Title
Futzing and Moseying: Interviews with Professional Data Analysts on Exploration Practices

Journal
IEEE Transactions on Visualization and Computer Graphics, 25(1)

ISSN
1077-2626

Authors
Alspaugh, Sara
Zokaei, Nava
Liu, Andrea
et al.

Publication Date
2018-08-20

DOI
10.1109/tvcg.2018.2865040

Peer reviewed

Futzing and Moseying: Interviews with Professional Data Analysts on Exploration Practices

Sara Alspaugh and Nava Zokaei and Andrea Liu and Cindy Jin and Marti A. Hearst

Abstract—We report the results of interviewing thirty professional data analysts working in a range of industrial, academic, and regulatory environments. This study focuses on participants' descriptions of exploratory activities and tool usage in these activities. Highlights of the findings include: distinctions between exploration as a precursor to more directed analysis versus truly open-ended exploration; confirmation that some analysts see "finding something interesting" as a valid goal of data exploration while others explicitly disavow this goal; conflicting views about the role of intelligent tools in data exploration; and pervasive use of visualization for exploration, but with only a subset using direct manipulation interfaces. These findings provide guidelines for future tool development, as well as a better understanding of the meaning of the term "data exploration" based on the words of practitioners "in the wild."

Index Terms—EDA, exploratory data analysis, interview study, visual analytics tools

1 Introduction

The professional field known variously as data analysis, visual analytics, business intelligence, and more recently, data science, continues to expand year over year. This interdisciplinary field requires its practitioners to acquire diverse technical and mental skills, and be comfortable working with ill-defined goals and uncertain outcomes. It is a challenge for software systems to meet the needs of these analysts, especially when engaged in the exploratory stages of their work.

Simultaneous with increasing interest in this field has been interest in the role of *exploration* within the process of analysis. John Tukey famously described exploratory data analysis (EDA) —"looking at data to see what it seems to say" — in his 1977 book on the subject [23].

To better understand the less structured, more exploratory aspects of data analysis, we conducted and coded interviews with thirty experienced professionals in the field. These participants worked for consulting firms (11/30), large enterprises (8/30), technology startups (6/30), academia (3/30), and regulatory bodies (2/30) and averaged 12.8 years of experience. Our goals were to understand typical exploration scenarios, the most challenging parts of exploration, and how software tools serve or underperform for users today. Among our findings were an augmentation of the stages of data analysis proposed by Kandel et al. [7], and shown in Figure 1. We augment this model by identifying exploratory activity throughout the analysis process (italicized) and propose an additional phase, EXPLORE (bolded), to capture core EDA activities that do not fit cleanly into the other phases.

Although supporting exploratory analysis has been named as the motivation for design work in the literature (e.g., [11, 16, 17, 25]), to the best of our knowledge, this interview study is the first to report data in which users explicitly describe exploratory analysis work.

For the purposes of this study, we consider **exploration** to be open-ended information analysis, which does not require a precisely stated goal (although some EDA practice begins with preliminary, motivated hypotheses). Exploration is opportunistic; actions are driven in reaction to the data, in a bottom-up fashion, often guided by high-level concerns and motivated by knowledge of the domain or problem space. In this characterization, activity ceases to be exploratory when it becomes clear, from the formulation of the goal, what needs to be done to attain a given result; that is, when a top-down plan of action coalesces and can be specified in advance from start to finish, precisely. Example exploratory questions are "what's going on with my users?" or "has anything interesting happened this quarter?"

We acknowledge that by this definition, analysis activity exists along a spectrum from exploratory to directed. Although participants discussed a wide-range of activities, we focus on the exploratory aspects in this paper as these are novel compared to what has been reported previously in the literature. Our methodology and the exploratory analysis spectrum is discussed further in Sections 3 and 4, respectively.

Section 2 describes related work in interviewing data analysts. Sections 4–7 describe the findings, with Section 4 shedding light on how practitioners conduct the exploration process, Section 5 describing challenges to analysis, Section 6 describing current tool usage, and Section 7 describing participants' desires for improvements to software tools. Section 8 discusses the implications of these findings, especially for the design of new tools, and Section 9 draws conclusions.

2 Related Work

Several recent interview studies have shed light on how analysts do their work. Closest to this study is the study of Kandel et al. [7], who in 2012 interviewed enterprise analysts in the context of their larger organization. By contrast, our interview population includes more independent consultants and analysts working at smaller startup companies, and as well as a few analysts in academia and regulatory institutions (see Table 1). Furthermore, Kandel et al's focus was on the overall analysis process and the organizational context, while we are primarily concerned with the exploratory aspects. Kandel et al. organize the data analysis process into five major phases, discussed in more detail below.

Kandogan et al. [8] interviewed 34 business analysts at one major corporation, finding that friction between data, people, and tools was a major impediment to efficient data analysis. They classified analysis as either routine, ad hoc, or self-service, without a focus on exploratory work. In contrast to our study, very few used visualizations and most used Excel. Similarly, Russell [20] presented a case study showing that an advanced visualization was adopted within an organization only after it was repeatedly redesigned and improved, and tightly integrated into the software environment workflow of the intended users.

Recently, Kim et al. [12] interviewed 16 participants in a large organization (Microsoft) and identified five roles that data scientists play within software development teams: Insight Providers, who work with engineers to collect the data needed to inform decisions that managers make; Modeling Specialists, who use their machine learning expertise to build predictive models; Platform Builders, who create data platforms, balancing both engineering and data analysis concerns; Polymaths, who do all data science activities themselves; and Team

- *Sara Alspaugh (alspaugh@berkeley.edu), Nava Zoka (nava.zokaei@berkeley.edu), and Cindy Jin (cjin43@berkeley.edu) were with UC Berkeley when working on this paper.*
- *Andrea Liu (andrealiu@berkeley.edu) and Marti A. Hearst (hearst@berkeley.edu) are currently with UC Berkeley.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

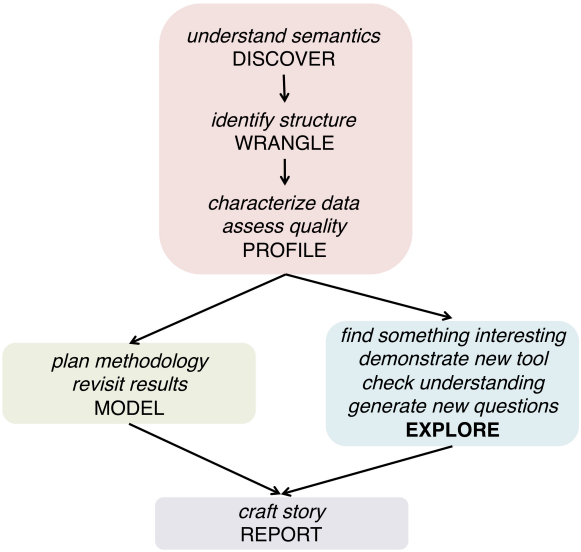

Fig. 1. How exploratory activity fits within the overall analysis process. The capitalized words correspond to the phase of analysis with boldface indicating new terms augmenting those phases beyond those of Kandel et al [7]. Italics introduce types of exploratory activity within those phases.

Leaders, who run teams of data scientists and spread best practices. Our interview study included people of each classification. Fisher et al. [3] also interviewed 16 data analysts within Microsoft who were working with large datasets; their focus was on the special considerations that accompany analysis work on cloud architectures.

Information-intensive tasks such as intelligence analysis, scientific research, and legal discovery are often referred to in the literature as sensemaking [19]. After interviewing intelligence analysts about how they do their work, Pirolli and Card [18] described the process as an information foraging loop consisting of seeking, filtering, reading, and extracting information, and a sensemaking loop consisting of iterative development of a mental model that best fits the information seen to what was known before. Sensemaking is a larger process that often encompasses what we are calling exploratory analysis. Thus several studies of the sensemaking practices of intelligence analysts are related to this work [1, 15, 26].

For example, Kang and Stasko [7] performed a longitudinal field study of intelligence analysts in training, and, aligned with this study, found that the process is akin to basic research and is exploratory by nature. Analysts also wished for better tool integration and data provenance tracking. In contrast to our findings, the intelligence analysts collaborated with others intensively, used many different analysis tools for any given problem, and wanted automated tools to suggest connections and automated document and linking suggestions.

O'Day and Jeffries [14] studied the analysis stage of the sensemaking process of 15 business analysts, classifying 80% of this into six main types: finding trends, making comparisons, aggregation, identifying a critical subset, assessing, and interpreting. The remainder consisted of cross-referencing, summarizing, and finding evocative visualizations.

Creators of innovative visualization tools have long described the use of their tools in case studies. For instance, Inselberg vividly describes how to use Parallel Coordinates as a kind of visual detective [6], and Tweedie and Spence show how discoveries can be made with the Attribute Explorer [24]. Our interview study taps into the current state of practice with real people in the field to inform future development.

3 Methodology

3.1 Participant Recruitment

We solicited interview participants by sending recruiting emails to our professional contacts and mailing lists. The recruitment letter solicited "professionals who analyze data as a *major part* of their daily job to interview about their approach to EDA (spelling out "EDA" as "exploratory data analysis" on first use), including topics such as workflow, tools used, type of data analyzed, and techniques employed." In its "additional background" portion, the letter explained: "EDA is an approach to analyzing data, usually undertaken at the beginning of an analysis, to familiarize oneself with a dataset. Typical goals are to suggest hypotheses, assess assumptions, and support selection of further tools, techniques, and datasets. Despite being a necessary part of any analysis, it remains a nebulous art, that is defined by an attitude and a collection of techniques, rather than a systematic methodology." Participants were compensated $25/hour in gift certificates for their work and interviews lasted on average for 90 minutes.

Potential participants were screened via a background survey to ensure they had at least four years of analysis experience (including graduate school work); from these, 34 participants were interviewed; four were omitted from further analysis because they were found not to do exploratory analysis as part of their work. The final set of 30 is shown in Table 1. The table highlights two separate groups – the first (18/30), in white, are those who programmed as their primary means of exploring data. Those highlighted in grey (12/30) primarily used visual analytics tools with direct manipulation interfaces or a mix of the two modes.

3.2 Interviews

One author conducted the interviews in the fall of 2015. Wherever possible, we interviewed participants in-person at a location of their choosing, but in some cases we conducted interviews remotely via video-conferencing software. We recorded and transcribed each interview for later coding. Most interviews were one-on-one sessions ranging from approximately one to four hours in length, with an average of 1 hour 40 minutes; however, we interviewed p14-p17 in a group of four and p28-p33 in a group of six. All interviews were completed prior to coding.

The interviews were semi-structured, each consisting of open-ended questions about the participant's occupation, data, tools and techniques, workflow, and their opinions on potential future software automation for data exploration. In each interview, we sought to first establish contextual information about the analyst's role in his or her company or research group, and the types of analyses they perform. We let conversation flow naturally rather than follow a script, but made sure that all of our questions were addressed at some point during the interview, with the exception of three interviews that were time-constrained (p25-27). Guided by the list of questions, we sought to learn the following:

- What are typical data exploration scenarios? (§4)
- How does data exploration relate to the other parts of analysts' workflow? (§4)
- What are the tedious parts of data exploration? What are the most challenging parts? (§5)
- What tools and techniques do analysts use to explore data? (§6)
- Do analysts use (interactive) visualizations? If so, how? (§6)
- What automation have analysts developed for themselves to facilitate exploration? (§6)
- Which features do they most appreciate about the tools they use, and which are they lacking? (§7)
- If advanced automation could be harnessed to help analysts explore data, how would they like this to work, ideally? (§7)

3.3 Analysis

Three authors collaborated to iteratively develop a code book based on the interview transcripts, resulting in 75 codes in addition to codes for 82 software tools. (Software tool codes are assigned additional subcodes.) The codes are organized into a hierarchy with the following top-level categories:

- Background (e.g., data characteristics, professional role)
- Workflow stage (e.g., exploration goals, data collection challenges, analysis methods),

ID	Yrs Exp	M/F	Sector	Role	Specialty	Tool Type	Main Software Tool(s)	Exp?
00	4	M	academia	grad student	combustion	prog	Python	EXP
01	5	M	academia	grad student	neuroscience	prog	Python	EXP
02	2	M	regulatory	analyst	finance	prog	R, SAS, Python	ANTI
03	7	M	enterprise	analyst	data science	prog	Python, R	–
05	10	F	startup	analyst	machine learning	prog	R, H20 Flow	–
06	9	F	regulatory	analyst	epidemiology	mix	SAS	–
07	5	M	enterprise (tech)	analyst	data science	mix	R, Tableau, Scala	EXP
08	28	M	academia	professor	computer networking	prog	R, Awk	EXP
09	52	M	enterprise (tech)	analyst	data science	prog	R, Tableau	–
10	6	M	startup	analyst	data science	mix	Python, [employer's]	–
11	4	M	startup	executive	data science	prog	R, Python	–
12	12	M	enterprise (tech)	management	data warehousing	mix	SQL, Pentaho, Excel	–
13	50	M	consulting	analyst	business intelligence	dm	Tableau	–
14	24	F	consulting	analyst	data visualization	dm	Tableau	EXP
15	4	M	enterprise (health) / consulting	analyst	business intelligence	dm	Tableau, SQL, Alteryx	EXP
16	8	M	consulting	analyst	business intelligence	dm	Tableau	EXP
17	16	M	enterprise (finance) / consulting	analyst	business intelligence	dm	Tableau, Alteryx, R	EXP
18	6	M	startup	analyst	data science	mix	Python, Splunk	EXP
19	6	M	enterprise (tech)	analyst	business intelligence	mix	Splunk	EXP
21	2	M	startup	analyst	software engineering	mix	Periscope	EXP
22	3	M	startup	analyst	machine learning	prog	Scala, Python	ANTI
25	8	M	enterprise (tech)	analyst	data science	prog	Scala, Python	–
26	23	M	consulting	executive	data science	prog	Excel, R, Python	EXP
27	16	M	consulting	executive	data science	prog	R, Python	EX/AN
28-33	9.5	M	consulting	analyst	data science	prog	Python, Jupyter, SQL	–

Table 1. Demographics and other information about interviewees. For the final group interview (p28-33), the transcription service did not distinguish speakers, but they tended to be in agreement on most topics as they worked together for the same consulting firm; thus, we attribute them as a group. The number of years of experience for this group given is the median, which is very close to the average of 9.75. For other measures for this group, the most common value is shown. "Tool Type" indicates *main* software interface type: **prog**ramming, **d**irect **m**anipulation, or a **mix** of the two. Participants falling into the latter two groups ("dm" and "mix") are highlighted in grey; these are visual analytics tool users. "Main Software Tools" are those they used primarily. "EXP" refers to if the participant mentioned a case of trying to "find something interesting" in the data,"ANTI" indicates expressing a view explicitly opposed to this practice, and "EX/AN" means both were mentioned.

- Tools used (e.g., Jupyter, Tableau), and subcodes (e.g., pro, con)
- Desired tools and features
- Homegrown automation (e.g., self-created scripts and tools)

For coding, a participant utterance was defined as one turn taken in conversation, for a total of 8683 utterances. Each utterance was coded as a whole, and each utterance could be assigned more than one code.

Coding took place over a period of approximately five months. Three passes were taken over each transcript. Pass 1: two coders independently applied codes to each utterance in the transcript. Pass 2a: one coder reviewed the codes given by her partner to each utterance, and updated her codes with any she felt they had missed or that upon reflection, she agreed with. Pass 2b: the other coder did the same with the first coder's updated codes. This resolved some coding differences that resulted not from a fundamental disagreement but were merely oversights. Pass 3: any remaining difference in codes represents true differences of opinion, so in pass 3, the third coder made a tie-breaking pass. Due to the level of care in developing of the code book and coding method, the average agreement between coders prior to the tie-breaking pass was .91 as measured by Cohen's kappa [13].

In the following sections, we discuss our findings. Throughout the text, the numbers in parentheses describe how many participants *explicitly* supported the point in question; if a participant is not counted it does not necessarily mean they do not agree, but just that they did not mention it explicitly.

4 The Role of Exploration in Analysis

In this section we describe different types of exploratory activity analysts reported. We show how this activity fits within the overall analysis process by augmenting the analysis pipeline identified by Kandel et al [7]. First, we elaborate upon the distinction we made, in this study, between exploratory analysis and directed analysis.

4.1 Exploratory versus Directed Analysis

In the introduction, we set up a contrast between exploratory work and more directed analysis. Though we attempt to draw a boundary to distinguish exploration from other activity, in reality all analysis exists on a spectrum from open-ended, high-level, and opportunistic to goal-driven, low-level, and precise. Moreover, these activities are interleaved and analysts transition frequently from activity at one end of the spectrum to the other and back again throughout their workflow. Nonetheless, since our focus in this investigation is exploration, we attempt to make the distinction within the discussion here and illustrate the difference with some quotes from participants.

In response to a question about what kind of exploratory work he did, one participant replied:

> A lot of putzing, a lot of trying to parse our text logs to see if I could find anything helpful. Yeah... Same as like futzing... Kind of moseying... I don't know, just poking around with things and see what happens. –p18

We include this quote because it has been debated, including by several of our participants, whether analysts actually practice this kind of exploratory, undirected type of analysis, rather than being guided by a clear goal; this quote is one example in favor of the former viewpoint.

Conversely, non-exploratory analysis includes activity like answering a specific question about the value of a certain metric, converting a dataset from one format to another, or training a model to predict a particular outcome. Here an example question might be "how many users have not logged in within the past month?" or "what ad is this user most likely to click on?" An example of directed (non-exploratory) analysis is:

> The project I'm doing, I have a very specific question... how does this area connect to this area during this type of thinking? That way I kind of know what I'm going to do, I'm going to look at this, I'm going to look at the connectivity,

> I'm going to look at how that connectivity changes during different types of thinking, and I'm going to look at how that differs between people. –p01

As discussed above, there is not a clear boundary between exploratory and non-exploratory analysis. An example of the gray area in between is:

> Primarily at its root I do video and voltage versus time, basically. And then correlations and all that sort of stuff to try to extract a meaningful result from those. –p00

In these cases, we relied on human judgment and the coding process described in Section 3.3; this example we coded as both non-exploratory ("video and voltage versus time" is specific) and exploratory ("try to extract a meaningful result" is vague and open-ended).

4.2 Frequency of Exploration Activities

The four core types of exploratory activity in the EXPLORE phase (see Figure 1), in order of the number of participants who described it, are: (1) looking for something interesting, (2) checking understanding, (3) generating new questions, and (4) demonstrating a new tool. We discuss each in turn.

Almost half of participants (12/30) described trying to uncover interesting or surprising results. This is sometimes an explicit goal, often guided by an interest in a range of topics ("implied area of curiosity" –p26) or by a very high-level and open-ended question. One participant describes it in the following way:

> If I work with a client, sometimes they know exactly what they want, and then, for me, it's a question of deciding if that's the right thing for them and if I can sort of guide them in a direction that I think might be more effective for what they're looking for. Sometimes they just have a data set and they want to know what does it tell, what are the interesting stories in it. I tend to, even in that first case, I would tend to explore the data and look for those interesting stories just because I'm curious. –p15

Though the above quote is from a consultant, this activity is also engaged in by some academics. The following participant describes the importance of finding surprising results in his field.

> Even if you take a lot of data and you make a coherent [story], and then you show something people already thought was the case, that's hard to publish. –p08

The next most commonly described activity (9/30) was comparing the data with the current understanding of an underlying phenomenon, in a highly open-ended fashion. This can include troubleshooting, or investigating the source of an issue. One participant confirmed that his exploration includes creating different plots of the data and trying to see if he can explain everything he is visualizing to determine what he needs to investigate further:

> What you are looking for... when you say, "Look, Tableau, just show me these seven fields, just show me a chart that lets me understand." If you look at it, most of the time, you are going to go, "Yeah, that is what I figured. The more I sell, the higher my profit is and my bigger customers are driving the majority of the profit." What you are looking for is those outliers... Is there fraud involved? –p13

In the next category, some participants (5/30) described coming up with new analysis questions or hypotheses by looking through the data in a free-form fashion and letting ideas spontaneously arise e.g., brainstorming.

> There's learnings that I'm having in the data that are leading more questions but aren't necessarily germane to the task at hand. Or I see this data element, that I was in a meeting with somebody last week, and like, oh yeah, we want to find that out at some point. So there's just kind of tagging it for later. –p15

Lastly, some participants (5/30) described engaging in exploration to test or demonstrate a new analysis methodology or tool, specifically to show what insights it can reveal. This is related to the above case, in that often participants are looking for something interesting to showcase the methodology or tool they developed.

> If [the potential customer is] not sure if [our tool] is going to be useful for them, or they're not quite sure how to make use of it, they have access to it but they're like, "I don't really know what to do," there's a team... who will use [our tool] to generate and annotate a visualization... They'll do some analysis and come up with some interesting findings, they'll be like, "Hey, we found this vulnerability that just came out in tools that you use. This is the kind of thing that you could discover using [our tool]." –p10

4.3 Opposition to Exploratory Analysis

Several participants (4/30) cited reasons that they do not or try not to explore data. These include: desire to avoid common exploratory analysis pitfalls like finding spurious correlations, working in environments in which data collection is costly enough that analyses must be planned in advance rather than ad hoc, and concern about wasting time due to not being guaranteed to find interesting things.

Two mentioned their desire to avoid "fishing expeditions", multiple comparisons, and spurious correlations. Some believe that it is better to come up with questions that are more precise and measurable.

> I'm so terrified of it that I make sure that like I early on steer the objective of the, I steer the objective towards making a much more comfortable prediction problem. Like I'm kind of forthright about like, you know if you want me to go hunting, I'm going to come back with something. If you want I can do that thing where I just like lay sufficiently many n-squared things correlate them with each other until I find a perfect correlation for you kind of thing. I can draw whatever picture that you want me to draw. –p22

Three analysts (p01, p02, p27) disputed that it made sense to ever play with the data without being driven by a specific question because in their fields, datasets can be expensive and difficult to collect, so it only makes sense to do so after carefully defining an analysis plan, rather than engaging in ad hoc exploration.

> Because it takes a lot of work to pull a data set together. No one just hands it to you. –p02

In these cases, it also makes sense to invest in shared pre-processing pipelines, as noted by p01. However p27 later admitted that clients often came to him without specific questions, so he did engage in ad hoc exploration in practice despite his reservations.

Three analysts were not interested in exploring data to find surprising results because of the risk of not finding anything interesting. Relatedly, one participant mentioned that some people want to avoid exploration because they think of it as pure overhead that they should minimize, because the value of exploration can't be easily quantified:

> You're not making incremental progress to a goal because, as we've discussed, the goal may be undefined or just extremely loosely defined. –p26

4.4 Exploratory Activity in Other Workflow Phases

After putting all of the activities we labeled as exploration into activity subgroups, we found that many activities that participants considered exploratory could be considered part of other phases of analysis; these are shown in their corresponding phases in italics in Figure 1.

The majority of participants (21/30) described exploring to try to understand the semantics of the data and what it represents, as well as the data generation logic. We classify this activity as part of the DISCOVER phase.

The majority of participants (19/30) mentioned exploring to identify how to ingest the data and restructure it, which data to pull from a

separate system if necessary, and how to integrate the different data sources. This is the beginning of the WRANGLE phase. Since this is well-described in the literature we do not belabor it here.

The same number of participants (19/30) described as exploration the act of characterizing data distributions, trends, correlations, etc., usually using standard charts. This is sometimes called descriptive analysis or profiling, so we placed it in the PROFILE phase. Since this is also covered in the literature, we omit further description.

Most participants (25/30) considered exploration to include the act of examining data correctness and concurrence with assumptions. This is sometimes called data cleaning, but following Kandel et al., we place it also in the PROFILE phase.

One activity participants (16/30) described that often seemed borderline between exploratory and non-exploratory analysis is planning their analysis approach. Planning includes figuring out how to accomplish a high-level goal by understanding the customer, scoping the problem in the context of the data, and then deciding upon the analysis plan to pursue. This also includes defining new metrics for later monitoring and evaluation; for example:

> The open-ended question really came up first three months ago in first defining what these metrics should be. It's based on how I define things like script creation, or what sort of content creation metric that we're tracking. –p11

This activity is part of the MODEL phase.

The final category of exploratory activity that a few participants (3/30) described was crafting a story. This involves exploring different approaches to identify the best way to present the data for downstream consumers and relate the data to what they care about, or to operationalize the analysis (e.g., build a proof-of-concept). This is part of the REPORT phase.

5 Challenges

The previous section defined exploratory analysis in the eyes of the study participants; this section relates the aspects of exploratory analysis that they find challenging.

5.1 Challenges in the EXPLORE Phase

The most common complaint (9/30) was that exploratory analysis is too time-consuming and hands-on, consisting of largely overhead. Many analysts thought that data profiling in particular should be more automatic.

As mentioned in Section 4.3, several analysts (6/30) mentioned that sometimes an exploratory analysis yields nothing of value or nothing interesting, but it is necessary to present it anyway; the challenge is how to do this. One Tableau expert explained:

> I think often times there is no interest or value but they still need a dashboard because they need something to turn into an executive. Then you just make it as pretty as you possibly can. Put a bunch of lipstick on it. –p14

Next analysts (4/30) said a common challenge is analyzing data for customers or clients who do not know what they want to ask about the data, or who have vaguely defined interests, since there is nothing to constrain or guide the potential space of exploratory work.

> This still happens with surprising regularity, we get clients who are just like can you tell me what's interesting in this data and how I can make a bazillion dollars in it because I've read this article in Forbes and it says that there's gold in these there hills, and all I need to do is take my data and exhaust it, I can turn it into money. –p26

Lastly, some analysts (3/30) were concerned about their exploration process resulting in a biased outcome if they always focused on the same things or started in the same way, especially when the data set is very large and can't all be examined. Here the challenge is developing a strategy to ensure good coverage, while avoiding the problems mentioned in Section 4.3.

5.2 Challenges in Other Workflow Phases

Over half of the analysts (18/30) said one of their biggest challenges was lack of documentation, metadata, and provenance. Many complained that understanding the data often required trying to talk to people knowledgeable about the data in person, but that often these people were difficult or impossible to access. Similarly, many analysts (18/30) described one of their main challenges as being figuring out what structure the data is in. Analysts specifically cited challenges with unstructured data, highly nested data structures, legacy data structures not adapted to modern conventions, data integration, and entity resolution. Unlike Kandel et al., many analysts we talked with lacked support from a data warehousing or IT team. A common theme (18/30) was the challenge of dealing with different "versions" of the data, stemming from changes, either planned or accidental, in the data generation and collection process. The majority of the time, those generating or collecting the data do not notify analysts of such changes in advance.

Analysts (18/30) mentioned data quantity as one challenge; specifically the difficulty of manually examining large numbers of columns, tables, or data sources, in order to figure out what they all mean. Another challenge was waiting for the analysis environment to execute even simple operations over large datasets, resulting in context-switching costs when execution completes. Some analysts worked in environments where even reasonable-sized slices of the data (e.g., by user) are too big to fit on a single laptop, making it difficult to profile the data; this makes thoughtfully subsampling the data a challenge. Lastly, several analysts (4/30) mentioned the challenge of visualizing large datasets, because even when it is computationally fast to do so, the resulting visualization can be too cluttered to make sense of.

A common complaint (18/30) was the difficulty of tracking the exploration process, including documenting the purpose of various pieces of code or other artifacts, and the provenance of results. Analysts faced this difficulty both from the perspective of capturing, versioning, and documenting their own work, and from the perspective as consumers of others' work, or their own work from a past time. Many analysts described struggles with trying to understand what they had done in the past when looking at their old scripts or figuring out what another analyst had done when taking over someone else's project. In many cases, even finding the relevant work is a big challenge.

> I have many dashboards like this for the different things I've launched and it constantly happens that I forget about dashboards that I've made or I forget which ones are like the correct ones, or are supposed to be interesting. –p21

Another commonly mentioned challenge (17/30) was coping with poor quality or "dirty" data. It is particularly challenging for analysts such as consultants to diagnose such problems when the data comes from a domain that they are not experts in.

Though it was not a focus of our interviews, most analysts seemed to work mostly independently. Nonetheless, several analysts (12/30) described challenges in collaboration and coordination or lack thereof. Some mentioned the problem of different analysts coming to wildly different conclusions from the same data when given an open-ended question, due causes such as defining metrics based on the low-level data in different ways.

Analysts (10/30) had a variety of difficulties around visualization, such as lack of interactivity, difficulty in working with certain types of data such as dates in financial settings or 3D data, the challenge of choosing the right chart type, and the time-consuming nature of formatting plots especially because of bad defaults.

Analysts (8/30), who primarily worked on prediction problems, described several machine learning-specific challenges, namely around ensuring that model features meet certain requirements, such as not occurring later than the event they wish to predict (i.e., data leakage), or not having too many categories when their chosen algorithm does not work well in such situations.

6 Pros and Cons of Existing Tools

We summarize the software analysis tools that participants discussed. Although our focus is exploration, discussions of software's strengths

and weaknesses tended to encompass all aspects of the analysis process. The first section discusses how participants use visualization and interaction in general; this is not specific to any subset of tools but spans all of them. Where relevant, we note how many participants for a given topic came from each population of participants (visual analytics users versus programmers) highlighted in Table 1.

Some pros for tools in general include: being open-source, being well-supported by the participant's organization, and having staying power in the marketplace (i.e., being not likely to be made obsolete soon). Reasons participants (10/30) preferred open-source software include transparency (being able to read the source code to understand why something is happening, and change it if needed), security (having complete control over the code and how it is installed), and legal considerations (open-source licences help companies avoid infringing contract agreements and intellectual property laws).

6.1 Visualization

Almost all of the participants (28/30) used visualization to explore data; the exceptions were p05 who explicitly said she did not, and p03 said he only uses histograms a little bit for exploration and mostly uses visualization for presenting data to end consumers.[1] Both p03 and p05 are predominantly programmers rather than visual analytics users.

The types of visualizations mentioned were basic charts like time-series, bar charts, and scatter plots, with some additional visualization types mentioned by participants with special kinds of data requiring domain specific tools, such as fMRI data (p01). Four of the participants who are Tableau experts described using more advanced visualization types, like Sankey diagrams.

Nine participants explicitly said they use interactive visualizations as opposed to static visualizations and seven explicitly said they do not. Of those who explicitly said they do, five are primarily direct manipulation users as shown in Table 1, while one was primarily a programmer (p11), and the other three were a mix (p06, p07, p12). Of those who said they do not use interactive visualizations, five were primarily programmers and used tools that do not enable visualization, while the other two used visual analytics tools that combine direct manipulation and textual input methods. Three acknowledged lack of support explicitly as the reason they do not use interaction (p00, p08, p09). One participant described making up for lack of interaction by frequently re-plotting instead; for him, interaction was not a major necessity and he would rather use his current toolset than move to one that enabled better interactivity. One said interactive visualization was the primary feature missing from his toolset:

> I really like interacting graphics, so I wish RStudio had more interacting stuff. It's coming, RStudio are doing some interesting things. . . Painting, brushing, so if you have multiple class display you see some points on here. You brush those, they turn color. . . –p09

Interestingly, this same participant said he did use Tableau to create interactive dashboards for executives, but he preferred R for his exploration and analysis work. The six data science consultants interviewed as a group had a long discussion about their desire for better interactivity in Jupyter notebooks; whether or not they use interaction currently was unclear. Only one participant (p21) who said they didn't use interactive visualization frequently used a tool that provides that capability.

6.2 Programming Languages and Command Line

Nearly all of the participants (28/30) use programming languages or command line tools to some extent, though for 18 it is their dominant tool set, while four use them infrequently (see Table 1). The two remaining participants, visual analytics users (p12, p13), used scripting tools in the past but, no longer do. Participants cited using command line tools and scripting languages, like Bash and Perl, as glue code, as both a benefit and a drawback: they are useful for pipelining other tools, but such pipelines can be hard to keep organized and well-documented, making them susceptible to hard-to-diagnose errors and difficult to maintain, modify, or share with others. The standard tradeoffs between programming languages were mentioned (e.g., typed vs non-typed enforcement languages, for example, when comparing Python to Scala).

[1] We were not able to ask p26 about visualization due to time constraints.

6.3 Homegrown Automation

Many participants (19/30) described tools they had created for themselves to perform repetitive tasks; we call this "homegrown automation". This includes scripts or wrappers for commonly needed functionality, especially for visualization, but also for tasks in the WRANGLE and PROFILE phases:

> I would just use functions that I have written that automate all those things. . . I've written my own set of tools that look for how many missing data, what kind of percentage of these columns are categorical? If they're categorical, are there a lot of categories? Are there are a few? Things like that are pretty important. –p05

One participant had written their own visualization library. A few participants (3/30) wrote code to profile all of the columns of a dataset (e.g., create various graphs of each column), which they would then look through individually; one participant was wary of doing this as he thought it was better to be more selective at what he looks. Two participants described automating the parameter space search in their analysis (e.g., choice of regularization parameter in a machine learning model). A few (3/30) described trying to script their entire analysis pipeline, in part as a form of documentation of how the pipeline should be executed, and to prevent problems in the future in trying to remember how to rerun their work. Two participants wrote code to help the less technical people in the organization do their analysis and generate reports. Those who were primarily Tableau users did not have many such tools to automate common or repetitive tasks, citing the lack of support of macros or templates as one reason, but also leaving the impression of not really needing them. One participant mentioned macros in Alteryx, another graphical tool often used with Tableau. Several participants (6/30) mentioned a barrier to homegrown automation: the difficulty of generalizing any given solution enough so that it can be used in other situations or by other people. Many participants (12/30) explicitly described a strategy we call "copy-paste reuse":

> I ended up having the same exact Python script with like very minor variations. But instead of having it with beautiful git commit history or anything like that, I just replicated the file approximately like 25 times with just timestamps embedded in the file name so that I would know when I'd made it because I was able to match that to what I was doing. –p21

Other barriers to automation include: things that require a lot of human judgment even when they are tedious or repetitive (1/30), and not wanting to introduce a bias into analysis by always approaching data in the same way (1/30). A couple participants (2/30) described some minor regrets at taking the time to write code to automate something only to find later that there already exists a tool that already serves that purpose that they could have used instead.

6.4 Database-Related Tools and Spreadsheets

The second most-used type of tools were databases and related tools, used by at least 18 participants. There were some complaints about SQL, including the difficulty of writing complex queries (9/30) and the difficulty of dealing with different versions of SQL (1/30). A couple participants expressed finding it straightforward to use and appreciated that SQL queries are transparent and modular in certain cases compared to direct manipulation tools. Many participants complained about poorly documented databases and the difficulty of taking stock of all the available datasets within an organization (see Section 5). Some were aware of tools that try to address these problems (like DbVisualizer and Alation). Most participants seemed to interact with databases directly through SQL on the command line, with the clear exception of p12, who used the Pentaho suite of tools extensively.

Many participants (18/30) mentioned using spreadsheets, primarily on an occasional basis for quick, straightforward visualizations of small tabular datasets, to make small edits to datasets, or to exchange datasets. One drawback of storing data in spreadsheets, which some participants mentioned, is that there is no means to enforce restrictions on the structure of tables in spreadsheets, and they shouldn't be used as a replacement for a database.

6.5 Visual Data Analytics and BI Tools

Almost half of participants (14/30) reported using visual analytics tools, with twelve using them as their primary analysis tool, and three others mentioning them in their interview. This category includes Tableau, SAS, Splunk, Stata, Alteryx, Periscope, and others. As with programming languages, these tools target different use cases. But compared to programming languages, their interfaces and supported use cases can vary more from one to another, and their supported use cases are often more narrow.

Participants appreciated the speed and ease of visualization creation in these tools. Some felt that direct manipulation is superior for exploratory work. A few participants commented that these tools (and spreadsheets) are good for people who are unable to program. One participant put it this way — in this case in reference to Trifacta, though the sentiment is similar to those expressed by others about other direct manipulation tools:

> Is it useful? I think the tool is super useful. Is it useful for me? Not necessarily. I do like to have my entire script, or multiple scripts, to describe my whole process from end to end, I don't want to automate stuff in between; I want to have my own code. So for me it may not be as useful, but for some businesses or someone who doesn't code much or doesn't like to code, I think that could be really useful. –p27

Additional drawbacks named include: when tools are a black box and do not reveal clearly how they are manipulating the data, when they do not support easy data or work export, and when they become slow and bloated due to adding too much functionality. One participant expressed the opinion that being unable to program was a handicap for analysis, and another that the users of such tools were, in his experience, not intelligent.

The participants who use visual analytics tools are a subset of those who use visualization to explore data (i.e., everyone who reported using visual analytics tools confirmed using visualization for exploration). Two (p06, p21) who use visual analytics tools as their primary tool for exploring data said they do not use interactive visualization for exploration because they don't need it. One participant (p09) who reports using a visual analytic tool, but not using interactive visualizations, uses Tableau only for presentation, and R primarily for exploration.

6.6 Notebooks and Markup

Just under half of the participants (13/30) mentioned using Jupyter notebook or similar approaches like R Markdown in RStudio. All but one of these 13 participants (p07) used programming as their primary analysis tool. One benefit mentioned is notebooks make it easier to document the analysis process, which was a challenge discussed in Section 5. A drawback for some participants is the lack of support (at the time of the interview) for multi-user deployments. Other criticisms of notebooks included that it is hard to create rich interactive visualizations in them, they cause scripts to execute more slowly, they are bad for engineering analysis pipelines, and they are hard to extend.

6.7 Miscellaneous

Seven participants mentioned the data cleaning and wrangling tool Trifacta in favorable terms, but only one said they actually used it. Three participants use cluster computing frameworks for doing analysis of large data sets. Two participants described exploring using tools for capturing and analyzing web events. One participant described using a data catalogue tool, such as DbVisualizer or Alation, while seven more mentioned encountering them in the past; such tools could help with the data discovery challenge described in Section 5.

7 Desired Data Exploration Tools and Features

In this section, we describe participants' ideas for hypothetical tools and features that they would like to have, that are missing from their current tool sets.

7.1 A Desire for Tool Integration

Many participants (12/30) expressed dissatisfaction with the fragmented analysis tool space. Participants want to have better integrated tools in a consolidated work environment, or to have one tool that does everything they need, so they don't have to be "swivel chairing across twenty different tools" (p28), since switching tools burdens one's focus. (Intelligence analysts reported a similar issue to Kang and Stasko [10].) Thus, to improve adoption, new tools should integrate well with existing workflows and toolsets, with one participant calling for increased partnerships and integrations across companies to improve compatibility. Relatedly, consultants need to be able to work in client environments; therefore, lightweight and/or open-source tools would lead to better adoption in these situations. One participant specifically called out the need to be able to call APIs from within their applications rather than having to write data out to the command line to input into external tools. The downside, pointed out by p13 and p28, however in making a tool extensible, is causing it to become bloated, making it slower and more difficult to use.

Some participants (5/30) described using a certain product only because there was good support for it in their organization. The fact that this overrode many other serious issues the participants had with it suggests that one desirable product feature is having it be strongly supported by the user's organization, and anything that product designers can do to enable better support tools will help with that [20].

7.2 Trade-offs Between Direct Manipulation and Coding

Frequently mentioned (8/30) was the trade-off between the increased control and expressiveness of programming languages, like R and Python, versus the efficiency and ease-of-use of direct manipulation tools, like Tableau. This is a well-known and long-standing usability issue for technical users [21]. For instance, Takayama and Kandogan [22] conducted a survey of system administrators that found most considered command language interfaces to be more reliable, robust, accurate, trustworthy, and faster when compared to GUIs, although they were split about which kind of tool was easier to use. In these cases it may make sense to switch tools, but switching also imposes a burden, as discussed in Section 5. Thus, participants tend to prefer a direct manipulation tool only when they know how to do the majority of their work in it.

As a compromise between these trade-offs, several participants expressed the desire to be able "open the hood" of direct manipulation tools so they can write the code themselves if needed. Interestingly, both participants who were primarily programmers and participants who were primarily visual analytics users suggested ideas similar to this, though from different starting points. Participants p28-p33 suggested a combined code editor and GUI, such as a Jupyter notebook where every cell also has a Tableau-like GUI for working with visualizations. The hope for such a tool would be to combine the flexibility and expressiveness of programming languages for transforming data, with the conciseness and ease-of-use of GUIs for creating visualizations.

> In terms of just what I'd be interested in seeing come down the line, having something that I can bolt onto say the Jupyter notebook is much more interesting to me than having like, a Tableau-type thing where I can't do everything I need to do. –p28

Of course, programming languages are not a panacea, and require much more work to accomplish tasks that can be done with a few swipes in a GUI.

> I prefer to do it in Tableau because Tableau can be so much faster at that. Drag something and then, boom, see it there. –p15

Other participants discussed how GUIs are a better fit for exploration in particular:

> p17: We've got really two kind of approaches to dealing with questions, either we can start with our pre-defined question and then refine from there. And what that means is that we'll sometimes miss the best possible solution because we're narrowing in on our bias. Versus if we start off with an exploratory in mind, we will try a whole bunch of solutions, we'll get a whole bunch of fails. That whole theory about failing fast. I want to fail fast and fail often...
>
> p16: I'd say this is scripting versus visualization.
>
> p17: Yeah. With Alteryx, Alteryx is a refinement. Alteryx is: I already know what I want to do, and let me refine there and get there step by step and get closer. Tableau is very open-ended, it is more of that direct manipulation of I want to put this over here and then I want to do this to it...Tableau is the most direct manipulation tool I've used, but it doesn't go far enough for me. With Tableau, I've still got this pill that I'm putting on a shelf...

One participant (p27) desired a tool to help him translate his code from one language to another, which would also help him learn new languages; an example of a tool that does this is mpld3, which translates Python matplotlib plotting code to JavaScript D3 plotting code.

7.3 Automatic Wrangling, Profiling, and Cleaning

Some participants (5/30) expressed interest in tools that would perform operations typically considered part of data wrangling, such as detecting dataset structure to optionally convert it into a normalized or flexible input format, and inferring the relationship between two different datasets (including semantically, even when there is not an exact syntactic match) to automatically integrate the datasets. One drawback that some participants (3/30) pointed out is that the process of manual data wrangling is valuable for understanding the dataset; this concern applies equally to automatic data cleaning tools.

Some participants (7/30) suggested tools for automatically profiling data that would automatically calculate certain statistics and create plots for every field or relevant field combination in the dataset. Some participants (7/30) pointed out tools that already exist for this, though not all of those who were aware of such tools used them, citing reasons like lack of integration with their current toolset, or their preference to write code to do it themselves. Several participants (3/30) had written their own scripts and packages for automatically profiling datasets. One participant described using automatic profiling functionality in Splunk that generates graphs and other summaries of input datasets.

Conversely, many participants (9/30) also expressed skepticism towards the utility of such tools, citing a distrust of "code generation" (p28) tools and a preference to write their own code so they have complete control; this concern pertains to all of the automated tools discussed in this section. Several participants (7/30) expressed the opinion, also general to many of the tools suggested here, that "the parts that are easy are easy, and the parts that are hard are the parts that would be difficult to automate" (p28); in other words, that in practice these tools would not be able to address the true pain points of this work.

Some participants (6/30) expressed interest in tools that could automatically clean and validate the data. One participant worked in a field that had already developed a pipeline for automatically cleaning their fMRI datasets, which are expensive to collect, and thus worth the investment. Four participants in total expressed skepticism, including for reasons described above. Two thought automating cleaning and profiling would be detrimental to their understanding and work.

> I think the problem is that, if you want to be able to understand the data best, you kind of have to know how messy it is at the rawest form...being able to do the dirty work yourself is still pretty valuable for data scientists, even if it is more time consuming. –p07

Some (4/30) thought the idea of raising flags on suspicious attributes of the data, rather than automatically removing erroneous information, was a more realistic approach.

7.4 Automatically Generated Visualizations and Insights

One analyst thought automatically recommending visualizations would not be useful unless it was done at the very beginning of the analysis, because he decides what visualization to make at the beginning, and this guides all his later analysis choices. A related idea is automatically recommending which subsets of data are relevant to a given analysis, though several analysts expressed skepticism as to the feasibility of this in many contexts. A handful of analysts (3/30) expressed interest in tools that would help find interesting relationships in the data, either through guided exploration or recommendations of some sort. One participant points out a drawback of this:

> As I was mentioning that, I was thinking, boy, what a two-edged sword such a tool could be...The fact that if you get there too quickly, you actually don't understand at all. That notion of, hey, algorithm, find me cool stuff, could be hugely abused in this regard, which is, yes, you find cool stuff, but you don't really know what it means. –p08

One participant thought that any sort of artificial intelligence in analysis systems would necessarily be much less powerful (yet more expensive) than humans for the foreseeable future, and thus limited in usefulness, due to lacking in context and intuition.

Relatedly, a number of participants (12/30) agreed that in order for recommenders to be useful, they must correctly navigate the trade-off, discussed above, between lack-of-control (from trying to automate too much) and tedium (from automating too little and leaving lots of repetitive work to the user). Otherwise, they argued recommenders or code generation tools force the user to work in a roundabout fashion to manipulate the tool to accomplish their goal.

7.5 Analysis Provenance

Several (4/30) participants expressed a need for better capturing and versioning of the analysis process, including linkage between pieces of code and the artifacts they generate, so that the provenance of particular analysis results can be recovered. At least one participant described trying to encode and document their entire pipeline so that it could be reproduced from start to finish with ease. The top request of another was the ability to highlight regions of their visualizations and recover the provenance of all data points in that region, including the attributes of the input and intermediate data in the pipeline.

8 Discussion

8.1 Clarifying the Role of Exploratory Analysis

In our study, we sought to clarify the role of exploration in the analysis process. We found that part of the reason the definition of exploration is so difficult to pin down is that not only is it an open-ended process, but also exploratory activities pervade the entire analysis process, at least for some analysts. We propose the addition of a new phase to Kandel et al's workflow model, EXPLORE, to encapsulate activities that don't fit well in the other phases. Analysis in EXPLORE is distinguished from that in MODEL by its more open-ended rather than goal-driven nature. The definition of the EXPLORE phase helps to describe what constitutes EDA in practice.

Some authors write about EDA as getting an overview of the data [2], and others write about doing detective work to find interesting phenomena in the data [6]. Our study shows that practitioners in the field really do "ask questions of their data" in an exploratory fashion. Within the EXPLORE phase, we categorize four main types of activity that have not been heavily documented to date: **find something interesting**, **check understanding** of existing phenomenon in an open-ended fashion, **generate new questions** or hypotheses, and **demonstrate new tools** or methods to prove they are useful for analysis.

Participants described their exploration-related challenges; these include: justifying the **overhead** of ad hoc orienteering work whose

results are hard to quantify, what to do when a (potentially lengthy) exploratory investigation yields **no interesting results**, coming up with a plan of action without a **clearly formulated problem** to address, and choosing an exploration **strategy** that avoids pitfalls like spurious correlations.

8.2 Discussing Tools for Data Exploration

We investigated the pros, cons, and desiderata of the tools participants use for exploration.

The use of programming or scripting languages was widespread (28/30) even among those who primarily use direct manipulation tools, speaking to the pervasiveness of programming as a modus operandi for some exploratory tasks. Databases and spreadsheets were also widely used, but there were more complaints about lack of documentation of data stored using these tools than about their usability per se.

Notebooks were popular among those who used them and several participants proposed that they allow for integrated interactive visualizations. In the time since the interviews were conducted, the popularity of Jupyter notebooks and similar tools have continued to increase for data analysis, and the notebooks themselves have begun to include better support for visualization and widgets to support interactivity. Competing tools that integrate visualization into notebook-style data analysis have also emerged, including interactive widgets for Jupyter notebooks, Zeppelin, which allows for language independent visualization including SQL queries, Observable, which provides a notebook-style interaction for the popular d3 JavaScript visualization language, and a web-based version of the Wolfram language.[2]

Visual analytics tools were nearly as widespread in use as programming and scripting languages, but not as the predominant tool set; participants appreciated the speed and ease with which they could create visualizations in these tools but disliked that they were black boxes. Though visualization was widely used for exploration, it was surprisingly not universally used, with two participants explicitly saying they did not use it. Interactive visualizations were less widely used, with nine participants explicitly mentioning using them and seven explicitly not. Participants who use visual analytics tools like Tableau as their primary tool for exploration tended to use more advanced and interactive visualizations. The most commonly given reason for not using interactive visualizations (when one was provided) was that the participant's favored tool set did not support them.

Participants noted the trade-offs between programming and direct manipulation and desired a compromise solution that combines the two interfaces within one tool environment, so that they do not have to make these trade-offs. In general, participants wanted integrated tool sets so they didnt have to switch environments for different tasks. They also value tools having institutional support.

Despite being of great research interest, at the time of our interviews, Trifacta and various cluster computing tools were not widely used among our participants. We were particularly interested in "homegrown automation"—scripts, macros, templates, and other shortcuts participants created for themselves to save time and effort. Many participants used these, which points to a potential opportunity for tools to increase automation or ways to create reusable modules to encapsulate common workflows. However, there are barriers to doing this, such as the difficulty of generalizing code to make it reusable, which needs further research to overcome.

Lastly, due to widespread research interest in this topic, we tried to gauge interest in tools for automating exploration and encountered less interest and more skepticism than anticipated. This differs from the results of Kang and Stasko [10] who found that intelligence analysts have great interest in automated analysis tools. Many participants discussed the challenge of lacking analysis provenance but comparatively few suggested tools for improved provenance capture.

8.3 Recommendations for Tool Development

Several key recommendations for future tool development emerge from this study:

1. Combine direct manipulation with command line tools. The emerging trend toward embedding interactive visualizations within notebook-style programming tools suggests that the field is moving in this direction.

2. Make it easier to create reusable modules to encapsulate common workflows in analysis tools. The prevalence of "homegrown automation", such as small scripts, templates, and macros for personal reuse, indicates that this would be useful.

3. Work on integrating with other popular tools, and on ways to make it easier to switch to and from one tool to another, such as by supporting easy import and export functionality.

4. Continue to research and develop tools for recording history and provenance of both analysis and data. Although this has long been an area of research, and innovative solutions such as HARVEST [4], Ground [5], and LabBook [9] are being developed, many participants cited this as an ongoing problem, so solutions have yet to percolate into general practice for reasons that should be investigated.

5. Consider business concerns and the relation of tools to the entire product ecosystem just as carefully as the design of the core functionality and interface of the product. In many cases, participants were interested in functionality offered by research and products, but practical impediments (lack of integration, worries about lack of control, or need to write code for the missing functionality) prevented adoption.

6. When creating tools to automate exploratory tasks, be careful to avoid making the user feel a lack of control or visibility into what the tool is doing, as this was a concern some participants cited as a reason they were not interested in such tools.

8.4 Limitations

One limitation of our work, common to many interview studies, is that our sample of participants is not drawn randomly from the overall analyst population. Furthermore, because we recruited participants by reaching out across our professional networks, this introduces a bias in the overall subset of analysts we reached. For example, the vast majority of our analysts were based in the San Francisco Bay Area, all are English-speaking, and analysts in the tech industry are likely overrepresented. Our participant sample also shows a gender imbalance that may or may not be reflective of the underlying analyst population; the same may be said for the types of tools analysts use, as our sample skews slightly towards programmers.

9 Conclusions and Future Work

In this work, we interviewed thirty professional data analysts with the goals of clarifying the role of exploration in analysis, identifying common challenges, and understanding how analysts use software tools for exploration, with a focus on what tools and features are needed. We found evidence of four main types of exploratory activity that is underrepresented in the literature, as well as evidence of exploratory activity in all other phases of the analysis process. We then described the challenges analysts face and benefits and shortcomings of tools analysts use for exploration, along with the ideas that analysts had about tools and features they would like to have. Given the community's demonstrated interest in creating tools, particularly "intelligent" ones, for data exploration, we hope this information will be provide useful evidence regarding the data exploration experiences of practitioners and will help inspire new research directions.

Acknowledgments

We thank our interviewees. We thank the anonymous reviewers and Melanie Tory for giving valuable feedback. This work supported in part by a gift from Tableau and by supporters of the UC Berkeley AMPLab (https://amplab.cs.berkeley.edu/amp-sponsors/).

[2] jupyter.org/widgets, zeppelin.apache.org/, beta.observablehq.com/, www.wolfram.com/language/elementary-introduction/2nd-ed/

REFERENCES

[1] P. Cowley, L. Nowell, and J. Scholtz. Glass box: An instrumented infrastructure for supporting human interaction with information. In *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*, pp. 296c–296c. IEEE, 2005.

[2] S. Few. Exploratory vistas: Ways to become acquainted with a data set for the first time. *Visual Business Intelligence Newsletter*, Jul/Aug/Sep 2011.

[3] D. Fisher, R. DeLine, M. Czerwinski, and S. Drucker. Interactions with big data analytics. *Interactions May+June*, 19(3), 2012.

[4] D. Gotz and M. Zhou. Characterizing users' visual analytic activity for insight provenance. *IEEE Information Visualization Conference (InfoVis)*, 2009.

[5] J. M. Hellerstein, V. Sreekanti, J. E. Gonzalez, J. Dalton, A. Dey, S. Nag, K. Ramachandran, S. Arora, A. Bhattacharyya, S. Das, et al. Ground: A data context service. In *CIDR*, 2017.

[6] A. Inselberg. Multidimensional detective. In *Information Visualization, 1997. Proceedings., IEEE Symposium on*, pp. 100–107. IEEE, 1997.

[7] S. Kandel, A. Paepcke, J. M. Hellerstein, and J. Heer. Enterprise data analysis and visualization: An interview study. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2917–2926, 2012.

[8] E. Kandogan, A. Balakrishnan, E. M. Haber, and J. S. Pierce. From data to insight: work practices of analysts in the enterprise. *IEEE computer graphics and applications*, 34(5):42–50, 2014.

[9] E. Kandogan, M. Roth, P. Schwarz, J. Hui, I. Terrizzano, C. Christodoulakis, and R. J. Miller. Labbook: Metadata-driven social collaborative data analysis. In *Big Data (Big Data), 2015 IEEE International Conference on*, pp. 431–440. IEEE, 2015.

[10] Y.-a. Kang and J. Stasko. Characterizing the intelligence analysis process: Informing visual analytics design through a longitudinal field study. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, pp. 21–30. IEEE, 2011.

[11] A. Key, B. Howe, D. Perry, and C. Aragon. Vizdeck: self-organizing dashboards for visual analytics. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pp. 681–684. ACM, 2012.

[12] M. Kim, T. Zimmermann, R. DeLine, and A. Begel. The emerging role of data scientists on software development teams. In *Proceedings of the 38th International Conference on Software Engineering*, pp. 96–107. ACM, 2016.

[13] M. L. McHugh. Interrater reliability: the kappa statistic. *Biochemia medica: Biochemia medica*, 22(3):276–282, 2012.

[14] V. L. O'Day and R. Jeffries. Orienteering in an information landscape: how information seekers get from here to there. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pp. 438–445. ACM, 1993.

[15] E. S. Patterson, E. M. Roth, and D. D. Woods. Predicting vulnerabilities in computer-supported inferential analysis under data overload. *Cognition, Technology & Work*, 3(4):224–237, 2001.

[16] A. Perer and B. Shneiderman. Balancing systematic and flexible exploration of social networks. *IEEE transactions on visualization and computer graphics*, 12(5):693–700, 2006.

[17] A. Perer and B. Shneiderman. Integrating statistics and visualization: case studies of gaining clarity during exploratory data analysis. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 265–274. ACM, 2008.

[18] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, vol. 5, pp. 2–4, 2005.

[19] D. Russell, M. Stefik, P. Pirolli, and S. Card. The cost structure of sensemaking. In *ACM Conference on Human Factors in Computing Systems (CHI)*, 1993.

[20] D. M. Russell. Simple is good: Observations of visualization use amongst the big data digerati. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI 2016)*, pp. 7–12. ACM, 2016.

[21] B. Shneiderman, c. Plaisant, M. Cohen, S. Jacobs, N. Elmqvist, and N. Diakopoulos. *Designing the User Interface: Strategies for Effective Human-Computer Interaction (6th Edition)*. Pearson, 2017.

[22] L. Takayama and E. Kandogan. Trust as an underlying factor of system administrator interface choice. In *CHI'06 extended abstracts on Human factors in computing systems*, pp. 1391–1396. ACM, 2006.

[23] J. W. Tukey. *Exploratory data analysis*, vol. 2. Reading, Mass., 1977.

[24] L. Tweedie, B. Spence, D. Williams, and R. Bhogal. The attribute explorer. In *Conference companion on Human factors in computing systems*, pp. 435–436. ACM, 1994.

[25] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE transactions on visualization and computer graphics*, 22(1):649–658, 2016.

[26] W. Wright, D. Schroh, P. Proulx, A. Skaburskis, and B. Cort. The sandbox for analysis: concepts and methods. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 801–810. ACM, 2006.