# Enterprise Data Analysis and Visualization: An Interview Study

Sean Kandel, Andreas Paepcke, Joseph M. Hellerstein, and Jeffrey Heer

**Abstract**—Organizations rely on data analysts to model customer engagement, streamline operations, improve production, inform business decisions, and combat fraud. Though numerous analysis and visualization tools have been built to improve the scale and efficiency at which analysts can work, there has been little research on how analysis takes place within the social and organizational context of companies. To better understand the enterprise analysts' ecosystem, we conducted semi-structured interviews with 35 data analysts from 25 organizations across a variety of sectors, including healthcare, retail, marketing and finance. Based on our interview data, we characterize the process of industrial data analysis and document how organizational features of an enterprise impact it. We describe recurring pain points, outstanding challenges, and barriers to adoption for visual analytic tools. Finally, we discuss design implications and opportunities for visual analysis research.

**Index Terms**—Data, analysis, visualization, enterprise.

## 1 Introduction

Organizations gather increasingly large and complex data sets each year. These organizations rely on data analysis to model customer engagement, streamline operations, improve production, inform sales and business decisions, and combat fraud. Within organizations, an increasing number of individuals — with varied titles such as "business analyst", "data analyst" and "data scientist" — perform such analyses. These analysts constitute an important and rapidly growing user population for analysis and visualization tools.

Enterprise analysts perform their work within the context of a larger organization. Analysts often work as a part of an analysis team or business unit. Little research has observed how existing infrastructure, available data and tools, and administrative and social conventions within an organization impact the analysis process within the enterprise. Understanding how these issues shape analytic workflows can inform the design of future tools.

To better understand the day-to-day practices of enterprise analysts, we conducted semi-structured interviews with 35 analysts from sectors including healthcare, retail, finance, and social networking. We asked analysts to walk us through the typical tasks they perform, the tools they use, the challenges they encounter, and the organizational context in which analysis takes place.

In this paper, we present the results and analysis of these interviews. We find that our respondents are well-described by **three archetypes** that differ in terms of skill set and typical workflows. We find that these archetypes vary widely in programming proficiency, reliance on information technology (IT) staff and diversity of tasks, and vary less in statistical proficiency. We then discuss how organizational features of an enterprise, such as the relationship between analysts and IT staff or the diversity of data sources, affect the analysis process. We also describe how collaboration takes place between analysts. We find that analysts seldom share resources such as scripts and intermediate data products. In response, we consider possible impediments to sharing and collaboration.

Next we characterize the analysis process described to us by our respondents. Our model includes **five high-level tasks**: *discovery, wrangling, profiling, modeling* and *reporting*. We find that discovery and wrangling, often the most tedious and time-consuming aspects of an analysis, are underserved by existing visualization and analysis tools. We discuss recurring pain points within each task as well as difficulties in managing workflows across these tasks. Example pain points include integrating data from distributed data sources, visualizing data at scale and operationalizing workflows. These challenges are typically more acute within large organizations with a diverse and distributed set of data sources.

We conclude with a discussion of future trends and the implications of our interviews for future visualization and analysis tools. We argue that future visual analysis tools should leverage existing infrastructures for data processing to enable scale and limit data migration. One avenue for achieving better interoperability is through systems that specify analysis or data processing operations in a high-level language, enabling retargeting across tools or platforms. We also note that the current lack of reusable workflows could be improved via less intrusive methods for recording data provenance.

## 2 Related Work

Many researchers have studied analysts and their processes within intelligence agencies [5, 18, 24, 25, 30]. This work characterizes intelligence analysts' process, discusses challenges within the process, and describes collaboration among analysts. Although there is much overlap in the high-level analytic process of intelligence and enterprise analysts, these analysts often work on different types of data with different analytic goals and therefore perform different low-level tasks. For example, enterprise analysts more often perform analysis on structured data than on documents and emails.

Others have characterized tasks and challenges within the analysis process [1, 21, 26, 34, 35]. Amar et al. [1] propose a set of low-level analysis tasks based on the activities of students in an Information Visualization class. Their taxonomy largely includes tasks subsumed by our *profile* and *model* tasks and does not address the other tasks we have identified. Russell et al. [34] characterize high-level sensemaking activities necessary for analysis. We instead identify specific tasks performed by enterprise analysts. Sedlmair et al. [35] discuss difficulties evaluating visualization tools in large corporations, including acquiring and integrating data. We discuss common challenges within these subtasks. Kwon and Fisher [26] discuss challenges novices encounter when using visual analytic tools. In contrast, our subjects are largely expert users of their tools.

Fink et al. [9] performed an ethnographic study of cyber-security analysts. They find that visual analytic tools in this domain have limited interoperability with other tools, lack support for performing ad hoc transformations of data, and typically do not scale to the necessary volume and diversity of data. We find similar issues across multiple domains.

Several researchers have articulated the importance of capturing provenance to manage analytic workflows [2, 11, 12, 15]. Such sys-

- *Sean Kandel is with Stanford University, e-mail: skandel@cs.stanford.edu.*
- *Andreas Paepcke is with Stanford University, e-mail: paepcke@cs.stanford.edu.*
- *Joseph M. Hellerstein is with UC Berkeley, e-mail: hellerstein@cs.berkeley.edu.*
- *Jeffrey Heer is with Stanford University, e-mail: jheer@cs.stanford.edu.*

*Manuscript received 31 March 2012; accepted 1 August 2012; posted online 14 October 2012; mailed on 5 October 2012.*
*For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.*

1077-2626/12/$31.00 © 2012 IEEE Published by the IEEE Computer Society

tems often include logs of automatically recorded interaction histories and manual annotations such as notes. Here, we discuss the difficulty of recording provenance in enterprise workflows, which typically span multiple tools and evolving, distributed databases.

Multiple research projects have demonstrated benefits for collaborative analysis and developed tools to foster such collaboration. Isenberg et al. [20, 21] discuss design considerations for supporting synchronous, co-located collaboration. Similar to intelligence analysts [25], we have found that most enterprise analysts collaborate asynchronously. We discuss how and when these analysts collaborate. Others [7, 14, 37] discuss design considerations to support work parallelization and communication in asynchronous social data analysis. We discuss the types of resources that analysts must share to enable collaboration and the impediments they face.

Researchers have also advocated the use of visualization across more phases of the analysis life-cycle [22]. Our analysis corroborates this suggestion. Examples include visualizations to assist schema mapping for data integration [13, 33] and visual analytics for data deduplication [6, 23]. Our interviews indicate that such tools are sorely needed, and that visualization might be further applied to tasks such as discovery and data manipulation.

## 3 Methods

We conducted semi-structured interviews with enterprise analysts to better understand their process and needs. We use the term "analyst" to refer to anyone whose primary job function includes working with data to answer questions that inform business decisions.

### 3.1 Participants

We interviewed 35 analysts (26 male / 9 female) from 25 organizations. Our interviewees held a number of job titles, including "data analyst", "data scientist", "software engineer", "consultant", and "chief technical officer". The organizations were from 15 sectors including healthcare, retail, social networking, finance, media, marketing, and insurance (see Figure 1 for the complete list). The organizations ranged in size from startups with fewer than 10 employees to corporations with tens of thousands of employees. The analysts ranged from Ph.D. graduates in their first year of work to Chief Data Scientists with 10-20 years of experience.

We recruited interviewees by emailing contacts at organizations within our personal and professional networks. In some cases, we emailed analysts directly. In others, we emailed individuals who forwarded us to analysts within their organization. This recruitment strategy introduces potential bias in our results. For example, the majority of our interviewees were based in Northern California. Also, many of the analysts were sophisticated programmers. To be clear, our research goal is to characterize the space of analytic workflows and challenges, *not* to quantify the prevalence of any specific activity. Other methods, such as surveys or analyzing online job postings, would be better suited for quantifying our findings.

### 3.2 Interviews

We conducted semi-structured interviews with 1 to 4 analysts at a time. We began each interview with a quick introduction describing the purpose of the interview: to understand analysts' day-to-day work practices and any challenges they face. Each interview lasted from 45 minutes to 2 hours. Whenever possible, we interviewed analysts on location at their job. For interviewees outside of Northern California, we conducted interviews over the phone or via Skype.

We asked open-ended questions and encouraged interviewees to describe their lived experiences, such as *"walk us through a recent analysis task"* or *"describe a time you worked with another analyst."* In each interview, we sought to learn the following:

- What tasks do analysts perform?
- What kinds of data sources and formats do they work with?
- What tools do they regularly use and how do they use them?
- How do analysts vary in terms of programming proficiency?
- How do analysts vary in terms of statistical proficiency?
- What are the "results" of analysis?
- What happens to these results "downstream"?
- What are recurring bottlenecks and pain points?
- How important is scalability?
- How important is sociability?
- What is the relationship between analysts and other business units?
- Where are analysts situated within their corporate hierarchy?

In addition to open-ended questions, we asked interviewees to describe the tools and data sets they use within their current organization. During the interviews we took extensive notes.

### 3.3 Analysis

We analyzed our interview data using an iterative coding method. We grouped common practices, tools, challenges and organizational issues into high level categories. We iterated and refined these categories as we gathered more data. In the remainder of the paper, we describe the types of analysts we encountered and the social context in which they perform analysis. We then describe recurring patterns in the analytic process and enumerate the most common and severe challenges faced. Throughout, we use representative quotes from respondents to support our analytic claims.

## 4 Analysts and Organizational Context

We found that analysts vary widely in their skill sets and common tasks. We categorized analysts into three archetypes based on the tasks they perform and the tools they use. We then report recurring themes in how organizations structured both personnel and data, and discuss collaboration practices within analysis teams.

### 4.1 Analyst Archetypes

We asked analysts to describe recent tasks, the tools they used to complete them, and others in their organization who helped them. Based on the responses, we found that analysts varied widely in their programming proficiency and diversity of tools used. We found that analysts generally fall into three archetypes: *hacker*, *scripter*, and *application user*. For each group we discuss the discrepancies in programming proficiency, the range of tasks typically performed, reliance on IT staff and statistical sophistication.

#### 4.1.1 Hackers

Hackers were the most proficient programmers of the three groups and the most comfortable manipulating data. They typically used at least three different types of programming languages. In addition to working with an analysis package (e.g., R or Matlab), they frequently used a scripting language (Python, Perl) and a data processing language (SQL, Pig [29], *etc*). As one data scientist described:

> *I'm not a DBA, but I'm good at SQL. I'm not a programmer but am good at programming. I'm not a statistician but I am good at applying statistical techniques.*

Hackers typically had the most diverse and complex workflows of the three archetypes, characterized by chaining together scripts from different languages that operate on data from distributed sources. Because of their skill set, hackers often completed flexible workloads without assistance from coworkers such as IT staff. For instance, they were more likely to acquire a new data source outside of the organization's data warehouse and integrate it with internal data. Because of their knowledge of query languages such as SQL or Pig, they could also typically run jobs at scale on their own. In some cases, these analysts even helped build and maintain an organization's central data warehouse and database engines.

Analysts in this group often perform less sophisticated statistical models than scripters. They reported that working with larger data sets limited the types of statistical routines they could run on the data. Also, because this group relied less on IT staff for completing certain tasks, they spent more time in early-stage analytic activities prior to modeling.

| | | | Hacker | | | | | | | | | | | | | | | | | | | | | | Scripter | | | | | | | Application User | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Analytics | Biology | Datamart | Finance | Finance | Healthcare | Healthcare | Healthcare | Healthcare | Insurance | Marketing | Marketing | News Aggregator | Retail | Retail | Social Networking | Social Networking | Social Networking | Visualization | Visualization | Web | Web | Analytics | Analytics | Analytics | Finance | Healthcare | Media | Retail | Finance | Insurance | Retail | Retail | Sports | Web Security |
| Process | Discovery | Locating Data | X | X | X | X | X | X | X | X | X | | | | | X | | X | X | X | X | X | | | | | | X | X | | | | | | | | |
| | | Field Definitions | X | X | X | X | X | X | X | X | X | | X | X | | X | | X | X | X | | X | | | | | | | | | | | | | | | |
| | Wrangle | Data Integration | X | X | X | X | X | X | X | X | X | X | X | X | | X | X | X | X | X | | | | X | X | | X | | | X | X | | | | | X | |
| | | Parsing Semi-Structued | X | X | X | X | | | X | | | X | X | X | | X | X | | X | X | X | X | X | X | X | | X | X | X | X | X | | | | | X | |
| | | Advanced Aggregation and Filtering | X | | | | X | | X | X | X | | | | X | X | X | | X | X | X | X | | X | | | X | | | | X | | | | X | | |
| | Profile | Data Quality | X | X | | X | X | X | X | X | X | X | X | X | | X | X | X | X | X | X | X | X | X | X | | X | X | X | X | X | X | X | | | X | X |
| | | Verifying Assumptions | | X | | X | X | | X | X | | X | X | X | | X | X | X | X | X | X | X | | X | X | | X | X | | X | X | X | X | | | | X |
| | Model | Feature Selection | X | | X | X | | | | | | | X | X | X | X | X | | X | X | X | X | X | X | | X | X | | | | X | X | | X | X | | |
| | | Scale | X | X | X | X | X | X | | X | X | X | X | X | X | X | X | X | X | X | X | | X | X | X | X | X | | X | | X | X | X | X | X | X | X |
| | | Advanced Analytics | X | | X | | X | | | | | | X | X | X | X | X | | | | | | | | X | | X | | X | | X | | | X | X | | |
| | Report | Communicating Assumptions | | | | | | | X | X | | | X | X | | X | X | X | X | X | | | | X | | | X | X | X | X | X | | | | | X | X |
| | | Static Reports | | X | X | | X | | | X | X | | X | X | | X | X | X | | | | X | | X | | | | X | X | X | X | | | | | X | |
| | Workflow | Data Migration | X | X | X | X | X | X | | X | X | | | | | X | | X | X | X | | X | | X | | | | | X | | X | | | | | | |
| | | Operationalizing Workflows | | X | X | | X | X | | X | | | X | X | | X | X | X | X | X | | | | X | | | | | X | X | X | | | | | | |
| Tools | Database | SQL | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | | | X | | X | | | | | |
| | | Hadoop/Hive/Pig | X | | X | | | | | X | | | | | | | X | | X | X | | | | X | | | | | | | | | | | | | |
| | | MongoDB | | | | | | | | | | | | | X | | | | | | X | | | | | | | | | | | | | | | | |
| | | CustomDB | X | | | | | X | X | X | X | | | | | | | | | | | | | X | X | | | | | | | | | | | | |
| | Scripting | Java | X | | X | | X | | | X | X | X | | | X | X | X | | X | X | X | | | X | | | | | | | | | | | | | |
| | | Perl | | | | | | | | | | | | | | | | | | | X | X | | | | | | | | | | | | | | | |
| | | Python | X | | X | X | X | X | X | X | | X | X | X | | | | X | X | X | | | | X | | | X | | | | | | | | | | |
| | | Clojure | | | | | | | | | | | | | X | | | | | | | | | | | | | | | | | | | | | | |
| | | Visual Basic | | X | | | | | | | | | | | | | | | | | | | | | | | | | | | | X | | | X | | |
| | Modeling | R | X | | X | | | | | | | | X | X | | | X | X | X | X | X | | X | X | X | X | X | | | X | X | | | | | | |
| | | Matlab | | | | X | | | | | | X | | | | | X | | | | X | | X | | | | X | | X | | | | | | | | |
| | | SAS | X | | | | | | | | | | | | | | X | | | | X | X | X | | | | | | | | | | | | | | |
| | | Excel | | X | | X | X | | | | X | X | X | X | | X | X | X | | | | | X | X | X | | | X | | | X | X | X | X | X | X | X |

Fig. 1. Respondents, Challenges and Tools. The matrix displays interviewees (grouped by archetype and sector) and their corresponding challenges and tools. *Hackers* faced the most diverse set of challenges, corresponding to the diversity of their workflows and toolset. *Application users* and *scripters* typically relied on the IT team to perform certain tasks and therefore did not perceive them as challenges.

Hackers reported using a variety of tools for visualization, including statistical packages, Tableau, Excel, PowerPoint, D3, and Raphäel. Six hackers viewed tools that produce interactive visualizations as reporting tools and not exploratory analytics tools. Since they could not perform flexible data manipulation within visualization tools they only used these tools once they knew what story they wanted to tell with the data.

### 4.1.2 Scripters

Scripters performed most of their analysis within a software package such as R or Matlab. They were able to perform simple manipulations such as filtering and aggregating data, but typically could not perform custom operations such as parsing log files or scraping data off the web. They generally operated on data that had been pulled from the data warehouse by IT staff and stored in an expected format. Some of these analysts could write simple SQL queries (e.g., without joins) to pull data into their analytic tool of choice. In some cases, they were comfortable writing scripts in a scripting language, but typically do not know how to create scripts that run at scale.

Scripters applied the most sophisticated models among the analysts we observed. Advanced modeling was potentially enabled by the breadth of libraries available for analytic packages and the percentage of time these analysts devoted to modeling. The implementation and application of algorithms was more easily done when dealing with data resident on one machine (as opposed to distributed). Scripters often produced visualizations using the statistical package during exploratory analysis. Using the same tool for visualization and analysis permitted them to iterate fluidly between the two tasks. In some cases scripters used a separate tool, such as Tableau, to create interactive dashboards for reporting after the significant insights had been discovered.

### 4.1.3 Application User

The last set of analysts performed almost all operations in a spreadsheet or other dedicated analysis application (e.g., SAS/JMP, SPSS, *etc*). Like scripters, they typically required someone to prepare data for them by pulling it from the warehouse. One Excel user's account was quite typical of most spreadsheet users:

> *All data is in a relational database. When I get it, it's out of the database and into an Excel format that I can start pivoting. I ask the IT team to pull it.*

Application users typically worked on smaller data sets than the other groups and generally did not export data from the spreadsheet except for building reports. In some cases, advanced application users wrote scripts using an embedded language such as Visual Basic. To produce visualizations they typically created charts in Excel or exported data to a reporting tool such as Crystal Reports.

## 4.2 Organization

Enterprise analysts work within the context of a larger organization. Political and social conventions within the organization can and do affect the analysis process. We now discuss three recurring themes.

### 4.2.1 The Relationship between Analysts and IT Staff

Analysts often interacted closely with IT staff to complete aspects of their job. We observed that the IT team regularly provides four primary functions in support of analysis. First, they often maintain data within a centralized warehouse. This maintenance includes ingesting new data sources and ensuring quality across these sources. If an analyst requires new data in the warehouse, the analyst will often work with IT staff to communicate these requirements.

Second, the IT team assists analysts in acquiring data. Analysts, especially application users and scripters, rely on IT staff to query data from the warehouse and export it in an accessible format. For instance, 12 analysts reported having the IT team write SQL queries and convert the output to delimited files or spreadsheets.

Third, the IT team is responsible for operationalizing recurring workflows. One analyst at a media company described the workflows he built as "prototypes". After experimenting with samples of data, the analyst would send a high-level description of the workflow steps — written in English — to IT staff. IT staff then implemented the process to run reliably and at scale. Even hackers relied on IT staff to operationalize tasks that were critical to other business units or had challenging scalability requirements. For example, one analyst at a hedge fund required the IT team to operationalize his workflows to achieve low-latency for high-frequency trading.

Finally, the IT team serves as a source of documentation. Even analysts comfortable writing complex SQL and Hadoop jobs often require IT staff to help find the appropriate data and understand its structure and schema. This reliance on IT staff was particularly true in organizations where data was distributed across many different data sources. Hackers were most likely to use the IT team explicitly for this function, as they were more likely to access data directly from the warehouse. Scripters and application users relied on this function implicitly when receiving data from members of IT.

### 4.2.2 Distributed Data

Analysts, especially in large organizations, worked with data stored in a variety of repositories. 21 analysts reported working with data stored in at least three different formats. For instance, three hackers used data stored in spreadsheets in a shared file system, account data stored in a relational database, and log files stored in HDFS (the Hadoop distributed file system). Many analysis tasks involved integrating data from multiple sources:

> *We had three different sources of data for a customer support interaction. Data gets tracked in customer support, part of Salesforce. The end-user data is stored in our user management system. And all the events are logged into the data warehouse event logs. And if a user buys, this gets logged in the credit card system. So you may have four sources. For me, I have to pull data out of each data source. As it turns out, all the data is in different warehouses ...in different schemas.*

Some analysts performed this integration themselves. In other cases, analysts — especially application users and scripters — relied on the IT team to assemble this data for them.

### 4.2.3 Consumers of Analysis

The results of analysis served many different departments within the organization. For instance, analysts worked with marketing, business development, sales, operations, and design teams. They often translated high-level business questions into low-level analytic tasks. At the end of analysis they typically generated reports in the form of summary statistics, charts or recommendations. Other analysts also consumed these reports to inform future analysis.

Analysts typically shared static reports in the form of template documents or PowerPoint presentations. In some cases, the results of analysis were dynamic reports such as interactive dashboards that enabled end users to filter or aggregate data. In other cases, reports were simply recommendations of actions to take. The results were typically shared via email, a shared file system, or during a group meeting or presentation. Consumers and analysts often iterated on reports to pursue newly-generated hypotheses, modify assumptions, or redefine the problem specification. Because consumers often translate their questions loosely, analysts sometimes misinterpret them. After arriving at a result, these results were often archived in a personal or shared drive, but were not consulted on a regular basis and were difficult to search:

> *We meet every week, we find some interesting insight, and we say that's great. There's no repeatable knowledge, so we end up repeating the process 1 year later.*

## 4.3 Collaboration

In addition to working with IT staff and other business units, analysts were often part of their own analysis unit. Here we describe at which points in their process analysts collaborated and which resources they shared throughout their workflows.

### 4.3.1 Collaboration Process

Analysts reported meeting regularly with other analysts to discuss long-term projects and immediate next steps. However, most analysts reported that they rarely interacted with other analysts to complete a given task. One sentiment, echoed by many analysts, was *"working on a team is the exception in my experience."*

### 4.3.2 Shared Resources

We found analysts shared four types of resources: data, scripts, results and documentation. All the organizations we talked to contain some central repository through which analysts access a large proportion of their data. During analysis, it was common to perform transformations such as sampling, filtering or aggregating data to compute intermediate data products that were used for downstream analysis. These products were typically disseminated via email or on shared drives. In some cases, these intermediate products were stored in the data warehouse. In a few organizations, there were attempts to monitor when new data sets were created. One analyst described a chat room that all analysts monitored throughout the day. When someone created a new data set, a message was *"automatically sent to the chat room, producing ambient knowledge."*

The least commonly shared resource was data processing scripts. We found that analysts typically did not share scripts with each other. Scripts that were shared were disseminated similarly to intermediate data: either through shared drives or email. Analysts rarely stored their analytic code in source control. Analysts with engineering backgrounds noted the difference in process between product code and analysis code; one joked that even when she stored code in a repository, *"svn is more like backup than version control."*

On the other hand, analysts frequently shared their results with each other. These results often took the form of reports or charts. Some analysts used Crystal Reports, others constructed graphs or simply presented summary statistics of their models. Analysts viewed these results during planning meetings or presentations, but did not typically consult these reports during subsequent analysis. The results were not stored in a searchable repository. Most reports were static, preventing others from modifying input parameters or assumptions to see how the results might change. In a few cases, the result of analysis was a parametrizable function or an interactive dashboard with support for filtering or aggregating data.

### 4.3.3 Impediments to Collaboration

We observed three common impediments to collaboration. First, the diversity of tools and programming languages made it difficult to share intermediate code. One quantitative analyst at a hedge fund reported that sharing scripts was too difficult because of the diversity of languages the analysts used: one analyst used Mathematica, another used Perl, and he used Python.

Second, the same analysts reported that finding a script or intermediate data product someone else produced was often more time-consuming than writing the script from scratch. Many (18/35) reported that it was difficult to search for a desired intermediate product. These products were often difficult to interpret, as documentation for these data sets was often more sparse than for data in the warehouse. Because of this, analysts resorted to "blast emails" such as *"has anyone made a data set filtering out users from Latin America?"* This difficulty may in part result from the way data and scripts were typically shared: by storing them on a shared drive or via email. Even when an analyst could find an appropriate script or data set, the product may lack documentation of how it should be used and what assumptions must hold. As one analyst said:

> *You're wary of reusing code because if you blindly reuse it you might miss obvious things that are important to my own code... the same data set can be used for thousands of analyses, so unless you are reusing it for the same exact thing then you might make mistakes.*

Many analysts (25/35) also expressed a general attitude that intermediate products such as code and data were *"ad hoc"*, *"experimental"* or *"throw-away."* As a result, analysts spent less time writing modular, reusable, and parametrizable code and rarely produced documentation. Other analysts noted that a lot of their work ultimately did not validate any useful hypothesis, and so they end up discarding the data or code. One analyst observed, *"you go down a lot of dead ends, and you come up with a bunch of hypotheses. 8 out of 10 are* *dead ends."* The same analyst went on to say he lacked a process to tell others *"don't look for the correlation here, because I looked and its not here. Especially your dead ends – there are no remnants of that."* Analysts intentionally discarded intermediate products because the end result did not seem to be insightful.

## 5 CHALLENGES IN THE ANALYSIS PROCESS

We characterized the tasks within the analysis process based on interviewees' descriptions of their work. We identified five high-level tasks that repeatedly occurred in respondents' analysis efforts:

**Discover** data necessary to complete an analysis tasks. Example tasks include finding a data set online, locating the database tables in a MySQL database, or asking a colleague for a spreadsheet.

**Wrangle** data into a desired format. Example tasks include parsing fields from log files, integrating data from multiple sources into a single file or extracting entities from documents.

**Profile** data to verify its quality and its suitability for the analysis tasks. Example tasks include inspecting data for outliers or errors and examining the distributions of values within fields.

**Model** data for summarization or prediction. Examples include computing summary statistics, running regression models, or performing clustering and classification.

**Report** procedures and insights to consumers of the analysis.

Not all analyses require all five tasks, and not all analysts perform each of them. We now discuss recurring pain points and the challenges of managing workflows across tasks.

### 5.1 Discovery

Throughout their work, analysts acquired data necessary to complete their tasks. Within large organizations, finding and understanding relevant data was often a significant bottleneck.

#### 5.1.1 Where is my data?

For 17 analysts, finding relevant data distributed across multiple databases, database tables and/or files was very time consuming. Organizations often lacked sufficient documentation or search capabilities to enable efficient identification of desired data. Instead, analysts relied on their colleagues: they often asked database administrators or others for help. One analyst described the problem:

> *It is really hard to know where the data is. We have all the data, but there is no huge schema where we can say this data is here and this variable is there. It may be written but the wiki is very stale: pointers don't point to the right place and it changes really fast. The best thing you can learn working here is whom to ask, because in their head a lot of people know a lot of stuff. It's more like folklore. Knowledge is transmitted as you join.*

Some organizations also restricted access to data, requiring an appropriate administrator to grant privileges. In some cases, the administrator who set up the data may have already left the company.

#### 5.1.2 Field Definitions

The difficulties in discovering data were compounded by the difficulty of interpreting certain fields in a database. In at least 16 instances, analysts described situations in which fields were coded and required lookups against external tables. Foreign key definitions help identify the appropriate table to perform lookups, but these definitions were often missing in relational databases and non-existent in other types of data stores. Even without coding, missing units or metadata created ambiguity. For instance, one analyst noted that many date-time fields were stored without timezone information. The analysts had to reconstruct timezones from corresponding geographic information.

In 8 reported cases, schema drift lead to redundant columns. One company we interviewed had a database table containing four columns containing job titles for its users. These columns evolved over time, were often conflicting and there was no documentation describing which column was up-to-date or appropriate to use.

### 5.2 Wrangling

Once an analyst discovered the appropriate data to use, she often needed to manipulate the acquired data before she could use it for downstream analysis. Such data wrangling, munging, or cleaning [22] involves parsing text files, manipulating data layout and integrating multiple data sources. This process, whether managed by IT staff or by analysts, was often time consuming and tedious.

> *I spend more than half of my time integrating, cleansing and transforming data without doing any actual analysis. Most of the time I'm lucky if I get to do any analysis. Most of the time once you transform the data you just do an average... the insights can be scarily obvious. It's fun when you get to do something somewhat analytical.*

Prior work [8, 32] lists a number of data wrangling challenges. We identify three common issues shared by our interviewees.

#### 5.2.1 Ingesting Semi-Structured Data

A number of analysts (23/35) reported issues processing semi-structured data. The most common example was ingesting log files. Parsing log files may require writing complex regular expressions to extract data into relevant fields. Interleaved log files containing multiple event types in a single file can further complicate parsing.

One analyst described working on log files generated by a telecommunications company. The log files contained three types of records for text messages: outbound messages, inbound messages, and receipt confirmation. The analyst needed to define criteria to divide the logs into logical segments. Verifying that criteria accurately splits records can be difficult, especially with collections of log files containing terabytes of data.

So-called "block data" is another common data format that was difficult to parse. In a block format, logical records of data are spread across multiple lines of a file. Typically one line (the "header") contains metadata about the record, such as how many of the subsequent lines (the "payload") belong to the record.

Data from 3rd party services often required a level of processing before analysis could begin. Email campaign providers, credit card processing companies, and other external services often delivered user reports in idiosyncratic formats. One analyst responded that:

> *"[Email campaign providers are] very inconsistent about feed format. Bounces, responses, etc ... are hard to bring in in the right format. We are constantly munging the data to get into our SQL database."*

Although many data sets arrived in these formats, most visualization tools do not support such semi-structured formats, preventing their use at early stages of data transformation.

#### 5.2.2 Data Integration

Another difficulty, reported by 23 analysts, was integrating data from multiple sources. Identifiers useful for joining records across data sets were often missing in one or more data sets, inconsistent across data sources or incorrect for certain records. One hospital analyst recounted issues integrating patient medical data:

> *The easiest patient identifier is the medical record number (MRN). We should have consistent MRNs in any data source but 5 to 10 percent of MRNs are mistyped or incorrect or blank. In emergencies, a patient may get assigned a temporary MRN. Later it's reassigned but sometimes we forget to reassign. We have to identify patients by other means: first name, last name, birthdate, gender. Several data points together might identify a single patient. There may be slight inconsistencies.*

When identifiers were missing or incorrect, analysts derived new methods for linking records. Like the medical researcher above, analysts would match records using rules based on other fields that did not uniquely identify distinct records.

Analysts reported three types of inconsistency in identifiers during integration. First, identifiers used slight variations in spelling or formatting that make direct matches difficult. For instance, a patient's first name might be stored as "John" in one record and "Jonathan" in another. Some analysts defined ad hoc rules ("fuzzy matches") to detect similar items. The analysts then inspected the matches to verify that the records referred to the same entity.

Second, data sources used two different encodings to represent the same identifier. For instance, a state might be identified by its full name (e.g., California) or by its as Federal Information Processing Standard (FIPS) code (e.g., 6). In this case, an analyst must find or construct a mapping between identifiers.

In the third case, identifiers used inconsistent units of measurement or class definitions. Multiple analysts described attempts to consolidate their respective company's industry codes with the North American Industry Classification System (NAICS). Others described difficulties integrating geographic data with varied region definitions. Similarly, many data sets use overlapping conventions for financial quarters. The situation is complicated when sets of regions overlap and one standardization does not subsume the others:

> *The biggest challenges have been making two sources work together in terms of units of analysis. Our industry titles are different than standard industry titles... it would be nice to have mappings between standardizations. We are matching internal standards and outside standards which is hard... We have a region "SF Bay Area" which includes San Jose. CVSA is mapped to metro areas and so San Fran and San Jose are different areas. We need a method of grouping, but they won't even overlap the same. It's worse than hierarchical. You end up losing the data somewhere along the path.*

These integration problems were made more difficult when the data was stored across multiple databases. In response, most analysts reported having to migrate all of the data sets into the same data processing framework.

The lack of support for data integration also impedes the effective use of exploratory visualization tools. Because analysts were often unable to integrate external data to augment visualizations within a tool, they must resort to assembling data outside of the tool. One analyst noted that she spent most of her time integrating data together from disparate data sources to drive visualizations.

### 5.2.3 Advanced Aggregation and Filtering

Some analysts (16/35) noted difficulty performing ad hoc grouping of observations, as in path or funnel analysis [36]. One analyst at a web company investigated the sequence of actions users took before converting to a paying customer, upgrading their accounts, or canceling their accounts. The source data set was a log of user activities on the website, with each entry corresponding to a single activity by a single user. The analysts needed to group activities not only by user, but also by event time, where the time was conditional on other events in the log (i.e., prior to closest conversion). These types of queries in SQL often involve nested subqueries. Similar subqueries are necessary to write filters such as "delete all users who never upgraded their account." Analysts find it difficult or impossible to express these queries within current visualization tools. In response, they must process data outside of the tool and then load it back in, significantly slowing their analysis process.

## 5.3 Profiling

Once required data is assembled and integrated, analysts enter a phase of diagnosing data quality issues and understanding what assumptions they can make about their data. As one analyst quipped: *"I'd rather the data go away than be wrong and not know."*

### 5.3.1 Data Quality

Data sets may contain a number of quality issues that affect the validity of results, such as missing, erroneous or extreme values. Many analysts (22/35) reported issues dealing with missing data. In some cases, observations contained missing or null attributes. Analysts reported using a number of methods for imputing missing values. One organization even had an intern build a dedicated interface for displaying and correcting missing values across data sources. In other cases, entire observations were missing from a data set. Missing observations were much more difficult to detect.

Another common problem is heterogeneous data in a column: a column with an expected type may contain values of another type. This might occur due to errors in automated processes such as log file generation, errors in human data entry, or because of an explicit decision to "overload" the use of a column. For example, a database table at one organization contained a longitude field that was empty for many of the observations. Instead of creating a new column, some analysts decided to overload this field to store additional data unrelated to longitude. This type of error also occurred when IT teams introduced a new field into a log file, breaking existing scripts that expect the files in a certain format.

Other errors include multivariate constraints. For instance, one analyst described a scenario:

> *in one data set there were 4 males [between the ages] 0 to 9 who were pregnant. If I make an assumption about what that means and filter the data, then I am destroying data. For instance, you might infer hospital care in that particular hospital is not very good.*

Analysts reported using visualization and statistical routines to detect errors in their data. One medical analyst described using visualization for outlier detection in machine-generated data: *"We don't have probability rules to detect outlying events. Once we look at enough data, we'll know exactly what is an artifact."* Others relied more heavily on statistical methods to detect outliers: *"I find [visualization] pretty useless. I'm very much a numbers hound. I'm more into playing with the data. I'm very comfortable with numbers. Visualization adds a layer between me and numbers."* This analyst inspected distributions and summary statistics and identified observations that fell outside the normal range. Generally, most analysts reported using a combination of visualization and statistics to inspect data. During inspection, they were also able to gain an understanding of what assumptions they could make about their data.

### 5.3.2 Assumptions

Analysts make assumptions during analysis that inform the types of transformations they use, how they sample data and which models are appropriate. Common assumptions included how values were distributed within an attribute (was an attribute normally distributed?), what values were unique (were there duplicates?), and how different attributes related to each other (was X always greater than Y?). Other assumptions required domain expertise to verify.

> *Once you play with the data you realize you made an assumption that is completely wrong. It's really useful, it's not just a waste of time, even though you may be banging your head.*

An analyst in online advertising described an analysis of ad impressions. For a given campaign, the analysis assumed there were at most 15 impressions per user. However, they saw that some users were receiving up to 300 impressions. The analysts then

> *checked if the campaign settings were set correctly, talked to other people about the logic, and then finally started zeroing in on those people. Then you realize if they change states then they are eligible for another 15 [impressions]. Then it affects how you organize the campaign. In practice it tends not to be just data prep, you are learning about*

> *the data at the same time, you are learning about what assumptions you can make.*

## 5.4 Modeling

After all the required data was assembled and understood, analysts could begin modeling their data.

### 5.4.1 Feature Selection

Many analysts (20/35) reported the biggest difficulty in constructing a model was understanding which of the data fields were most relevant to a given analysis task. It was particularly difficult to understand the relationships among features spread across multiple databases. Also, many fields needed to be transformed before useful patterns would emerge. As one analyst said:

> *In practice right now the biggest differentiator is feature selection: knowing what columns to pay attention to and how to sensibly transform them. Do you take the log of these, do you combine these two? A lot of work is just finding what the units of the columns should be.*

### 5.4.2 Scale

Most respondents (31/35) noted that existing analytic packages, tools or algorithms did not scale with the size of their data sets. The threshold for when data sets were too big was obviously different depending on the tool used. For instance, some application users still used Microsoft Excel 2007, because their organizations would not allow them to upgrade to newer versions. In these cases, analysts could not perform analysis on more than 1,000,000 rows. Scripters were typically limited by the memory requirements of their machine.

Hackers were less limited by large amounts of data, because they could typically run distributed jobs over multiple machines. However, hackers were often limited by the types of analysis they could run because useful models or algorithms did not have available parallelized implementations. As one hacker described, it is difficult to *"take powerful algorithms that work on medium data and make them pluggable in the big data stack."*

Other analysts used sampling but cited that sampling was hard to do correctly without introducing bias into the analysis. Some noted that sampling was especially difficult when the "interesting" observations were sparse. One analyst described their difficulty performing sampling for modeling conversions during funnel analysis:

> *Subsampling can exclude information you actually need... It's not very reasonable for infrequent observations. If you sample down you lose a lot of conversions.*

Issues with scale were even more prominent when dealing with visualization tools. In two cases, respondents had not heard of existing tools (such as Tableau) that would have been sufficient for their reported data sizes. For others, scale was fundamentally an issue, both in terms of the number of observations and the number of attributes. Existing visualization tools simply could not load enough data into the tool. In other cases, data could be loaded, but operations such as brushing and linking could not be performed at interactive rates. To cope with scale, three of our respondents were building custom data processing engines. One company built their own database engine that pre-computes possible combinations of filters and rollups in their charts. To combat combinatorial explosion, they analyze which columns are typically viewed together.

Interviewees also believed that visualization does not scale to high dimensional data. Some stated that most exploratory tools do not allow them to visualize more than two or three dimensions:

> *[G]raphical representation is at best two or three dimensional. Three dimensions won't tell me very much about how 300 variables interact.*

### 5.4.3 Visualizing Statistical Models

Analysts would like to apply advanced analytics routines and visualize the results. Though many tools have facilities such as drawing best-fit regression lines, analysts using more advanced machine learning methods (14/35) expressed a desire for visualization tools to help explore these models and visualize their output. However, analysts' descriptions of these potential tools were often vague and imprecise: they sensed a need, but were unsure of the form that a successful solution would take.

## 5.5 Reporting

Analysts typically reported insights gained from modeling to other analysts or business units. The two most-cited challenges were communicating assumptions and building interactive reports.

### 5.5.1 Communicating Assumptions

One complaint about distributing and consuming reports (made by 17 analysts) is poor documentation of assumptions made during analysis. Analysts typically performed a sequence of operations that can affect the interpretation of results, such as correcting outliers, imputing missing data or aggregating data. These operations are often context specific, with no standards for each analysis.

In other cases, analysts imposed their own definitions on underspecified concepts. One medical analyst analyzed patient episodes that correspond to all visits to a hospital to treat a given symptom. However, the database did not contain an episode identifier associated with each patient visit. The analysts had to use heuristics, such as the duration between visits, to group hospital visits into episodes. This heuristic was imprecise, as hospitals may treat a patient concurrently for two different symptoms or for the same symptom after a long period of time. Analysts often lost track of all the operations they performed and their rationale for performing them.

Even when assumptions were tracked, they were often treated as footnotes instead of first-class results. One analyst cited that his boss often looked at summary charts without reading the fine print. For instance, an average calculated from three data points would be marked with an asterisk that was then regularly overlooked.

### 5.5.2 Static Reports

A number of analysts (17/35) also complained that reports were too inflexible and did not allow interactive verification or sensitivity analysis. Often reporting and charting tools were used directly on the output data and contained no knowledge of how the original input data was filtered, transformed or modeled. Much of this work was done before output data was loaded into the tool. Because reporting tools have no access to data provenance, it was often impossible to modify parameters or assumptions to see how the conclusions would change. Viewers can not verify questions such as "how might user acquisition rates change if more money was spent on marketing?"

## 5.6 Workflow

We found that analysts engaged in an iterative, non-linear process in which they cycle among the tasks described above. Managing workflows across these steps brings a number of its own challenges.

### 5.6.1 Data Migration

Analysts, especially hackers, often used multiple tools and databases to complete their tasks. Different tools and environments often required data in different formats. About half of our respondents (16/35) claimed that the most tedious part of analysis was moving data between tools and warehouses. One data scientist noted the tedium of having to *"Run a Hadoop job, then run a Hadoop job on results, then awk it... Hadoop job chained to Hadoop job chained to a Python script to actually process data."* Scripters and applications users often used separate tools for reporting than they used for wrangling, profiling and modeling.

#### 5.6.2 Operationalizing Workflows

During analysis, analysts generated a number of intermediate products including scripts, spreadsheet formulas and data sets. It was often difficult to assemble these products into a repeatable, reliable and scalable process. Analysts reported that they often explored multiple hypotheses in parallel and create multiple intermediate data products in the process. Reconstructing a repeatable workflow is difficult without a coherent linear history of the operations performed. Even with a coherent history, an existing workflow may break when applied to new or updated input data. This new input data may contain nuances not accounted for that would cause existing code to break. Finally, analysts reported that they wrote experimental code that could not run on large data sets or at necessary speed in real-time systems. They therefore required the IT team to operationalize many of their workflows.

## 6 Future Trends

Looking forward, trends in technology and the analytic workforce will compound the challenges faced by enterprise analysts, with strong implications for the design of visual analytic tools.

### 6.1 The Availability of Public Data

As more and more public data — including government records, financial records, and social network activity — becomes available, organizations will allocate more resources to ingest and integrate this data with their own. Ingesting publicly available data can often be difficult, requiring analysts to crawl and scrape websites or parse data from unstructured and semi-structured sources. In some cases, public data is made accessible through web APIs. In many other cases, organizations — especially those required by law to disclose information — release data in formats that are difficult to process (such as PDF files). An analyst at a large hedge fund noted that their organization's ability to make use of publicly-available but poorly-structured data was their primary advantage over competitors.

In recent years, there have been an increasing number of so-called "data marts", such as InfoChimps.org, that aim to make public data more accessible. Even so, integrating public data with an internal warehouse poses challenges. As discussed previously, many organizations develop internal coding standards for entities such as geographic locations or industry codes. Often, these codes differ from external data. Two sources of public data might also have different coding standards. Moreover, public data often lacks documentation, posing additional challenges to discovery and profiling.

### 6.2 The Rise of Hadoop

Of our analysts, 8/35 reported using Hadoop and IDC predicts the market for Hadoop software will increase by an order of magnitude by 2018 [28]. The increasing popularity of Hadoop could compound challenges in discovery. With relational databases, organizations typically design a database schema and structure incoming data upon load. This process is often time-consuming and difficult, especially with large complex data sets. With Hadoop, analysts typically take advantage of is its ability to operate on less structured data formats. Instead of structuring the data up front during ingest, organizations commonly dump data files into the Hadoop Distributed File System (HDFS) with little documentation. Analysis of this data then requires parsing the data during Map-Reduce jobs or bulk reformatting to load into relational databases. While remaining unstructured, the data may be difficult to search and profile due to the lack of a defined schema. In some cases, the analysts who originally imported and understood the data may no longer work at the company or may have forgotten important details.

### 6.3 A Growing Demand for "Hacker" Analysts

Over the next few years, we see three factors driving an increasing demand for "hacker"-level analysts. First, constrained IT departments are making it necessary for analysts to be self-serving. When discussing recruitment, one Chief Scientist said *"analysts that can't program are disenfranchised here"*; IT support was prioritized for shipping products, not helping analysts experiment on code.

Second, the increasing scale of data requires many organizations to perform in-database analytics. Analysis software tools such as R and Matlab do not currently scale. Instead, analytic routines are performed within the data warehouse, typically in a shared-nothing parallel database (such as those offered by Aster, Greenplum, or Teradata) or via Map-Reduce or related higher-level languages such as Pig. Analysts therefore need to be adept at both statistical reasoning and writing complex SQL or Map-Reduce code.

Finally, organizations are frequently relying on multiple processing frameworks and tools as requirements evolve. For instance, some organizations will use relational databases to support interactive queries and analysis, rely on Hadoop for batch jobs and processing log files, and also require analysts who can build "prototype" models in R. One analyst noted:

> *Diversity is pretty important. A generalist is more valuable than a specialist. A specialist isn't fluid enough. We look for pretty broad skills and data passion. If you are passionate about it you'll jump into whatever tool you need to. If it's in X, I'll go jump in X.*

These observations are supported by a recent McKinsey report [27] which estimates the demand for big data analysts (a category similar to our observed "hackers") will triple by 2018.

### 6.4 Analysis Teams Are Growing

As the number of analysts increase across organizations, the size of analytic teams should also grow. We expect that efficient collaboration will become both increasingly important and difficult. We see a growing emphasis on better collaboration practice within the larger organizations we observed. This emphasis was shared particularly among managers who observed the inefficiencies of poor collaboration amongst their subordinates. The managers noted that the inefficiencies led not only to repeated work but to inconsistent results. In one large retail company, the directer of six analytic teams noted that multiple analysts would submit conflicting reports of a metric, such as turnover. The analysts used inconsistent assumptions to calculate their results, most of which were not communicated to the business units consuming these reports.

## 7 Design Implications

We now discuss design implications for visual analytic tools based on the challenges and trends identified in our interviews.

### 7.1 Workflow Breakdowns

Our interviews suggest that many of the breakdowns in analysis occur in the early phases of a workflow or transitioning between tasks in a workflow. Despite this, visualization is often typically applied to isolated late stages of the workflow, including reporting and exploring a single data set at a time. Despite much research from the database and statistics communities [4, 8, 16, 19, 31, 32], little visualization research addresses discovery, wrangling or profiling challenges. Visual analytic tools that enable efficient application and assessment of these data mining routines could significantly speed up the analysis process.

Tools that extend their data query facilities to operate over partially structured data will enable analysts to immediately apply visualization and analytics to a much wider set of data and better serve early-stage analysis. Such tools may require additional algorithms and interaction techniques for type induction and structure inference. As an example, multiple analysts cited the popularity of the commercial tool Splunk, which enabled them to write queries directly against log files without first structuring their data. The analysts noted that Splunk had limited support for visualization and creating dashboards, leaving an opportunity for visualization systems that could enable analysts to begin visualization over unstructured data. Splunk's command-line interface was popular among analysts experienced in programming, but not approachable for those with less experience.

Tools that can connect directly to existing data warehouses can better integrate into analysts' workflows by limiting data migration. If a tool uses its own proprietary data source to process data, then an analyst must migrate data in and out of the tool for it to be useful, impeding fluidity. One analyst liked Google Refine's graphical interface for data transformation, but found it unsuitable for cleaning data in his SQL database because *"that requires me to export everything to CSV and play around there and then I have to put stuff back in the database."*

Analysis is often an iterative process of acquisition, wrangling, profiling and modeling. Although many tools today contain transformation languages, most lack support for common transformation tasks such as integrating new data, window functions and filters or aggregates with nested subclauses. For instance, languages should support filters that remove all employees with salaries in the 95th percentile and window functions that compute rolling averages in time-series data. More complex transformations might be more easily represented with procedural or imperative programming. The lack of support for such transformations requires analysts to transform their data outside of their tool.

Of course, increasing the complexity of a system increases the engineering burden of supporting direct connection to existing data warehouses. Well-designed declarative languages can decrease this burden by limiting the number of primitive operations that need to be implemented across various run times. Still, constraints on data processing languages may make it difficult to run certain types of transformations at scale, or at all, directly within a data warehouse. As one example, transformations that rely on relative positions of observations within a data set are not expressible in standard SQL.

### 7.2 Support Scalable Visual Analytics

One clear implication of our studies is the need for visualization methods that scale. Scaling visualization requires addressing both perceptual and computational limitations. Visualizations that render raw data suffer from over plotting with even moderately large data sets and certainly when applied to data sets containing billions of observations. Visual analytic tools must consider using density or aggregation based plots such as histograms and binned scatter plots [3] for large data sets.

One approach to improved scalability is to leverage existing data processing engines for manipulating data. By creating adapters to common systems such as parallel databases and Hadoop, analytic tools can leverage existing infrastructure to scale to large data sets. For instance, Tableau can translate statements in its internal representation into queries that run on distributed databases. However, simply connecting to existing systems can not achieve interactive rates supporting brushing and linking over large data sets. Visual analytic tools could benefit from server-side pre-processing and aggregation to enable interactive exploration in the client.

Tool builders should also consider how approximation approaches might be applied to scale visual analytic tools. Sampling data can speed up querying but may introduce bias [10]. Ideally, various sampling strategies could be applied directly to a data source from within the tool. This capability would enable more fluid application of various strategies and evaluation of their effectiveness. Other approximation approaches might include online aggregation [10, 17], whereby analysts can visualize the incremental results as they are computed. It remains future work to examine how low-latency query processing over data subsets of various resolutions impact both the quantity and quality of analysis.

### 7.3 Bridge the Gap in Programming Proficiency

The increasing demand for "hackers" highlights the types of tasks that need to be achieved to perform analysis within an enterprise. The inability of scripters and applications users to manipulate data from diverse data sources and at scale makes them dependent on others and limits their effectiveness. Visual analytic tools should strive to bridge this gap in programming skill by providing direct manipulation interfaces for tasks such as data acquisition and wrangling. To empower hackers, direct manipulation interfaces might also expose the underlying logic of the tool.

### 7.4 Capture Metadata at Natural Annotation Points

If available, a tool should augment intermediate products such as scripts and data with additional metadata. Such metadata might include the script's author, the rationale for an analysis procedure or assumptions made about the input data. The metadata can enable more efficient search over products and simplify the interpretation of results by others. How to best represent and interact with this metadata could itself be an interesting visual analytics problem.

However, many analysts are hesitant to spend time documenting their process because of the number of dead-ends they encounter and intermediate products that get thrown away. One approach to record metadata is to instead increase the utility of recorded metadata by imposing conventions or constraints. Where a user has to make a decision (i.e., file naming), can tools help them make a more useful choice? For instance, many analysts save intermediate data sets in files. All these files will require names, in which analysts often record valuable metadata in an inconsistent and unstructured format; e.g., using "customers_europe_18_24" to indicate they created a file storing customer data for European customers aged 18 to 24. Instead, a tool might impose some structure on the naming procedure so that this metadata can be searched over more easily in the future. By intervening at already existing annotation points, tools might limit the perceived overhead of annotation.

## 8 Conclusion

This paper presented the results of interviews with 35 data analysts within commercial organizations. We presented a model of phases of analytic activity and enumerated the challenges faced by analysts within these phases. Finally, we discussed the consequences of trends in technology and human resources, and presented corresponding design implications for visual analysis tools.

As the scale and diversity of data sources increases within enterprises, there is an opportunity for visual analytic tools to improve the quality of analysis and the speed at which it takes place. Tools that simplify tasks across the analytic pipeline could empower non-programmers to apply their statistical training and domain expertise to large, diverse data sets. Tools that help manage diverse sets of procedures, data sets, and intermediate data products can enable analysts to work and collaborate more effectively.

## References

[1] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Proc. IEEE Information Visualization (InfoVis)*, pages 111–117, 2005.

[2] S. P. Callahan, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, and H. T. Vo. VisTrails: visualization meets data management. In *Proc. ACM SIGMOD*, pages 745–747, 2006.

[3] D. B. Carr, R. J. Littlefield, W. L. Nicholson, and J. S. Littlefield. Scatterplot matrix techniques for large N. *Journal of the American Statistical Association*, 82(398):424–436, 1987.

[4] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani. Robust and efficient fuzzy match for online data cleaning. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, SIGMOD '03, pages 313–324, New York, NY, USA, 2003. ACM.

[5] G. Chin, O. A. Kuchar, and K. E. Wolf. Exploring the analytical processes of intelligence analysts. In *Proc. ACM Human Factors in Computing Systems (CHI)*, pages 11–20, 2009.

[6] P. Christen. Febrl -: an open source data cleaning, deduplication and record linkage system with a graphical user interface. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 1065–1068, New York, NY, USA, 2008. ACM.

[7] C. M. Danis, F. B. Viégas, M. Wattenberg, and J. Kriss. Your place or mine?: visualization as a community component. In *Proc. ACM Human Factors in Computing Systems (CHI)*, pages 275–284, 2008.

[8] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE Trans. Knowledge & Data Engineering*, 19(1):1–16, 2007.

[9] G. Fink, C. North, A. Endert, and S. Rose. Visualizing cyber security: Usable workspaces. In *Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on*, pages 45 –56, 2009.

[10] D. Fisher, I. Popov, S. Drucker, and m. schraefel. Trust me, I'm partially right: Incremental visualization lets analysts explore large datasets faster. In *Proc. ACM Human Factors in Computing Systems (CHI)*, pages 1673–1682, 2012.

[11] J. Freire, D. Koop, E. Santos, and C. T. Silva. Provenance for computational tasks: A survey. *Computing in Science and Engineering*, 10:11–21, 2008.

[12] D. Gotz and M. X. Zhou. Characterizing users' visual analytic activity for insight provenance. *Information Visualization*, 8:42–55, 2009.

[13] L. M. Haas, M. A. Hernández, H. Ho, L. Popa, and M. Roth. Clio grows up: from research prototype to industrial tool. In *ACM SIGMOD*, pages 805–810, 2005.

[14] J. Heer and M. Agrawala. Design considerations for collaborative visual analytics. *Information Visualization*, 7:49–62, 2008.

[15] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Trans. Visualization & Computer Graphics (Proc. InfoVis)*, 14:1189–1196, 2008.

[16] J. M. Hellerstein. Quantitative data cleaning for large databases, 2008. White Paper, United Nations Economic Commission for Europe.

[17] J. M. Hellerstein, P. J. Haas, and H. J. Wang. Online aggregation. In *Proc. ACM SIGMOD*, pages 171–182, 1997.

[18] R. J. Heuer. *Psychology of Intelligence Analysis*. Center for the Study of Intelligence, 1999.

[19] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[20] P. Isenberg, D. Fisher, M. Morris, K. Inkpen, and M. Czerwinski. An exploratory study of co-located collaborative visual analytics around a tabletop display. In *Proc. IEEE Visual Analytics Science and Technology (VAST)*, pages 179–186, 2010.

[21] P. Isenberg, A. Tang, and S. Carpendale. An exploratory study of visual information analysis. In *Proc. ACM Human factors in Computing Systems (CHI)*, pages 1217–1226, 2008.

[22] S. Kandel, J. Heer, C. Plaisant, J. Kennedy, F. van Ham, N. H. Riche, C. Weaver, B. Lee, D. Brodbeck, and P. Buono. Research directions in data wrangling: Visualizations and transformations for usable and credible data. *Information Visualization*, 10:271–288, 2011.

[23] H. Kang, L. Getoor, B. Shneiderman, M. Bilgic, and L. Licamele. Interactive entity resolution in relational data: A visual analytic tool and its evaluation. *IEEE Trans. Visualization & Computer Graphics*, 14(5):999–1014, 2008.

[24] Y. Kang, C. Görg, and J. Stasko. Evaluating visual analytics systems for investigative analysis: Deriving design principles from a case study. In *Proc. IEEE Visual Analytics Science and Technology (VAST)*, pages 139–146, 2009.

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

[26] B. Kwon, B. Fisher, and J. S. Yi. Visual analytic roadblocks for novice investigators. In *Proc. IEEE Visual Analytics Science and Technology (VAST)*, pages 3–11, 2011.

[27] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers. Big data: The next frontier for innovation, competition, and productivity, May 2011.

[28] C. W. Olofson and D. Vesset. Worldwide Hadoop-MapReduce ecosystem software 2012–2016 forecast. May 2012.

[29] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins. Pig latin: a not-so-foreign language for data processing. In *Proc. ACM SIGMOD*, pages 1099–1110, 2008.

[30] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proc. International Conference on Intelligence Analysis*, 2005.

[31] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10:334–350, 2001.

[32] E. Rahm and H. H. Do. Data cleaning: Problems and current approaches. *IEEE Data Engineering Bulletin*, 23, 2000.

[33] G. G. Robertson, M. P. Czerwinski, and J. E. Churchill. Visualization of mappings between schemas. In *Proc. ACM Human Factors in Computing Systems (CHI)*, pages 431–439, 2005.

[34] D. M. Russell, M. J. Stefik, P. Pirolli, and S. K. Card. The cost structure of sensemaking. In *Proc. ACM Human Factors in Computing Systems (CHI)*, pages 269–276, 1993.

[35] M. Sedlmair, P. Isenberg, D. Baur, and A. Butz. Evaluating information visualization in large companies: Challenges, experiences and recommendations. In *Proc. CHI Workshop Beyond Time and Errors: Novel Evaluation Methods for Information Visualization (BELIV)*, 2010.

[36] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web usage mining: discovery and applications of usage patterns from web data. *SIGKDD Explorations Newsletter*, 1(2):12–23, Jan. 2000.

[37] M. Wattenberg and J. Kriss. Designing for social data analysis. *IEEE Trans. Visualization & Computer Graphics*, 12(4):549–557, July 2006.