specialized to proactively respond to different types of utterances. Our dataset of 502 annotated utterance-response pairs can help researchers explore making agents aware, and modify their behavior to, different types of utterances.

Opportunities for multi-agent interactions. Our current research, and that of many others published at this time, remains focused on a single data worker interacting with a single GenAI conversational agent. Exploring multi-human interactions in the context of AI-assisted analysis is an obvious next step, but, an overlooked and equally relevant research trajectory is toward multi-agent interaction as well [19, 24]. An extension of our work can be to develop multi-agent systems, where each agent is specialized to respond to different categories of utterances. For example, an AI agent specifically trained in editing operations might do a better job making sequential refinements than a single general-purpose agent. Overall, this can improve the quality of agent responses because agents would be specialized to perform tasks.

7.1.2 Improving Agent's Abilities to Reason with Data. **The types of errors that the agent makes did not appear random in our study**. We found more errors with certain types of utterances, for example, stylistic changes of edit operations or explanatory E&E utterances. An application of our findings is to assess the performance of present and future GenAI conversation agents and to surface whether there persists a relationship between utterance type, the nature of agents' responses, and the propensity of errors. There exist quantitative benchmarks for chart question and answering [48] as well as captioning [70] but are limited to individual visualizations and do not capture sequential utterances found in visual analytic conversations. Moreover, defining quantitative metrics for generated code, targeting data analysis or otherwise, is difficult as subjectivity in human assessment is an important factor [11]. Our findings can help delineate the requirements of future benchmarks for visual analytic conversational settings.

GenAI agents do not know how to reason about the data underlying a visualization. We observed erroneous hallucinated responses to questions about the chart (e.g., Figure 6, P35) - but this was surprisingly uncommon. The low prevalence does not mean that the agent can reason with the data, instead, the agent likely leverages the dataset semantics with additional knowledge of the Titanic event to produce a convincingly accurate response. Concerns about these types of hallucinations have been observed in prior research and have proposed an approach to address them [48, 49]. However, in our investigation, the agent generates the code needed to produce the visualization, whereas prior research extracts the data from an image of a data visualization. This additional information encapsulated in the code, and not just the dataset, should be leveraged in future research to improve the capabilities of GenAI agents to reason with data.

7.1.3 Debugging with GenAI Agents. By design, our study is primed to limit the errors by using a dataset that an LLM was likely trained on. However, even under such performant settings, there are subtle issues that can make it difficult to verify the correctness of the agent's response. We specifically want to highlight LLMs recruiting and applying additional knowledge to compute its results. In Figure 6, we provide an example where the agent translates the total fare collected into 2021 currency – it provides a convincing and plausible response, but that is very difficult to determine if it is correct. We found other instances in our participants' logs of similar behavior. It is not clear if this behavior can or should be effectively limited, and if so, how it affects the model's overall behavior. We believe this is a fruitful area to explore in future work, both to achieve a fundamental understanding of how LLMs operate for data analyses and also to provide an effective audit trail for its responses.

7.2 Generalizability and Transferability of Findings

A limitation of our research is that we used a well-studied dataset to conduct our studies. While prior research has taken a similar approach [71, 75] without loss of generality, we address this concern with respect to our findings; in Section 3.4 we have already provided our rationale for this choice and do not repeat it here, instead focusing on its implications. ***First, while the dataset is likely part of the GenAI agents' training corpora, the utterances posed by the participants are not.*** In our research, we focus much more on participant utterances and actions to GenAI agent responses, which we discuss as the structure and evolution of EVA conversations. A small number of participants across both studies (n=4) did have prior exposure to this dataset, but the sample was too small to show distinct patterns of utterances compared to those who had not been exposed. ***Second, even if the dataset is part of the training corpora, there isn't a guarantee that a GenAI agent will retrieve the correct response to a user's utterance***. Through usability checks of the session quality in both studies, we confirm that it can. We took a random set of code examples generated by the AI agent in our study and attempted to search the internet for identical examples, but did not find identical matches. This observation suggests, but does not confirm, that even for known datasets GenAI agents may generate code in our study through some combination of retrieval (e.g., find one or more answers for utterance) and modification (e.g., change the code for the dataset or context slightly) [32]. This is interesting behavior to further interrogate in a follow-up study. ***Third, it is not possible to predict precisely how different GenAI agents would be on datasets that are novel to the agent.*** We could assume it was worse, but prior research [6], and our own examinations (Appendix C), suggests its not significantly worse. If we make the assumption that performance would be worse on novel data, we might find that, perhaps, participants spend more time in 'refinement loops', over others. We may find an absence of branching structures, or loops altogether, as participants abandon use out of frustration. The past few sentences should show *how* our contributions still provide a useful, human-centric, grounding to broadly examining GenAI use on novel datasets going beyond technical measures like accuracy, functional correctness, or token matching widely used to currently assess GenAI agents [7]. ***Finally, we were thoughtful in how we incorporated and reflected upon the findings of prior work.*** We used established taxonomies of analysis tasks or editing operations to link and translate our findings to the broader space of prior techniques, approaches, and studies.

In summary, while the choice of analysis dataset poses limitations on our findings, we argue that it does not significantly curtail the generalization and transferability of our findings. Our contributions reaffirm and extend prior research on branching EVA conversational structure and can be used as a basis of comparison to other models and datasets.

7.3 Limitations

Beyond the dataset choice, there are two other primary limitations of our study. First, we did not extensively evaluate the design choices of AI Threads. This was largely because multi-threaded conversational interfaces have long been studied and applied. While we do assess the impact of this design choice in our studies as part of the overall session quality and did not find issues, the fact that we did not more deeply evaluate AI Threads may have impacted our results. A second limitation is that data workers still do not routinely use GenAI agents, which limits our ability to contrast to their existing workflows. This is for a variety of reasons, including concerns about data privacy and the present capabilities of the technology. For this reason, we were limited in our ability to more deeply probe the use of GenAI conversation agents in existing data worker practices, nor comment on their potential use beyond pure speculation.

8 CONCLUSION

Automation facilitated through conversational generative AI agents has the potential to disrupt the activities of data work. Our research conducts a human-centric examination of the perspectives of data workers and the nature of the exploratory visual analytic conversations they conduct with these agents. Our findings summarize the components and evolution of these conversations and show how data workers and agents cooperatively analyze and visualize data. Moreover, they point to avenues of future research that leverage these conversational components to improve the experience of data workers leveraging GenAI in their activities. The responses and capabilities of agents will undoubtedly change in light of the rapid pace of technical advancements. Our findings provide a grounding for assessing these evolving capabilities with data workers.

REFERENCES

[1] Open AI. 2024. https://platform.openai.com/docs/assistants/tools/code-interpreter

[2] Robert Amar, James Eagan, and John Stasko. 2005. Low-Level Components of Analytic Activity in Information Visualization. In *Proc. InfoVis'05 (INFOVIS '05)*. IEEE Computer Society, USA, 15. https://doi.org/10.1109/INFOVIS.2005.24

[3] Simran Arora, Avanika Narayan, Mayee F. Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, and Christopher Ré. 2022. Ask Me Anything: A simple strategy for prompting language models. *arXiv:2210.02441* (2022).

[4] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. 2021. Program Synthesis with Large Language Models. *ArXiv* abs/2108.07732 (2021). https://arxiv.org/abs/2108.07732

[5] Leilani Battle and Jeffrey Heer. 2019. Characterizing Exploratory Visual Analysis: A Literature Review and Evaluation of Analytic Provenance in Tableau. *Computer Graphics Forum* 38, 3 (2019), 145–159. https://doi.org/10.1111/cgf.13678

[6] Sebastian Bordt, Harsha Nori, Vanessa Rodrigues, Besmira Nushi, and Rich Caruana. 2024. Elephants Never Forget: Memorization and Learning of Tabular Data in Large Language Models. arXiv:2404.06209 [cs.LG] https://arxiv.org/

abs/2404.06209

[7] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, David W. Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Igor Babuschkin, S. Arun Balaji, Shantanu Jain, Andrew Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew M. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. *ArXiv* abs/2107.03374 (2021). https://arxiv.org/abs/2107.03374

[8] Liying Cheng, Xingxuan Li, and Lidong Bing. 2023. Is GPT-4 a Good Data Analyst? https://arxiv.org/abs/2305.15038

[9] Anamaria Crisan and Brittany Fiore-Gartland. 2021. Fits and Starts: Enterprise Use of AutoML and the Role of Humans in the Loop. In *Proc. CHI'21.* Article 601, 15 pages. https://doi.org/10.1145/3411764.3445775

[10] Anamaria Crisan, Brittany Fiore-Gartland, and Melanie Tory. 2021. Passing the Data Baton : A Retrospective Analysis on Data Science Work and Workers. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2021), 1860–1870. https://doi.org/10.1109/TVCG.2020.3030340

[11] Victor Dibia, Adam Fourney, Gagan Bansal, Forough Poursabzi-Sangdeh, Han Liu, and Saleema Amershi. 2023. Aligning Offline Metrics and Human Judgments of Value for Code Generation Models. In *Findings of the ACL'23.* 8516–8528.

[12] Victor C. Dibia. 2023. LIDA: A Tool for Automatic Generation of Grammar-Agnostic Visualizations and Infographics using Large Language Models. *ArXiv* abs/2303.02927 (2023). https://arxiv.org/abs/2303.02927

[13] Xuan Long Do, Mohammad Hassanpour, Ahmed Masry, Parsa Kavehzadeh, Enamul Hoque, and Shafiq Joty. 2023. Do LLMs Work on Charts? Designing Few-Shot Prompts for Chart Question Answering and Summarization. *arXiv preprint arXiv:2312.10610* (2023).

[14] Iddo Drori, Sarah Zhang, Reece Shuttleworth, Leonard Tang, Albert Lu, Elizabeth Ke, Kevin Liu, Linda Chen, Sunny Tran, Newman Cheng, Roman Wang, Nikhil Singh, Taylor L. Patti, Jayson Lynch, Avi Shporer, Nakul Verma, Eugene Wu, and Gilbert Strang. 2022. A neural network solves, explains, and generates university math problems by program synthesis and few-shot learning at human level. *Proceedings of the National Academy of Sciences* 119, 32 (Aug. 2022). https://doi.org/10.1073/pnas.2123433119

[15] Jaimie Drozdal, Justin Weisz, Dakuo Wang, Gaurav Dass, Bingsheng Yao, Changruo Zhao, Michael Muller, Lin Ju, and Hui Su. 2020. Trust in AutoML: Exploring Information Needs for Establishing Trust in Automated Machine Learning Systems. In *Proceedings of the 25th International Conference on Intelligent User Interfaces* (Cagliari, Italy) *(IUI '20).* 297–307. https://doi.org/10.1145/3377325.3377501

[16] Ethan Fast, Binbin Chen, Julia Mendelsohn, Jonathan Bassen, and Michael S. Bernstein. 2018. Iris: A Conversational Agent for Complex Tasks. In *Proc. CHI'18.* 1–12. https://doi.org/10.1145/3173574.3174047

[17] Donghui Feng, Erin Shaw, Jihie Kim, and Eduard Hovy. [n. d.]. An Intelligent Discussion-Bot for Answering Student Queries in Threaded Discussions. In *Proc. IUI'06* (Sydney, Australia). 171–177. https://doi.org/10.1145/1111449.1111488

[18] Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and Ming Zhou. 2020. CodeBERT: A Pre-Trained Model for Programming and Natural Languages. *ArXiv* abs/2002.08155 (2020). https://arxiv.org/abs/2002.08155

[19] Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Erkang, Zhu, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, Peter Chang, Ricky Loynd, Robert West, Victor Dibia, Ahmed Awadallah, Ece Kamar, Rafah Hosn, and Saleema Amershi. 2024. Magentic-One: A Generalist Multi-Agent System for Solving Complex Tasks. (2024). arXiv:2411.04468 https://arxiv.org/abs/2411.04468

[20] Thomas Cason Frank E. Harrell Jr. 2017. *Titanic dataset.* https://www.openml.org/d/40945

[21] Tong Gao, Mira Dontcheva, Eytan Adar, Zhicheng Liu, and Karrie G. Karahalios. 2015. DataTone: Managing Ambiguity in Natural Language Interfaces for Data Visualization. In *Proc. UIST '15.* https://doi.org/10.1145/2807442.2807478

[22] Ken Gu, Madeleine Grunde-McLaughlin, Andrew M. McNutt, Jeffrey Heer, and Tim Althoff. 2023. How Do Data Analysts Respond to AI Assistance? A Wizard-of-Oz Study. https://arxiv.org/abs/2309.10108

[23] Ken Gu, Ruoxi Shang, Tim Althoff, Chenglong Wang, and Steven M. Drucker. 2023. How Do Analysts Understand and Verify AI-Assisted Data Analyses? https://arxiv.org/abs/2309.10947

[24] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large Language Model Based Multi-agents: A Survey of Progress and Challenges. In *Proc. IJCAI'24.* 8048–8057. https://doi.org/10.24963/ijcai.2024/890 Survey Track.

[25] Yi Guo, Nana Cao, Xiaoyu Qi, Haoyang Li, Danqing Shi, Jing Zhang, Qing Chen, and Daniel Weiskopf. 2023. Urania: Visualizing Data Analysis Pipelines for Natural Language-Based Data Exploration. *ArXiv* (2023). https://arxiv.org/abs/2306.07760

[26] Yi Guo, Danqing Shi, Mingjuan Guo, Yanqiu Wu, Qing Chen, and Nana Cao. 2021. Talk2Data: A Natural Language Interface for Exploratory Visual Analysis via Question Decomposition. https://api.semanticscholar.org/CorpusID:258685266

[27] Yi He, Shixiong Cao, Yang Shi, Qing Chen, Ke Xu, and Nan Cao. 2024. Leveraging Large Models for Crafting Narrative Visualization: A Survey. *arXiv preprint arXiv:2401.14010* (2024).

[28] Marti Hearst and Melanie Tory. 2019. Would You Like A Chart With That? Incorporating Visualizations into Conversational Interfaces. In *2019 IEEE Visualization Conference (VIS)*. 1–5. https://doi.org/10.1109/VISUAL.2019.8933766

[29] Enamul Hoque, Vidya Setlur, Melanie Tory, and Isaac Dykeman. 2018. Applying Pragmatics Principles for Interaction with Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 309–318. https://doi.org/10.1109/TVCG.2017.2744684

[30] J. D. Hunter. 2007. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* 9, 3 (2007), 90–95. https://doi.org/10.1109/MCSE.2007.55

[31] Hilary Hutchinson, Wendy Mackay, Bo Westerlund, Benjamin B. Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stéphane Conversy, Helen Evans, Heiko Hansen, Nicolas Roussel, and Björn Eiderbäck. 2003. Technology Probes: Inspiring Design for and with Families. In *Proc. CHI'03*. Association for Computing Machinery, New York, NY, USA, 17–24. https://doi.org/10.1145/642611.642616

[32] Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. 2024. A Survey on Large Language Models for Code Generation. arXiv:2406.00515 [cs.CL] https://arxiv.org/abs/2406.00515

[33] Shankar Kantharaj, Rixie Tiffany Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. 2022. Chart-to-Text: A Large-Scale Benchmark for Chart Summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 4005–4023.

[34] Ecem Kavaz, Anna Puig, and Inmaculada Rodríguez. 2023. Chatbot-Based Natural Language Interfaces for Data Visualisation: A Scoping Review. *Applied Sciences* 13, 12 (Jun 2023), 7025. https://doi.org/10.3390/app13127025

[35] Mary Beth Kery and Brad A. Myers. 2018. Interactions for Untangling Messy History in a Computational Notebook. In *Proc (VL/HCC)'18'*. 147–155. https://doi.org/10.1109/VLHCC.2018.8506576

[36] Mary Beth Kery, Marissa Radensky, Mahima Arya, Bonnie E. John, and Brad A. Myers. 2018. The Story in the Notebook: Exploratory Data Science Using a Literate Programming Tool. In *Proc. CHI'18*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3173574.3173748

[37] Dae Hyun Kim, Enamul Hoque, and Maneesh Agrawala. 2020. Answering Questions about Charts and Generating Visual Explanations. In *Proc. CHI'20*. 1–13. https://doi.org/10.1145/3313831.3376467

[38] Younghoon Kim, Kanit Wongsuphasawat, Jessica Hullman, and Jeffrey Heer. 2017. GraphScape: A Model for Automated Reasoning about Visualization Similarity and Sequencing. In *Proc. CHI'17*. https://doi.org/10.1145/3025453.3025866

[39] Guozheng Li, Xinyu Wang, Gerile Aodeng, Shunyuan Zheng, Yu Zhang, Chuangxin Ou, Song Wang, and Chi Harold Liu. 2024. Visualization Generation with Large Language Models: An Evaluation. *arXiv preprint arXiv:2401.11255* (2024).

[40] Haotian Li, Yun Wang, Q. Vera Liao, and Huamin Qu. 2023. Why is AI not a Panacea for Data Workers? An Interview Study on Human-AI Collaboration in Data Storytelling. https://arxiv.org/abs/2304.08366

[41] Halden Lin, Dominik Moritz, and Jeffrey Heer. 2020. Dziban: Balancing Agency & Automation in Visualization Design via Anchored Recommendations. In *Proc. CHI'20*. 1–12. https://doi.org/10.1145/3313831.3376880

[42] Can Liu, Yun Han, Ruike Jiang, and Xiaoru Yuan. 2021. ADVISor: Automatic Visualization Answer for Natural-Language Question on Tabular Data. In *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)*. 11–20. https://doi.org/10.1109/PacificVis52677.2021.00010

[43] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the Middle: How Language Models Use Long Contexts. arXiv:2307.03172 [cs.CL]

[44] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 55, 9 (2023), 1–35.

[45] Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. Barack's Wife Hillary: Using Knowledge Graphs for Fact-Aware Language Modeling. In *Proc. ACL'19*. 5962–5971. https://doi.org/10.18653/v1/P19-1598

[46] Yuyu Luo, Nan Tang, Guoliang Li, Jiawei Tang, Chengliang Chai, and Xuedi Qin. 2022. Natural Language to Visualization by Neural Machine Translation. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (2022), 217–226. https://doi.org/10.1109/TVCG.2021.3114848

[47] Paula Maddigan and Teo Susnjak. 2023. Chat2VIS: Generating Data Visualizations via Natural Language Using ChatGPT, Codex and GPT-3 Large Language Models. *IEEE Access* 11 (2023), 45181–45193. https://doi.org/10.1109/ACCESS.2023.3274199

[48] Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In *Proc. ACL Findings'22*. 2263–2279. https://doi.org/10.18653/v1/2022.findings-acl.177

[49] Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. [n. d.]. UniChart: A Universal Vision-language Pretrained Model for Chart Comprehension and Reasoning. In *Proc. EMNLP'23*. 14662–14684. https://doi.org/10.18653/v1/2023.emnlp-main.906

[50] Michael McTear. 2020. *Conversational ai: Dialogue systems, conversational agents, and chatbots*. Morgan & Claypool.

[51] Tova Milo and Amit Somech. 2020. Automating Exploratory Data Analysis via Machine Learning: An Overview. *Proc. SIGMOD'20*, 2617–2622. https://doi.org/10.1145/3318464.3383126

[52] Robert J. Moore and Raphael Arar. 2018. *Conversational UX Design: An Introduction*. Springer International Publishing, Cham, 1–16. https://doi.org/10.1007/978-3-319-95579-7_1

[53] Michael Muller, Ingrid Lange, Dakuo Wang, David Piorkowski, Jason Tsay, Q. Vera Liao, Casey Dugan, and Thomas Erickson. 2019. How Data Science Workers Work with Data: Discovery, Capture, Curation, Design, Creation. In *Proc. CHI'19* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3290605.3300356

[54] Lorelli S. Nowell, Jill M. Norris, Deborah E. White, and Nancy J. Moules. 2017. Thematic Analysis: Striving to Meet the Trustworthiness Criteria. *International Journal of Qualitative Methods* 16, 1 (2017), 1609406917733847. https://doi.org/10.1177/1609406917733847 arXiv:https://doi.org/10.1177/1609406917733847

[55] phind inc. [n. d.]. Beating GPT-4 on HumanEval with a Fine-Tuned CodeLlama-34B. https://www.phind.com/blog/code-llama-beats-gpt4 accessed: 12/20/2023.

[56] Gabriel Poesia, Oleksandr Polozov, Vu Le, Ashish Tiwari, Gustavo Soares, Christopher Meek, and Sumit Gulwani. 2022. Synchromesh: Reliable code generation from pre-trained language models. *ArXiv* abs/2201.11227 (2022). https://arxiv.org/abs/2201.11227

[57] Eric D. Ragan, Alex Endert, Jibonananda Sanyal, and Jian Chen. 2016. Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes. *IEEE TVCG* 22, 1 (2016), 31–40. https://doi.org/10.1109/TVCG.2015.2467551

[58] Deepthi Raghunandan, Aayushi Roy, Shenzhi Shi, Niklas Elmqvist, and Leilani Battle. 2023. Code Code Evolution: Understanding How People Change Data Science Notebooks Over Time. In *Proc. CHI'23 (CHI '23)*. Article 863, 12 pages. https://doi.org/10.1145/3544548.3580997

[59] Sabbir M. Rashid, James P. McCusker, Paulo Pinheiro, Marcello P. Bax, Henrique O. Santos, Jeanette A. Stingone, Amar K. Das, and Deborah L. McGuinness. 2020. The Semantic Data Dictionary – An Approach for Describing and Annotating Data. *Data Intelligence* 2, 4 (10 2020), 443–486. https://doi.org/10.1162/dint_a_00058

[60] Donghao Ren, Matthew Brehmer, Bongshin Lee, Tobias Höllerer, and Eun Kyoung Choe. 2017. ChartAccent: Annotation for data-driven storytelling. In *Proc. PacificVis'17*. 230–239. https://doi.org/10.1109/PACIFICVIS.2017.8031599

[61] Adam Rule, Aurélien Tabard, and James D. Hollan. 2018. Exploration and Explanation in Computational Notebooks. In *Proc. CHI'18*. 1–12. https://doi.org/10.1145/3173574.3173606

[62] Vidya Setlur, Sarah E. Battersby, Melanie Tory, Rich Gossweiler, and Angel X. Chang. 2016. Eviza: A Natural Language Interface for Visual Analysis. In *Proc. UIST'16* (Tokyo, Japan). Association for Computing Machinery, New York, NY, USA, 365–377. https://doi.org/10.1145/2984511.2984588

[63] Vidya Setlur and Melanie Tory. 2022. How Do You Converse with an Analytical Chatbot? Revisiting Gricean Maxims for Designing Analytical Conversational Behavior. In *Proc. CHI'22*. Association for Computing Machinery, New York, NY, USA, Article 29, 17 pages. https://doi.org/10.1145/3491102.3501972

[64] Leixian Shen, Enya Shen, Yuyu Luo, Xiaocong Yang, Xuming Hu, Xiongshuai Zhang, Zhiwei Tai, and Jianmin Wang. 2023. Towards Natural Language Interfaces for Data Visualization: A Survey. *IEEE Transactions on Visualization and Computer Graphics* 29, 6 (2023), 3121–3144. https://doi.org/10.1109/TVCG.2022.3148007

[65] Arjun Srinivasan and Vidya Setlur. 2021. Snowy: Recommending Utterances for Conversational Visual Analysis. In *Proc. UIST'21 (UIST '21)*. Association for Computing Machinery, New York, NY, USA, 864–880. https://doi.org/10.1145/3472749.3474792

[66] Arjun Srinivasan and John Stasko. 2017. Natural Language Interfaces for Data Analysis with Visualization: Considering What Has and Could Be Asked. In *Proc EuroVis'17 Short Papers*. 55–59. https://doi.org/10.2312/eurovisshort.20171133

[67] Arjun Srinivasan and John Stasko. 2018. Orko: Facilitating Multimodal Interaction for Visual Exploration and Analysis of Networks. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 511–521. https://doi.org/10.1109/TVCG.2017.2745219

[68] Chase Stokes and Marti Hearst. 2022. Why More Text is (Often) Better: Themes from Reader Preferences for Integration of Charts and Text. In *IEEE Vis Workshop: Workshop on Exploring Opportunities and Challenges for Natural Language Techniques to Support Visual Analysis (NL Vis)*. IEEE. https://doi.org/10.48550/arXiv.2209.10789

[69] Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table Meets LLM: Can Large Language Models Understand Structured Table Data? A Benchmark and Empirical Study. In *Proc. WSDM '24*. 645–654. https://doi.org/10.1145/3616855.3635752

[70] Benny Tang, Angie Boggust, and Arvind Satyanarayan. 2023. VisText: A Benchmark for Semantically Rich Chart Captioning. In *Proc. ACL'2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Toronto, Canada, 7268–7298. https://doi.org/10.18653/v1/2023.acl-long.401

[71] Melanie Tory and Vidya Setlur. 2019. Do What I Mean, Not What I Say! Design Considerations for Supporting Intent and Context in Analytical Conversation. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*. 93–103. https://doi.org/10.1109/VAST47406.2019.8986918

[72] Jacob VanderPlas, Brian Granger, Jeffrey Heer, Dominik Moritz, Kanit Wongsuphasawat, Arvind Satyanarayan, Eitan Lees, Ilia Timofeev, Ben Welsh, and Scott Sievert. 2018. Altair: Interactive statistical visualizations for python. *Journal of open source software* 3, 32 (2018), 1057.

[73] C. Wang, J. Thompson, and B. Lee. 2024. Data Formulator: AI-Powered Concept-Driven Visualization Authoring. *IEEE TVCG* 30, 01 (2024), 1128–1138. https://doi.org/10.1109/TVCG.2023.3326585

[74] Dakuo Wang, Q Vera Liao, Yunfeng Zhang, Udayan Khurana, Horst Samulowitz, Soya Park, Michael Muller, and Lisa Amini. 2021. How Much Automation Does a Data Scientist Want? *arXiv preprint arXiv:2101.03970* (2021). https://arxiv.org/pdf/2101.03970.pdf

[75] Dakuo Wang, Justin D. Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. 2019. Human-AI Collaboration in Data Science: Exploring Data Scientists' Perceptions of Automated AI. *Proc. CSCW'19*, 24 pages. https://doi.org/10.1145/3359313

[76] Xindi Wang, Mahsa Salmani, Parsa Omidi, Xiangyu Ren, Mehdi Rezagholizadeh, and Armaghan Eshaghi. 2024. Beyond the Limits: A Survey of Techniques to Extend the Context Length in Large Language Models. In *Proc. IJCAI'24*. 8299–8307. https://doi.org/10.24963/ijcai.2024/917

[77] Michael L. Waskom. 2021. seaborn: statistical data visualization. *Journal of Open Source Software* 6, 60 (2021), 3021. https://doi.org/10.21105/joss.03021

[78] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. https://arxiv.org/abs/2201.11903

[79] Doris Xin, Eva Yiwei Wu, Doris Jung-Lin Lee, Niloufar Salehi, and Aditya Parameswaran. 2021. Whither AutoML? Understanding the Role of Automation in Machine Learning Workflows. In *Proc. CHI'21*. New York, NY, USA, Article 83, 16 pages. https://doi.org/10.1145/3411764.3445306

[80] Bowen Yu and Cláudio T. Silva. 2020. FlowSense: A Natural Language Interface for Visual Data Exploration within a Dataflow System. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2020), 1–11. https://doi.org/10.1109/TVCG.2019.2934668

[81] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large Language Models Are Human-Level Prompt Engineers. *ArXiv* abs/2211.01910 (2022). https://api.semanticscholar.org/CorpusID:253265328

[82] Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and Transferable Adversarial Attacks on Aligned Language Models. arXiv:2307.15043 [cs.CL]

Appendix

Appendix B provides additional technical details on the implementation of AI Threads. Appendix C provides examples of its use for various editing operations and understanding of different analytic intents. It also includes a demonstration of AI Threads on a dataset that is outside the training context of the LLM that we used in AI Threads's implementation.

A DESCRIPTION OF SUPPLEMENTAL MATERIALS

Here we provide an overview and a description of the content made available in our supplemental materials. These are available on OSF[5].

These online materials contain the following:

- The AI threads Code
- A zip file LLM_Comparison.zip) of the computation notebooks and additional Python scripts for comparing GenAI agents on visual analytic conversations
- A zip file (Study.zip) to the annotated (anonymized) participant conversations

Please note that running the AI Threads code and the LLM Comparison notebook requires that you have your own API keys. We provide examples of environment files needed to run our code.

B AI THREADS SYSTEM DESIGN

In this section we describe AI Threads's architecture that supports ingestion of user utterances and the display the result within the `Main Chat` and `Thread`.

B.1 Data Sources

Participants can upload a single CSV dataset to carry out their data analysis. A string referring to the file's location on the local system is also stored and passed to GPT-3.5 as part of its code generation. When a dataset is uploaded, using the procedure outlined in Figure B.1, AI Threads will automatically generate a `Data Dictionary` which includes each column's name, data type, and a value range (for nominal or ordinal columns) or a non-null example value (for all other data types). The LLM also generates a natural language description of each field.

The `Data Dictionary` is used as part of future instructional system prompts to generate valid and executable code; it is also displayed to end users (Section 3.3) as an editable reference table, where they can alter descriptions to provide more accurate information for the AI. These descriptions help the LLM identify and use the appropriate attribute and reason about its values (e.g., the utterances "*Filter gender by male*" vs. "*show only male*"), which supports a more naturalistic conversation out-of-the-box. We believe that descriptions about columns can also elicit higher-order reasoning from the LLM when necessary, for example whenever the user prompts the model to "*show me something interesting,*" by leveraging knowledge about semantically related concepts captured within the LLMs embedding space. Future work presents interesting possibilities for extending this approach by augmenting the LLM with a knowledge graph [45].

AI Threads can leverage knowledge that is external to the uploaded dataset as part of GPT-3.5's training data[6] to augment their existing analysis. For example, with the Titanic dataset participants can ask *"Who was the captain on the Titanic?"*, which is not available in the dataset itself.

---

[5] https://osf.io/6wxpa

[6] At the time of this writing, the details of GPT-3.5's training data have not been publicly disclosed.

Using a pandas script, AI Threads creates a markdown table from the uploaded dataset that includes column names, data types, and data values structured as follows:

```
|            | Data Type   | Range or Example       |
|:-----------|:-----------|:-----------------------|
...
```

This markdown table and the original filename are used to produce the full `Data Dictionary` using GPT-3.5. Before the model is queried, we give it the following system prompt:

System

```
We will generate markdown tables that begin with the following columns:
|            | Data Type   | Range or Example       | Description
             |
|:-----------|:-----------|:-----------------------|:-----------------------------|
```

Finally, the following user message is sent to GPT-3.5:

User

```
Here is a markdown table summarizing columns in [FILENAME]:
[PANDAS TABLE]
Generate descriptions for what the data column values mean and
concatenate the descriptions to the markdown table.
```

Fig. B.1. **Instruction prompt for generating a `Data Dictionary` of an uploaded dataset.**

B.2 Code Generation

User utterances initiate the generation and execution of Python code to apply the appropriate set of data transformations using `pandas` and produce visualizations using `seaborn` [77] and `matplotlib` [30]. In addition to producing a visual encoding, GPT-3.5 is also instructed to produce a text caption describing the encoding choices. Depending on how the utterance is posed, GPT-3.5 sometimes also generates an interpretation of what is shown.

B.2.1 Processing User Utterances. Prior V-NLI and conversational interfaces have used a variety of methods, from parse trees to heuristics, to decompose a participant's utterance, deduce its intent, and generate an appropriate output. We use GPT-3.5 to perform these tasks, providing it with instructional prompts to guide its visualization code generation or refinement. Subsequently, the relevant visualization code is extracted from GPT-3.5's original responses to display charts and captions to the analyst.

B.2.2 System Instructional Prompts. The GPT-3.5 ChatCompletion API delineates three role types to manage multi-turn conversations: system, assistant, and user. System instructional prompts can define an LLM's **role** and **desired behaviors** and augment the user messages to achieve a desired response.

In the `Main Chat`, the full instructional prompt provided to GPT-3.5 contains the agent's **role**, which is to answer questions about the dataset with visualizations, and **desired behaviors**, which is to render and display a chart with an accompanying caption). The system is also prompted to use **chain-of-thought (CoT) reasoning**, which has been shown to improve the performance of LLM code generation [78]. The CoT prompt is reinforced with a set of few-shot examples (See Section

In the `Main Chat`, the following system instructions are provided to GPT-3.5:

System

```
You are a data visualization assistant working with a dataset called [
FILENAME].
You will answer questions with this dataset.
When you do not use the dataset, say "I can't answer this question with
the data,
but I found this:" You will use pandas and seaborn libraries to write
Python
code which prints a single PNG image in a bytes64 representation. You
will then
also print an alt-text caption for the chart.
Let's work responses out step by step to be sure you have the right
answer.
The dataset: [DATASET DICTIONARY]
```

In the background, user conversations with the LLM are prepended with few-shot input-output examples as conversational context, e.g.:

User

```
How many rows are in the dataset?
```

Assistant

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
...

# Load the dataset
df = pd.read_csv('./workspace/{filename}')

# Get the number of rows in the DataFrame
num_rows = df.shape[0]

# Create the bar chart using seaborn, adjusting the size of the chart as
 needed
width = 8
height = 2
plt.figure(figsize=(width, height))
sns.barplot(x=[num_rows])
...

# Print a caption
print("This plot shows a single bar representing the total number of
rows in the dataset, which is " + str(num_rows) + ".")
```

The example response is truncated; see the full examples in Appendix B.3.

Fig. B.2. **Pre-defined context in the `Main Chat`.** Although not exposed to the user, an exchange with the GPT-3.5 CodeCompletion API in the `Main Chat` will include a system prompt and few-shot examples along with previous exchanges.

System

```
Today we are being asked to revise the following code.
The revised code should print a new chart in a bytes64 PNG
representation and
print a new alt-text caption in the same way as the original code.
The code: [ORIGINAL CODE]
The dataset [FILENAME] contains: [DATASET DICTIONARY]
```

Fig. B.3. **Instructions in a `Thread`.** The role of GPT-3.5 in a `Thread` is to extend an analysis branch, using the **original visualization** as a reference for refinement or continuation.

B.2.3). Lastly, the system message includes the previously generated **data dictionary**. Crafting such instructional system prompts can require specialized knowledge to produce via a manual trial-and-error process [44], and thus is something we do not expose to the user. We refined our prompting strategy over time and found it to qualitatively improve the chatbot's responses without the need for the user to input additional instructions.

In the `Thread`, the agent is given a different **role**, which is to revise a previously generated visualization code to change visual encodings or make stylistic changes. A user might also ask the model to present additional data columns, to aggregate the data in a new way, or to filter the data, which again requires information from the **data dictionary**. However, the design of this instruction ensures that each user prompt is interpreted to be a question or a command that continues or modifies a specific analysis (**the original code**), as opposed to starting a new analysis that will require the synthesis of fundamentally different code.

B.2.3 Few-Shot Examples. In the `Main Chat`, the initial system message is followed by few-shot examples of pre-designed **user** utterances and the **assistant's** code responses (Figure B.2). These few-shot examples include how to transform data with `pandas` and use `seaborn` to achieve the desired image and text caption outputs. Utterances in the few-shot examples were: "*How many rows are in the dataset?*" which teaches AI Threads to display simple numeric responses as both a bar chart and within a text caption, and "*Plot the distribution of every continuous measure,*" which signals how it should display visualizations with multiple facets. Both questions are also agnostic to the semantics of specific datasets.

B.3 Few-Shot Examples

The following are the few shot examples provided to the GPT-3.5 ChatCompletion API via AI Threads as conversational context following the system instructional prompt (Section B.2).

During pilot studies, we noticed that GPT-3.5 tended to hallucinate semantically relevant yet non-existent colors or color palettes when generating or modifying visualizations from color-related user utterances, unless it learned how to specify colors in the demonstrated approach. Defining every variable in the examples, even the chart's `width` and `height`, before they are used also allowed the model to learn how to correctly modify charts. The model is also taught how to incorporate computations to its captions.

Example 1:

[ROLE: USER]

```
How many rows are in the dataset?
```

[ROLE: ASSISTANT]

~~~
```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import base64
import io

tab10 = ["#4e79a7","#f28e2c","#e15759","#76b7b2","#59a14f","#edc949","#af7aa1",
"#ff9da7","#9c755f","#bab0ab"]
sns.set_palette(tab10)
sns.set(style="whitegrid")

# Load the dataset
df = pd.read_csv('./workspace/{filename}')

# Get the number of rows in the DataFrame
num_rows = df.shape[0]

# Create the bar chart using seaborn, adjusting the size of the chart as needed
width = 8
height = 2
plt.figure(figsize=(width, height))
sns.barplot(x=[num_rows])

# Add labels, annotations, and/or title
plt.xlabel('Number of Rows')

# Display the chart as a Bytes64 PNG representation with a tight layout
plt.tight_layout()
buffer = io.BytesIO()
plt.savefig(buffer, format='png')
buffer.seek(0)
png_data = base64.b64encode(buffer.getvalue()).decode('utf-8')
buffer.close()

plt.clf()
plt.close()

# Print the chart image
print(png_data)

# Print a caption
print("This plot shows a single bar representing the total number of rows in
the dataset, which is " + str(num_rows) + ".")
```
~~~

Example 2:

[ROLE: USER]

```
Plot the distributions of every continuous measure.
```

[ROLE: ASSISTANT]

~~~

```python
import pandas as pd
~~~

~~~
import seaborn as sns
import matplotlib.pyplot as plt
import base64
import io

tab10 = ["#4e79a7","#f28e2c","#e15759","#76b7b2","#59a14f","#edc949","#af7aa1",
"#ff9da7","#9c755f","#bab0ab"]
sns.set_palette(tab10)
sns.set(style="white")

# Load the dataset
df = pd.read_csv('./workspace/{filename}')

# Selecting the continuous variables from the DataFrame
continuous_vars = []

for col_name, dtype in df.dtypes.iteritems():
    if dtype==float:
        continuous_vars.append(col_name)

# Creating a new DataFrame with only the continuous variables
continuous_df = df[continuous_vars]

# Melt the DataFrame to create a long-form representation
melted_df = pd.melt(continuous_df)

# Plotting a faceted histogram
g = sns.FacetGrid(melted_df, col="variable", col_wrap=2, sharex=False, sharey=
False)
num_bins = 20
g.map(sns.histplot, "value", bins=num_bins)
g.set_titles("{col_name}")
g.set_axis_labels("", "Count")

# Display the chart as a Bytes64 PNG representation with a tight layout
plt.tight_layout()
buffer = io.BytesIO()
plt.savefig(buffer, format='png')
buffer.seek(0)
png_data = base64.b64encode(buffer.getvalue()).decode('utf-8')
buffer.close()

plt.clf()
plt.close()

# Print the chart image
print(png_data)
# Print a caption
print("This plot shows a grid of histograms, each representing a different
continuous variable. The x-axis represents the value range of each variable,
and the y-axis represents the count of occurrences.")
```
~~~

B.3.1 Code Extraction and Execution. Responses from the GPT-3.5 ChatCompletion API are often not valid Python code excerpts that can be directly executed by AI Threads. Given that GPT-3.5 is a successor to both Codex and GPT-3 models, its responses may contain 'friendly' textual responses (usually beginning with "Sure!" [82]) or explanations of a step-by-step approach in addition to code. In order to prioritize analytic coherence, AI Threads only extracts a valid code block from model responses, which include the chart's caption. When valid code blocks can be extracted, they are automatically executed and outputs (visualization and text) are pushed to the front-end interface. If valid code blocks are not generated, AI Threads responds by displaying the initial model response to the user.

For a variety of reasons, including API time-outs, the model may not generate executable code. In these cases, the participant can reattempt to generate the response from the front-end interface without the need to re-enter the utterance or provide a new one (Section 3.3). In other cases, the model may also output a simple text response, for example returning the average fare paid as a number instead of producing a visualization. In some instances, the text outputs of AI Threads pose clarifying questions to the analyst, for example in responding to the utterance *"tell me something that is cool"* an example response is *"as an AI assistant, I don't have access to personal information or preferences unless shared with me. If you have any specific preferences or questions, please feel free to share them, and I'll do my best to assist you".*

B.3.2 Different Thread, Different Context. AI Threads provides different system prompts and conversational contexts to GPT-3.5 depending on whether the user is interacting with the `Main Chat` and `Thread` panel (Figure 3). The conversational context is further tailored to a thread the user is inputting utterances to (Figure B.3). Code that is generated within a thread is appended as prior context to subsequent utterances. Alternating between `Thread` and `Main Chat` allows the user can proactively manage context. This is because the `Thread` panel contains only the context relevant to that particular visualization and analysis while the `Main Chat` will only include the utterance and results of charts as refined in each thread.

B.3.3 Alternative Visualization Libraries. We explored other visualization libraries but found that they resulted in less reliable responses. In some instances, the libraries were too new to produce reliably executable code, as was the case for `altair` [72]. We also explored different languages for visualization specifications. Prior work [12, 56] has explored the use of JSON-formatted `Vega-Lite` specifications to produce visualizations using LLMs, but we did not use these approaches for two reasons. First, using `seaborn` allows for direct integration with its dependencies `statsmodels`, `scipy`, and `fastcluster` for advanced data modeling. Keeping code for the data operations and visualization in the same ecosystem improved AI Threads's ability to reliably generate executable code. Second, one advantage of using `seaborn` [77] or `matplotlib` was that GPT-3.5 would produce comments to document its data transformations and visualization processes, which is a useful exposition of its chain-of-thought reasoning [14] that we could leverage in the future. However, getting equivalent information from a JSON specification proved to be difficult.

B.4 Preliminary Assessments of Performance

In Appendix C we present examples of AI Threads responding to visualization editing operations that were described by Kim *et. al.* [38] and to different visual analytic intents described by Srinivasan *et. al.* [65]. We also provide a walk-through of AI Threads using a dataset that we know is not found in the training data for our agent. Here, we present a short assessment of the impact of agent choice. We replay a subset of visual analytic conversations from a prior study [71] that interrogated the role of iterative refine via sequential utterances. In Figure B.4 we show an example conversation.

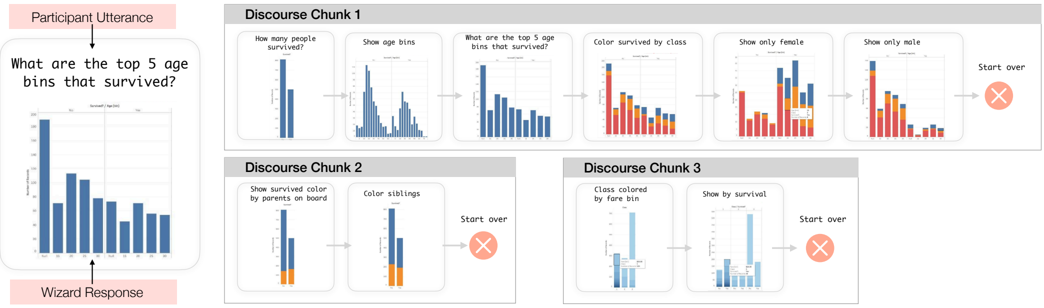


Fig. B.4. Example of a Visual Analytic Conversation between a participant and Wizard from Tory and Setlur [71]. Participants could choose to iterate over the existing visualization or start a new one. Certain utterances (e.g., 'start over') were explicit signals to begin a new visualization, but the Wizard could also make their own judgments. Tory and Setlur [71] segmented conversations into discourse chunks. Context is retained with a chunk, to modify the prior visualization, and is shifted in between chunks. In this example conversation, a participant made 13 utterances, which could be broken down into three discourse chunks. We replay conversations from [71] to conduct preliminary assessments ahead of conducting user studies.

B.4.1 Approach. We compared Code Interpreter (GPT-4[7]), GPT3.5[8], and CodeLlama[9]. For GPT-3.5 and CodeLLama, we provided the full conversational context up to the token limit, after which we excluded earlier utterances. Code Interpreter (GPT-4) had its own mechanisms for context management. We provide the utterances from Tory and Setlur [71] to these agents and record their results. Correctness alone is not a sufficient criterion for evaluating automatically generated code, there is a lot of subjectivity in these assessments [11]. We appraise the agents' responses on the following criteria: 1) is the code executable? 2) does it produce a visualization? and 3) is the response appropriate for the utterance? To assess the appropriateness of visualizations, we consider the chart type and the dataset attributes that are visualized. We manually reviewed the responses and made general assessments as to their appropriateness. In the supplemental material, we include a computational notebook that outlines additional considerations for conducting this re-analysis.

B.4.2 Assessments. The primary challenge we encountered in making these assessments was the non-deterministic nature of the agents' responses, where the same utterance yielded different results on separate occasions. In an extreme case, the given agent might produce non-executable code and upon being re-run provide a correct response – although such instances were infrequent. Notably, CodeLlama was more likely to produce non-executable code even in spite of multiple re-runs. When the agent generated executable code, the response was either a text or visualization. Code Interpreter exhibited a higher inclination toward textual responses compared to GPT-3.5. At times a text response is warranted, for example, the utterance *"how many people survived?"* could reasonably produce a visualization or a simple text response. For other utterances, for example *"What are the top 5 age bins that survived?"*, Code Interpreter also tended to produce text responses when we deemed a visualization might be more appropriate. When considering the content of the visualization responses, it is important to note that they did not identically match the results from [71] but were still correct. For example, the Wizard tended to use facets regularly (see

---

[7] We used the Assistants API with gpt-4-1106-preview and with code interpreter as a tool.

[8] GPT-3.5-turbo

[9] We explored models with different parameter sizes (7B, 13B, and 34B) and fine-tuning regimens. Here, we only report on the results of a 34B parameter model that was fine-tuned on an additional dataset of approximately 80K examples[55].

Figure B.4) whereas the GenAI agents used other encoding channels, like color or position. Similar to the Wizard, the agents commonly produced bar charts. An important, but complex, factor for assessing an agent's responses was their ability to retain context within a chunk, and thus build on the previous utterance, or shift context when appropriate. We observed that agents behaved inconsistently with context switches and retention. For example, they would 'forget' variables from prior utterances, treating them as independent instead of sequential. The inclusion of anaphoric references [16] improves this behavior, but, participants did not use them frequently or consistently. Additionally, utterances indicating context shifts (e.g., 'start over') were not consistently interpreted, aligning with findings from Tory and Setlur's study [71] that even human wizards experience difficulties in context management.

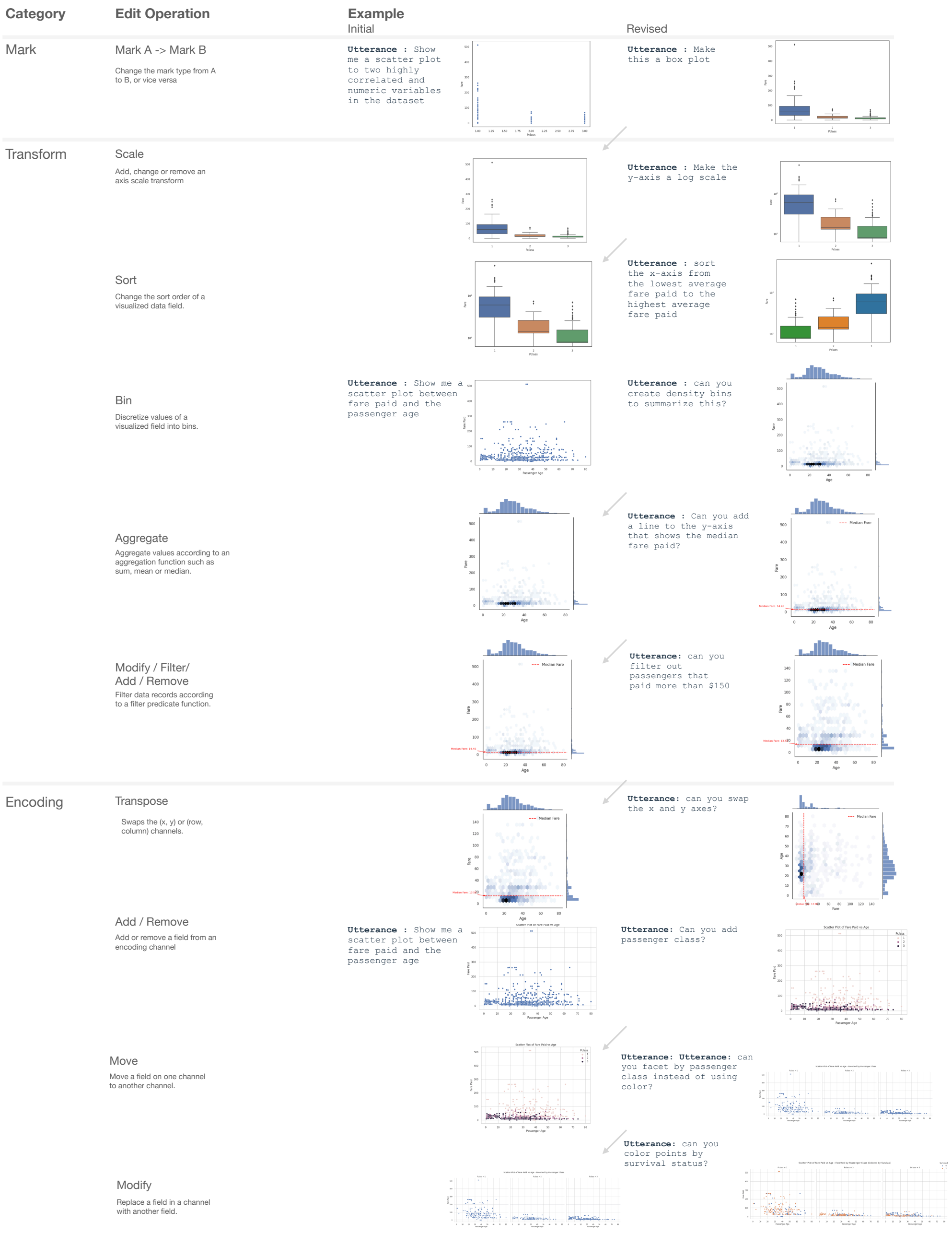
A final assessment we made was to ensure that the model was actually evaluating the dataset provided. To make this assessment we perturbed the values of the dataset, changed column names, and made other modifications to ensure the agent was analyzing the dataset provided and not just reproducing responses from its training data. Finally, we also compared to Code Interpreter [1] to ensure response consistency. Overall, we primarily assessed whether the agent consistently produced executable code that, when appropriate, displayed a visualization. In our user studies, we further investigate the appropriateness of the agent's responses to participant utterances.

Overall, this preliminary assessment indicated that GPT-3.5 consistently produced visualizations, often of equal or superior quality to GPT-4, and demonstrated significantly faster response times. Consequently, GPT-3.5 was selected as the generative AI agent for subsequent studies. The observations from this assessment influenced the context management strategy for the system (Section 3.3).

C AI THREADS ADDITIONAL DETAILS

C.1 Editing Actions & Utterance Intent Examples

AI Threads can modify a visual encoding in a variety of ways, in Figure C.1 we provide a concrete set of examples from the edit operations described in GraphScape [38]. Additionally, we explored how AI Threads responds utterances posed with different types of analytic intents. We draw on examples utterances from the SNOWY system [65]. Figure C.2 shows an example of the utterance, the resulting visualization, and a text output generated by the system to describe the data visualization that was generated. *Note that due to the size of the figures, they follow the subsequent Section B.3.*



Category	Edit Operation	Example: Initial	Example: Revised
Mark	Mark A -> Mark B Change the mark type from A to B, or vice versa	**Utterance** : Show me a scatter plot to two highly correlated and numeric variables in the dataset	**Utterance** : Make this a box plot
Transform	Scale Add, change or remove an axis scale transform		**Utterance** : Make the y-axis a log scale
	Sort Change the sort order of a visualized data field.		**Utterance** : sort the x-axis from the lowest average fare paid to the highest average fare paid
	Bin Discretize values of a visualized field into bins.	**Utterance** : Show me a scatter plot between fare paid and the passenger age	**Utterance** : can you create density bins to summarize this?
	Aggregate Aggregate values according to an aggregation function such as sum, mean or median.		**Utterance** : Can you add a line to the y-axis that shows the median fare paid?
	Modify / Filter/ Add / Remove Filter data records according to a filter predicate function.		**Utterance:** can you filter out passengers that paid more than $150
Encoding	Transpose Swaps the (x, y) or (row, column) channels.		**Utterance:** can you swap the x and y axes?
	Add / Remove Add or remove a field from an encoding channel	**Utterance** : Show me a scatter plot between fare paid and the passenger age	**Utterance:** Can you add passenger class?
	Move Move a field on one channel to another channel.		**Utterance: Utterance:** can you facet by passenger class instead of using color?
	Modify Replace a field in a channel with another field.		**Utterance:** can you color points by survival status?

Fig. C.1. **Example editing operations.** The arrows indicate that the editing operation was continues in a thread. The above figure show three prompts producing a encoding that is the further modified via editing operations in a thread. Arrows indicate a continuation from the previously refined encoding. Edit operations were taken from [38].

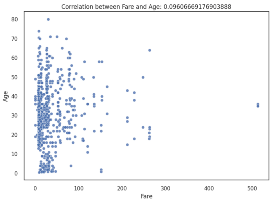
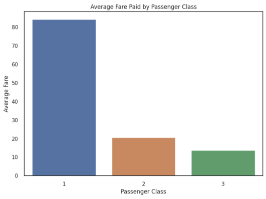
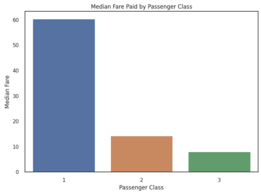
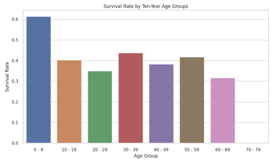
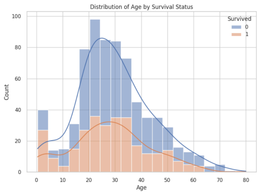
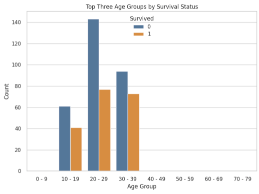
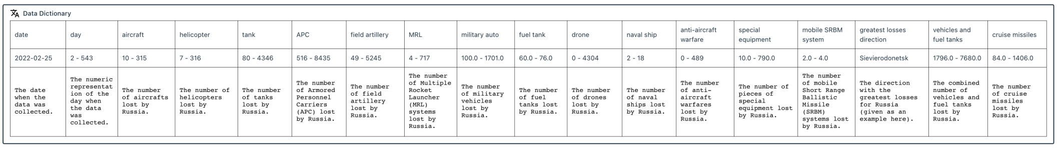
Intent	**Example** Utterance	Visualization	Output Text
Correlate	How are fare and passenger's age correlated?		This scatter plot shows the relationship between Fare and Age of the passengers. The x-axis represents the Fare amount paid by the passengers, while the y-axis represents their age. The dots represent individual passengers. The correlation coefficient between Fare and Age is 0.09606669176903888, indicating the strength and direction of the linear relationship between the two variables.
Group	On average what was the total fare paid by class?		This bar plot shows the average fare paid by passenger class. The x-axis represents the passenger class (1st, 2nd, 3rd), and the y-axis represents the average fare. The height of each bar represents the average fare paid by passengers in that class. It can be observed that on average, passengers in 1st class paid the highest fare, followed by 2nd class and then 3rd class.
Aggregate	Show median fare paid instead		This bar plot shows the median fare paid by passenger class. The x-axis represents the passenger class (1st, 2nd, 3rd), and the y-axis represents the median fare. The height of each bar represents the median fare paid by passengers in that class. It can be observed that the median fare paid increases from 3rd class to 1st class, indicating a higher fare for higher passenger classes.
Trend	Show the survival by 10 year age groups		This bar plot shows the survival rate for each ten-year age group. The x-axis represents the ten-year age groups, and the y-axis represents the survival rate. The height of each bar represents the survival rate of passengers in that age group. It can be observed that the survival rate is highest for children (0 - 9 years) and decreases as the age group increases until it reaches the lowest for the elderly (70 - 79 years).
Distribution	What was the age age distribution by survival status?		This histogram shows the distribution of age by survival status. The x-axis represents the age of the passengers, and the y-axis represents the count of passengers. The histogram is stacked, with the different survival statuses (0 = Not Survived, 1 = Survived) represented by different colors. The vertical axis also includes a smoothed density estimate (Kernel Density Estimate) to visualize the underlying distribution. It can be observed that there is a higher count of young passengers who did not survive compared to those who survived.
Filter	Show the top three age groups by survival status		This bar plot shows the count of survivors and non-survivors in the top three age groups. The x-axis represents the age group, and the y-axis represents the count of passengers. The bars are stacked, with different colors representing the survival status (0 = Not Survived, 1 = Survived) within each age group. It can be observed that the highest count of survivors is in the age group 20-29, followed by the age group 30-39, while the highest count of non-survivors is in the age group 20-29, followed by the age group 30-39.

Fig. C.2. **Examples of AI Threads responds to utterances with different intents**. Utterances are inspired by those suggested in [65].



Data Dictionary

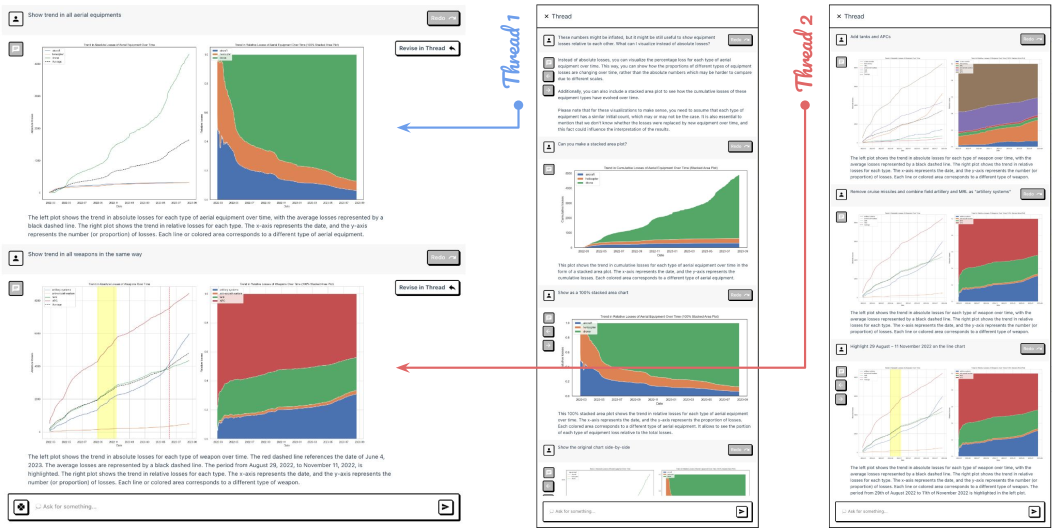
date	day	aircraft	helicopter	tank	APC	field artillery	MRL	military auto	fuel tank	drone	naval ship	anti-aircraft warfare	special equipment	mobile SRBM system	greatest losses direction	vehicles and fuel tanks	cruise missiles
2022-02-25	2 - 543	10 - 315	7 - 316	80 - 4346	516 - 8435	49 - 5245	4 - 717	100.0 - 1701.0	60.0 - 76.0	0 - 4304	2 - 18	0 - 489	10.0 - 790.0	2.0 - 4.0	Sievierodonetsk	1796.0 - 7680.0	84.0 - 1406.0
The date when the data was collected.	The numeric representation of the day when the data was collected.	The number of aircrafts lost by Russia.	The number of helicopters lost by Russia.	The number of tanks lost by Russia.	The number of Armored Personnel Carriers (APC) lost by Russia.	The number of field artillery lost by Russia.	The number of Multiple Rocket Launcher (MRL) systems lost by Russia.	The number of military vehicles lost by Russia.	The number of fuel tanks lost by Russia.	The number of drones lost by Russia.	The number of naval ships lost by Russia.	The number of anti-aircraft warfares lost by Russia.	The number of pieces of special equipment lost by Russia.	The number of mobile Short Range Ballistic Missile (SRBM) systems lost by Russia.	The direction with the greatest losses for Russia (given as an example here).	The combined number of vehicles and fuel tanks lost by Russia.	The number of cruise missiles lost by Russia.

Fig. C.3. **Ukraine-Russian war dataset**. The `Data Dictionary` for this dataset was automatically derived by AI Threads.

C.2 Example on an Additional Dataset

Finally, we demonstrate the use of AI Threads on a dataset collected after GPT-3.5's knowledge-cutoff date of September 2021. The Armed Forces of Ukraine has published data related to its military operations since the first week of the war, which began in 2022. A structured dataset was compiled from the Ukranian government's website as daily reports and continues to also be updated weekly on Kaggle. The dataset contains information about the cumulative number of losses per day sustained by Russian forces, including both military personnel and equipment. We provide walkthrough of a data analyst, Sam, using AI Threads. A summary of Sam's analysis is shown in Figure C.4.

Sam wants to quickly explore the dataset to find compelling visualizations that can be used to support their investigation of the Ukrainian-Russian war. They upload the dataset and AI Threads generates a data dictionary including the description of each column (Figure C.3). Sam begins by typing: *"Show trend in each equipment."* AI Threads correctly responds with a line chart displaying the cumulative loss of each Russian piece of equipment throughout the war. However, all 15 equipment types are difficult to distinguish using a single color palette. Sam directs AI Threads to '*'Show trend in all aerial equipments"*. The simplified line chart now clearly communicates that Russian drone losses are rising day by day. This is in stark contrast to the stagnation in the cumulative losses of aircraft and helicopters, even though the war has escalated.



Redo
Revise in Thread
Trend in Absolute Losses of Aerial Equipment Over Time
Trend in Relative Losses of Aerial Equipment Over Time (100% Stacked Area Plot)
aircraft
helicopter
drone
Average
Date
Thread 1
Thread 2
× Thread
Trend in Cumulative Losses of Aerial Equipment Over Time (Stacked Area Plot)
Cumulative losses
Trend in Relative Losses of Aerial Equipment Over Time (100% Stacked Area Plot)
Trend in Absolute Losses of Weapons Over Time
Trend in Relative Losses of Weapons Over Time (100% Stacked Area Plot)
artillery systems
anti-aircraft warfare
tank
APC
Ask for something...

Show trend in all aerial equipments

The left plot shows the trend in absolute losses for each type of aerial equipment over time, with the average losses represented by a black dashed line. The right plot shows the trend in relative losses for each type. The x-axis represents the date, and the y-axis represents the number (or proportion) of losses. Each line or colored area corresponds to a different type of aerial equipment.

Show trend in all weapons in the same way

The left plot shows the trend in absolute losses for each type of weapon over time. The red dashed line references the date of June 4, 2023. The average losses are represented by a black dashed line. The period from August 29, 2022, to November 11, 2022, is highlighted. The right plot shows the trend in relative losses for each type. The x-axis represents the date, and the y-axis represents the number (or proportion) of losses. Each line or colored area corresponds to a different type of weapon.

These numbers might be inflated, but it might be still useful to show equipment losses relative to each other. What can I visualize instead of absolute losses?

Instead of absolute losses, you can visualize the percentage loss for each type of aerial equipment over time. This way, you can show how the proportions of different types of equipment losses are changing over time, rather than the absolute numbers which may be harder to compare due to different scales.

Additionally, you can also include a stacked area plot to see how the cumulative losses of these equipment types have evolved over time.

Please note that for these visualizations to make sense, you need to assume that each type of equipment has a similar initial count, which may or may not be the case. It is also essential to mention that we don't know whether the losses were replaced by new equipment over time, and this fact could influence the interpretation of the results.

Can you make a stacked area plot?

This plot shows the trend in cumulative losses for each type of aerial equipment over time in the form of a stacked area plot. The x-axis represents the date, and the y-axis represents the cumulative losses. Each colored area corresponds to a different type of aerial equipment.

Show as a 100% stacked area chart

This 100% stacked area plot shows the trend in relative losses for each type of aerial equipment over time. The x-axis represents the date, and the y-axis represents the proportion of losses. Each colored area corresponds to a different type of aerial equipment. It allows to see the portion of each type of equipment loss relative to the total losses.

Show the original chart side-by-side

Add tanks and APCs

The left plot shows the trend in absolute losses for each type of weapon over time, with the average losses represented by a black dashed line. The right plot shows the trend in relative losses for each type. The x-axis represents the date, and the y-axis represents the number (or proportion) of losses. Each line or colored area corresponds to a different type of weapon.

Remove cruise missiles and combine field artillery and MRL as "artillery systems"

The left plot shows the trend in absolute losses for each type of weapon over time, with the average losses represented by a black dashed line. The right plot shows the trend in relative losses for each type. The x-axis represents the date, and the y-axis represents the number (or proportion) of losses. Each line or colored area corresponds to a different type of weapon.

Highlight 29 August – 11 November 2022 on the line chart

The left plot shows the trend in absolute losses for each type of weapon over time, with the average losses represented by a black dashed line. The right plot shows the trend in relative losses for each type. The x-axis represents the date, and the y-axis represents the number (or proportion) of losses. Each line or colored area corresponds to a different type of weapon. The period from 29th of August 2022 to 11th of November 2022 is highlighted in the left plot.

Fig. C.4. **Example analysis for Ukraine-Russian war dataset**. Two visualizations and shown in the `Main Chat` area in addition to their modifications via two separate `Thread` panels.

Sam understands that since the data is maintained by the Ukrainian forces, it may inflate the number of true Russian equipment losses. So Sam opens a `Thread` next to the line chart and asks: *"These numbers might be inflated, but it might be still useful to show equipment losses relative to each other. What can I visualize instead of absolute losses?"*. AI Threads responds by recommending a stacked area plot to present the data, so Sam further elaborates by first uttering *"Can you make a stacked area plot?"* and subsequently to *"Show as a 100% stacked area chart"* and finally to *"Show the original chart side-by-side"* – the result is combining line and area charts can highlight both absolute and proportional values. Finally, the line chart would seem more insightful if it depicted the average loss across all equipment types as a baseline to compare the trends of each piece of equipment. Sam directs AI Threads to draw the average loss as a dashed line on the line chart and exits the `Thread`.

Sam is pleased with their first visualization, which they see now in the `Main Chat` with the AI. However, the visualization only depicts a small subset of the data. Interested in building a narrative about well-known Ukrainian counteroffensives, Sam directs AI Threads to *"Show trend in all weapons in the same way"* (i.e., the same way as the previous result which Sam heavily refined in a separate `Thread`). Although AI Threads's new visualization correctly presents some weapon losses as a paired line and area chart, Sam is not entirely happy with the weapons shown.

Sam opens a `Thread` next to the figure. They direct AI Threads to *"Add tanks and APCs"* since tanks and APCs (Armored Personnel Carriers) are also used in offensive operations. Sam then directs the AI to *"Remove cruise missiles and combine field artillery and MRL as artillery systems."* Cruise missiles are single-use ammunition, unlike the rest which are multi-use equipment; field artillery and MRLs are both mid-range artillery systems. Sam now notices visual trends emerging in the line chart: two notable periods where there were relatively steeper inclines. They wonder if they correlate with the launch of major counteroffensives. Sam directs AI Threads to *"Highlight 29 August – 11 November 2022 on the line chart"* the duration of Ukraine's first counteroffensive, and is pleased to see the first dramatic increase across Russian weapon losses overlaying perfectly with the highlighted period.

Finally, the second counteroffensive was launched on the 4th of June, 2023 [10]. Sam directs AI Threads to *"Draw a reference line on June 4, 2023"* and notices that the dramatic rise in Russian artillery systems lost corresponds reasonably with the date. Sam exits the `Thread` and is ready to share the `Main Chat`, complete with charts plus titles and captions explaining their visual elements, with their colleagues at their next meeting.

---

[10] https://www.reuters.com/world/europe/russia-says-its-forces-thwarted-major-ukrainian-offensive-2023-06-04/