Making Sense of Exploratory Data Analysis

Adam Rule - January 10, 2017

Individuals and organizations increasingly rely on data analysis to generate insights and make decisions. When reviewing, revising, or replicating these analyses, it is often necessary to inspect the underlying process as small changes to how data are collected, cleaned, or modeled can lead to drastically different results. One increasingly popular means of tracking data analyses is to perform them in computational notebooks. These notebooks allow analysts to interleave code and results with explanatory text. Yet, analysts and their colleagues often struggle to make sense of them as exploratory analyses tend to produce notebooks that are long and disorganized. This confusion reflects an underlying tension between exploratory analysis and clearly explaining process.

This research seeks to understand the tools and techniques data analysts use to track and share their analytical process. It argues that understanding how the tension between data exploration and process explanation manifests itself in practice makes it possible to design software that enables analysts to communicate their process more effectively.

Three studies will evaluate this thesis. The first will examine how analysts currently track and share their analyses within academic laboratories. The second will examine the structure of computational notebooks shared on Github, a popular file-sharing website. Together, these studies will investigate how the tension between data exploration and process explanation manifests itself in practice. A third study will build on this result by developing, deploying and evaluating extensions to Jupyter Notebook, a widely used computational notebook, that augment how data analysts track and share their analyses.

1. INTRODUCTION

The cost of collecting, storing, and manipulating data has fallen dramatically over the past 50 years, enabling data to multiply in nearly every sphere of life. Businesses increasingly rely on data analysts – sometimes called *data scientists* or *business intelligence analysts* – to extract insights from data so they can make informed decisions. Scientists collect and model data to generate new knowledge. Governments munge data to learn about their constituents and set policy.

Demand for data analysts is outstripping supply. McKinsey, a consulting firm, estimates that by 2018 the United States alone will face "*a shortage of 140,000 to 190,000 people with analytical expertise and 1.5 million managers and analysts with the skills to understand and make decisions based on the analysis of big data*" [16]. There is an urgent need to both train new analysts and develop tools and techniques that enable them to work more effectively.

Data analysis is "*looking at data to see what it seems to say*" [31]. This looking typically follows an iterative process of generating and testing hypotheses. Insights are highly dependent on the questions asked as well as the methods and reasoning used to answer them; so much so that two analysts given the same dataset may draw vastly different conclusions. Resuming or replicating such a dynamic process is difficult. Even during analysis, deciding what to do next can require extensive knowledge of what one has already done and why.

1.1. The Problem: Making Sense of Exploratory Data Analysis

For these and other reasons, analysts and their colleagues often need to review the steps and reasoning underlying an analysis. They need to *make sense* of the analytical process so they can audit, replicate, or revise it. Yet, presenting exploratory data analysis in a way that others (or a future self) can understand is difficult.

Data analysis is routinely performed by writing and executing blocks of computer code. As analysts iteratively write and execute code, they tend to produce large collections of similarly named files (Figure 1A). These can be difficult to organize or make sense of. Taking time during the analysis to document each hypothesis tested, the method of testing it, and the result is tedious, especially as many analyses do not validate a useful hypothesis [14]. Yet without this documentation, explaining an analysis after-the-fact can be even more challenging.

Numerous software tools have been developed to help track and share data analyses [11, 13, 30]. One increasingly popular means of doing so is to use computational notebooks. These enable analysts to interleave code and computed results with explanatory text (Figure 1C). Whereas, before, analysts had to copy and paste code and results from their various files into a separate report (Figure 1B), computational notebooks enable them to write, run, and explain their analyses in a single document. Despite the advantages of computational notebooks, analysts struggle to make sense of them. In place of their large collections of similarly named files, they have long and disorganized notebooks full of partial results and "*dead-ends*". Analysts may spend considerable time cleaning and curating these notebooks before they are able to understand their own steps and reasoning, much communicate them to others. The challenge remains of reusing the byproducts of exploratory data analysis to clearly explain the process used to generate an insight.

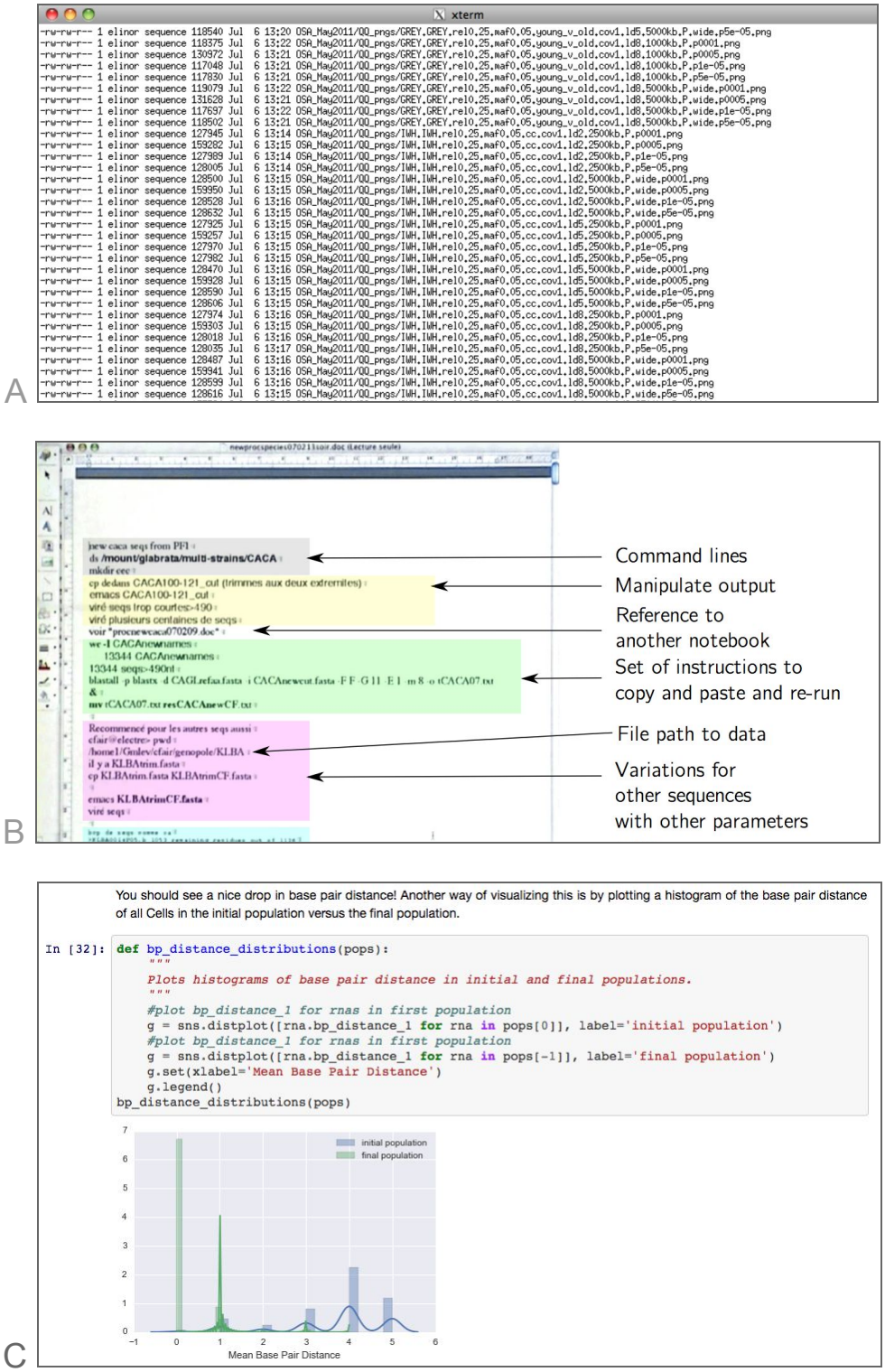

Figure 1: Multiple methods of tracking the process of exploratory data analysis
(A) Partial list of output files for a single computational biology analysis. From [10]
(B) Word document describing a computational biologist's analytical steps. From [29]
(C) Computational notebook combining code, a graphical result, and explanatory text. From [38]

1.2. Research Questions and Thesis

This thesis seeks to understand the challenges analysts face explaining their exploratory data analyses to others and themselves. It seeks to understand how information technology might assist analysts in making sense of their own and other's analyses, particularly, the steps and reasoning used to generate an insight. To that end, this research seeks to answer the following questions:

1. *What tools and techniques do analysts use to generate and share clear explanations of their exploratory data analyses?*
2. *What challenges do they face doing so?*
3. *How might computational notebooks better support the generation and sharing of clear explanations of exploratory data analyses?*

Underlying these questions is the thesis that:

> *Understanding how the tension between exploring data and explaining process manifests itself in practice makes it possible to design software that enables analysts to share their process more effectively.*

This research is also based on the assumption that:

> *Software design has the ability to impact the way analysts perform, record, and share their work.*

The proposed research involves observing data analysts and modifying the tools they use to perform and share their work. Since many analysts now rely on computational notebooks, these observations and modifications will focus on the use of Jupyter Notebook, a widely-used computational notebook. Before describing the proposed studies in detail, we review relevant literature and prior work.

2. RELATED WORK

2.1. Making Sense of Data Analysis

Data analysis is a widely practiced activity. One study, using figures from the Bureau of Labor Statistics, estimated that there were 10 million data analysts in the United States in 2012 [25]. Jupyter Notebook, one of many tools used by analysts, estimates that it alone has over 3 million users [8].

While it is difficult to precisely describe an activity practiced by so many people, several studies have characterized the general process of data analysis. Interviewing 35 enterprise data analysts, Kandel et al. found their work involved iterative phases of collecting, cleaning, profiling, modeling, and reporting data [14]. From his own observations, Guo similarly characterized data analysis as an iterative process of preparing, analyzing, reflecting on, and disseminating data [10]. These accord with Russell et al.'s earlier description that data analysis is iterative sensemaking; "*searching for a representation and encoding data in that representation to answer task-specific questions*" [24].

The iterative nature of data analysis poses problems when trying to track and convey the process of data analysis. From his interviews, Kandel concluded that analyst need to capture more metadata about their process, including parameters, their rationale, and assumptions made along the way, so they can understand the process later [14]. However, he highlighted that "*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". He suggests intervening at natural annotation points, such as when analysts name their output files, to intelligently suggest helpful filenames that include metadata about the analysis.

Guo similarly highlights the difficulty analysts face managing files and metadata [10]. He cites a personal email in which one bioinformatics PhD student writes:

> *Often, you really don't know what'll work, so you try a program with a combination of parameters, and a combination of input files. And so you end up with a massive proliferation of output files. You have to remember to name the files differently, or write out the parameters every time. Plus, you're constantly tweaking the program, so the older runs may not even record the parameters that you put in later for greater control. Going back to something I did just three months ago, I often find out I have absolutely no idea what the output files mean, and end up having to repeat it to figure it out.*

Moreover, Guo notes that when disseminating results "*the main challenge here is how to consolidate all of the various notes, freehand sketches, emails, scripts, and output data files created throughout an experiment to aid in writing*". Even with their results in hand, analysts struggle to reconstruct the parameters, steps, and reasoning used to generate them.

Sharing analytical process is also hampered by the amount of professional judgement involved. In their study of analysts at the International Monetary Fund (IMF), Harper and Sellen found that analysts routinely had to interpolate, transform, or fill in data in non-trivial ways [12]. While the final staff reports may have looked objective, there were many human decisions underlying them. Harper and Sellen concluded that the more professional judgement used to produce information, the harder it is to share asynchronously over email or through computer databases. They note that this does not necessarily mean that technology cannot be used to share such information, but rather that the social processes involved in this sharing needs to be supported as well. For example, they note that analysts at the IMF preferred to share their reports in person using paper printouts so they and the receiving party could work through the report page by page, discussing how various figures had been produced. Software for sharing these reports would need to support this social vetting as well.

This need for social vetting reflects how difficult it is to externalize decisions and understanding of the data in reports. In her seminal "The Marks are on the Knowledge Worker" Alison Kidd notes that the act of writing notes on paper is often more for the sake of informing the worker than for saving information [15]. Summarizing her argument, she states that "*the valuable marks are on the knowledge worker rather than on the paper or on the electronic file*". This echos Tabard's finding that the process of managing scripts and results is an essential part of computational biologists' analytical process as they make sense of their data [29]. It helps them discover what is worth saving, what is not, and how it fits together. These studies cast data analysis as a sensemaking process where the sense made is the result of managing information, but not necessarily apparent in the managed information itself.

Sharing data analyses is also complicated by the tension between actual and idealized processes. This is apparent in the creation of staff reports at the IMF which are meant to be self-explanatory, but require a meeting with the author to explain their construction if they are to be reused in any way [12]. Suchman

similarly noted the tension between actual and idealized practice in an accounting office, and argued that technology needs to support the actual "messy" work rather than merely the idealized form [28]. Data analysis may seem a rigorous, linear process, but that rigor is the result of iterative and exploratory work based on hunches and hypotheses.

2.2. Provenance in Visual Analytics

This thesis focuses on data analyses performed by editing and executing code. However, there are other forms of analysis that primarily involve interacting with statistical or visualization software. Understanding these other forms of analysis can inform the design of tools for programmatic analysis.

Substantial effort has been devoted to understanding how visualization software might help analysts track their work [6, 9, 13, 21]. In the visual analytics community, this information about analytical process is referred to as "*provenance*". Researchers in visual analytics broadly agree on the need to track provenance but define the term differently [21]. Among other things, provenance may refer to the origin of data, sequence of visualizations generated with it, or the line of reasoning guiding the analysis (Figure 2). Due to the ambiguity of the term "*provenance*", this thesis instead uses the terms "*process*", or "*steps and reasoning*" to refer to the sequence of thoughts and actions used to generate an insight. Reasons for tracking provenance are similarly diverse and may include supporting undo/redo actions, showing what parts of a dataset has been explored, or maintaining memory and awareness of previous states of the analysis.

Prior work in visual analytics has also identified significant barriers to tracking, visualizing, and understanding provenance [35]. When tracking provenance, there are tradeoffs between manual and automatic methods. Manual methods, such as note-taking, require substantial vigilance but tend to capture meaningful events and analysts' thought process. Automatic methods, such as logging interactions with software, require little user effort, but tend to capture only low level events, such as interactions with dropdowns and sliders, which reveal little about higher-level actions or the analyst's thoughts. Systems such as HARVEST have shown that it is possible to group these low-level events into more semantically meaningful actions such as querying or filtering, but have only been tested through informal interviews [7]. It remains to be seen if these advanced, automatic provenance tracking systems improve analysts ability to recall or communicate process. One way to manage these tradeoffs is to combine automatic and manual tracking. In their summary of the 2014 IEEE VIS workshop on provenance Xu et al. state that "*a promising direction is the development of 'hybrid' or 'semi-auto' approaches, i.e., mixing the manual and automatic capture to combine their strength*" [35]. However, few studies have evaluated this approach. One by Groth and Streefkerk found no significant difference in the duration or accuracy of analyses performed with access to a hybrid process history, automatic history, or no history at all [9].

When visualizing analytic process, many visual analytics systems use some variation of a node-link diagram or graph (Figure 3) [1, 2, 4, 9, 19, 20, 27]. These show an overview wherein nodes represent visualization states and links represent the actions taken to move from one state to another. Other variations of the node-link diagram depict dependencies between the programmatic routines used to generate the visualization [6] or are graphs users manually create to make sense of collections of textual documents [18, 33]. In early research, such as the VizTrail system shown on the left of Figure 3, node-link diagrams were simple graphs annotated with function or state names [6]. More recent examples

such as Graphtrail, on the right of Figure 3, expand nodes to show how the visualization looked at that stage of the analysis [4]. While providing an overview of the entire analysis, node-link diagrams do not always make clear why certain actions were performed or in what order. Moreover, evaluation of node-link diagrams has primarily consisted of case studies or small tests with 1-3 users.

Types of Provenance Information	
Data	The history of changes and movement of data, which can include subsetting, data merging, formatting, transformations, or execution of a simulation to ingest or generate new data
Visualization	The history of graphical views and visualization states
Interaction	The history of user actions and commands with a system
Insight	The history of cognitive outcomes and information derived from the analysis process, including analytic findings and hypotheses
Rationale	The history of reasoning and intentions behind decisions, hypotheses, and interactions

Purposes for Provenance	
Recall	Maintaining or recovering memory and awareness of the current and previous states of the analysis
Replication	Reproducing the steps or workflow of a previous analysis
Action recovery	Maintaining the action history that allows undo/redo operations and branching actions during analysis
Collaborative communication	Communicating and sharing data, information, and ideas with others who are conducting the same analysis
Presentation	Communicating the insights or progression of the analysis with those who are not directly involved with the analysis themselves, such as general public, upper levels of management, or analysts focusing on other areas
Meta-analysis	Reviewing the analytic processes themselves in order to understand and improve aspects of the analysis (such as process efficiency, training efficiency, or analytic strategies)

Figure 2: Summary of provenance types and purposes from [21]

Regarding the interpretation of provenance information, few studies in visual analytics have examined if provenance tools improve analysts' ability to remember or communicate their steps or reasoning [22]. Instead, most evaluations have measured whether access to provenance information reduces task time, increases the number of insights generated, or improves the accuracy of insights. One notable exception by Dou et al. looked at how well a custom provenance visualization helped analysts reconstruct another analyst's process [3]. They found that, for a wire-fraud detection task, analysts viewing a custom visualization of another analyst's interactions with the analysis software were able to reconstruct 60% of their strategies, 60% of their methods, and 79% of their findings. While encouraging, the visualizations used in this study were highly customized to the task of detecting wire fraud and not likely to generalize to other activities. Moreover the reviewers needed an average of 13 minutes to review an activity which originally took only 20 minutes to complete. They spent 65% of the time, reconstructing only 60% of the strategies and methods.

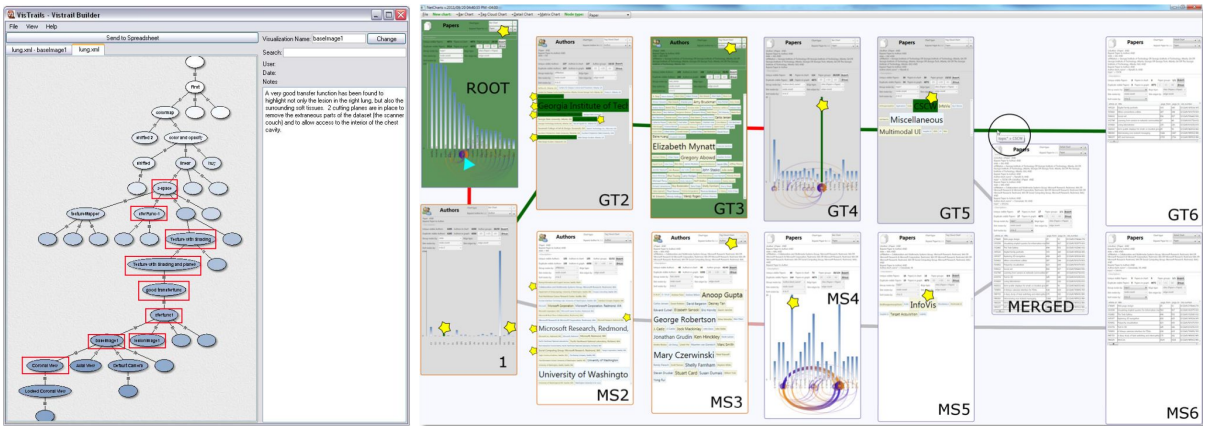

Figure 3: Example node-link diagrams of analytical steps in VizTrails [6] and GraphTrail [4]

3. INTERVIEWS WITH USERS OF COMPUTATIONAL NOTEBOOKS

The research described above characterizes data analysis as an iterative and subjective process. Moreover, it highlights the challenge of tracking, visualizing, and communicating analytic process in a way that can be easily understood. This challenge persists despite recent advances in the way data analyses are performed and shared. In the following section we describe interviews with users of computational notebooks, an increasingly popular means of data analysis, which highlight a fundamental tension between supporting exploratory analysis and generating clear explanations of analytical process. We argue that understanding this tension is critical to designing technologies that help analysts track, share, and make sense of exploratory analyses.

One increasingly popular means of tracking data analyses, particularly those that involve editing and executing code, is to conduct them in computational notebooks. Computational notebooks contain linear collections of cells in which analysts can write and execute blocks of code to analyze data, generate visualizations, or render richly-formatted text. Computational notebooks have seen widespread adoption in recent years due to their free or low cost, flexibility, and reliance on popular scripting languages with powerful analysis libraries. Jupyter Notebook (Figure 4), one of the most widely used computational notebooks, has an estimated 3 million users [8].

Despite their popularity, little research has looked into the efficacy of computational notebooks in helping analysts and their managers reconstruct analytical process. To better understanding the benefits and shortcomings of computational notebooks, in August and September 2016 we interviewed 6 graduate students at UC San Diego who use Jupyter Notebook to track their analyses. Interviewees were recruited from a range of disciplines (e.g., Bioinformatics, Cognitive Science, Neuroscience, Geological Science) via a convenience sample. Each interview lasted about 30 minutes during which time participants were asked to describe their use of Jupyter Notebook and show example notebooks documenting recent work.

Interviewees were generally enthusiastic about using Jupyter Notebook. In particular, they liked being able to store analysis code, output, and commentary in a single document. This was opposed to keeping multiple versions of separate script and output files in a folder on their computers. They also liked being able to flexibly share analyses will colleagues and managers via the notebooks' many export options. For

example, they could render and email the notebook as a static PDF or HTML document, or store it on GitHub where it would be publically available for colleagues to review online.

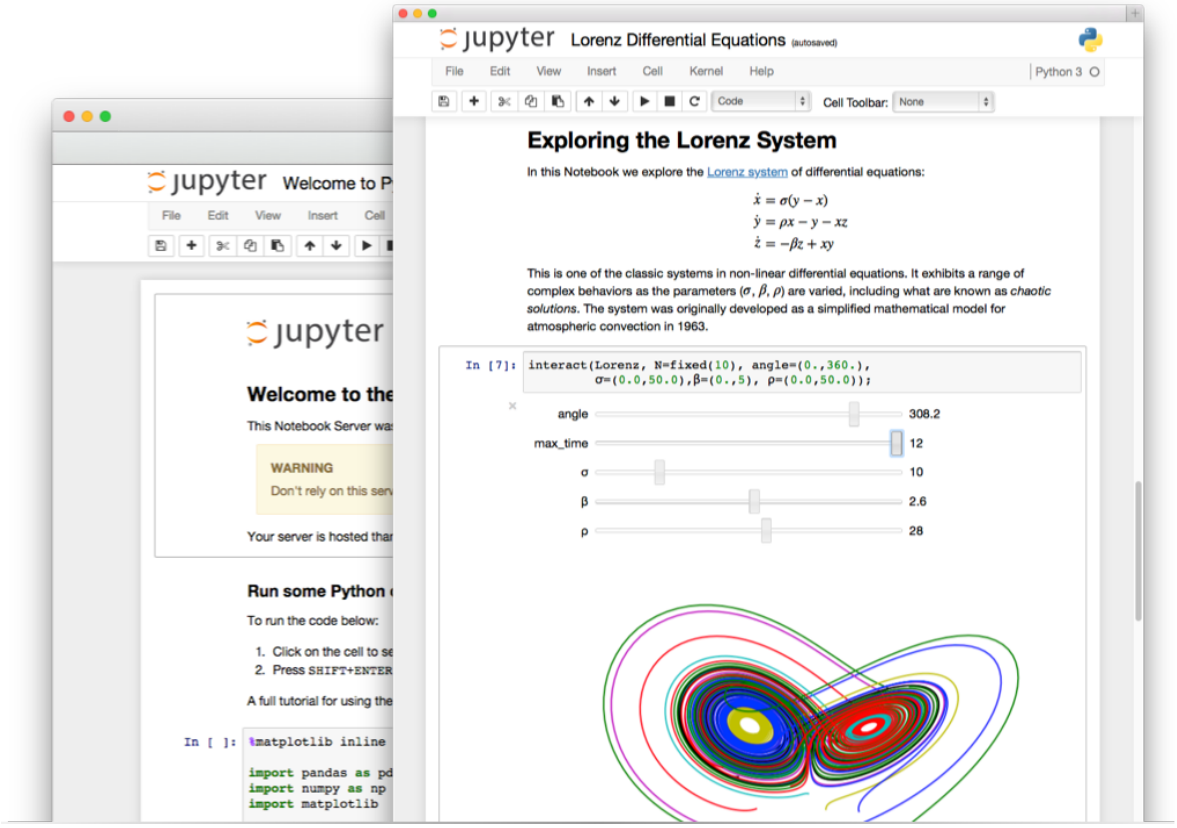

Figure 4: Example Jupyter Notebooks from Jupyter.org

However, interviewees also struggled to review and share their notebooks. All described notebooks as easily getting "*too long*" to navigate or read. In particular, interviewees struggled to identify high-level results or the steps taken to achieve those results. Several of the interviewees' had manual workarounds for this problem. Some participants listed high-level goals and results in the first cell of their notebooks, so they would be visible when they opened the notebook. Other workarounds included keeping separate "*working*" and "*sharing*" notebooks, or conducting analyses outside the notebook and only copying the most interesting results into the notebook. Others split their analysis into several smaller notebooks so it would be easier to find and comprehend each stage of analysis. For example, Figure 5 shows one bioinformaticians' self-described "*stream-of-consciousness*" notebook which, at over 45,000 pixels in length, took a significant amount of scrolling to navigate. This notebook has little explanatory text to differentiate its multiple, nearly identical analyses and plots. When later publishing the results of this analysis, the bioinformatician painstakingly split the notebook into 16 distinct notebooks documenting different steps in the analysis.

Figure 5: Stream-of-conscious notebook spanning 45,000 pixels. Navigating, much less making sense of, long exploratory notebooks like this one is difficult.

These workarounds reveal a tension between supporting exploratory analysis and generating clear explanations of the steps and reasoning used to generate an insight. While analysts can use the notebooks for exploratory analysis, the notebook's linear structure is more conducive to the step-by-step explanation expected when sharing results than the iterative back-and-forth of analysis. As a result, notebooks are often either too lengthy and disorganized to make sense of or too sanitized to reveal the complex sequence of steps and reasoning behind the analysis. Curating more complex notebooks to be effective memory and communicative aids remains a tedious manual process.

This tension between supporting exploratory analysis and clear explanation of process is not unique to computational notebooks, but is evident in the interviews, observations, and systems described in prior work. However, the advent of computational notebooks provides a unique opportunity to study this tension, and investigate how information technology can support the clear communication of analytical process. Notebooks centralize the analytical process and are widely used both for analysis and sharing of results. Moreover the most widely used computational notebook, Jupyter Notebook, supports extensions that may be used to both study how analysts actually perform their work and encourage new ways of working. The proposed research leverages this opportunity to study and modify how analysts make sense of exploratory data analyses.

4. RESEARCH PLAN

The proposed research aims to answer the following three questions:

1. *What tools and techniques do analysts use to generate and share clear explanations of their exploratory data analyses?*
2. *What challenges do they face doing so?*
3. *How might computational notebooks better support the generation and sharing of clear explanations of exploratory data analyses?*

It aims to do so by by 1) examining how data analysts within academic research laboratories conduct and explain their analyses to one another, 2) examining how exploratory data analyses are shared online via

computational notebooks, and 3) co-designing and evaluating modifications to computational notebooks with analysts.

4.1. Study 1: Computational Ethnography of Data Analysis

4.1.1. Research Question

The first study will address the first two research questions:

> *What tools and techniques do analysts use to generate and share clear explanations of their exploratory data analyses?*
>
> *What challenges do they face doing so?*

Effectively answering these questions requires careful consideration of methodology. Analysts may be able to self-report some of their tools and techniques during interviews, but a more robust answer is likely to come from observing them as they work on real-world analyses. Data analysis, however, is a cognitively demanding task, so observation methods such as contextual inquiry or think-aloud protocols which require participants to interrupt their work to talk with the researcher may be inappropriate as they would break the participant's train of thought. Alternatively guerilla observation, in which the researcher observes but does not talk to the participant, may strip any observations of the context needed to understand why certain tools or techniques are being employed. Any method of direct observation is limited by the amount of time a researcher is available to observe, which is particularly worrying when the phenomena of interest, in this case the explanation and sharing of analytical process, is relatively rare.

To address these methodological issues, we propose conducting a *computational ethnography* which mixes recording workstation activity, selective in-person observation, and post-recording interviews to unobtrusively study the everyday practice of data analysis. Computational ethnography "*leverages automated and less obtrusive (or unobtrusive) means for collecting in situ data reflective of real end users' actual, unaltered behaviors using a software system or a device in real-world settings*" [35]. Since much data analysts takes place on a computer, workstation recording will enable us to collect data about analysts typical work practices without requiring in-person observation. This will allow simultaneous collection of several participant's work practices. These recordings will be supplemented by observations of key moments when analysts are sharing their work (e.g. lab meetings, one-on-one meetings) and post-recording interviews in which analysts explain their tool and technique use while reviewing portions of their recorded work.

4.1.2. Prior Work: Traces Computer Activity Recorder

In prior research we developed and deployed Traces, a rich computer activity recorder, to study everyday knowledge work [23]. Traces includes facilities for tracking clicks, keystrokes, window and application events, and can capture a time-lapse video of user's computer screens by taking periodic screenshots. Traces also supports experience sampling wherein users can be asked to respond to a question with either written text or recorded audio.

In prior studies we leveraged Traces to conduct long-term recording of everyday knowledge work. In two separate studies participants used Traces to record two continuous weeks of computer activity. Recorded

data were saved to an external USB key so as to preserve storage space on users' computers and provide a physical reminder to participants that they were being recorded. After recording, we conducted interviews with participants in which we showed them portions of their recording and asked them to comment on what they were doing at the time. Using Chroniviz [5], software developed in our lab to analyze multiple streams of data, participants were able to view the time-lapse videos of their work and navigate to key moments of interest which they had either previously marked, or identified during the interview using features such as a sudden spike in mouse or keyboard activity (Figure 6).

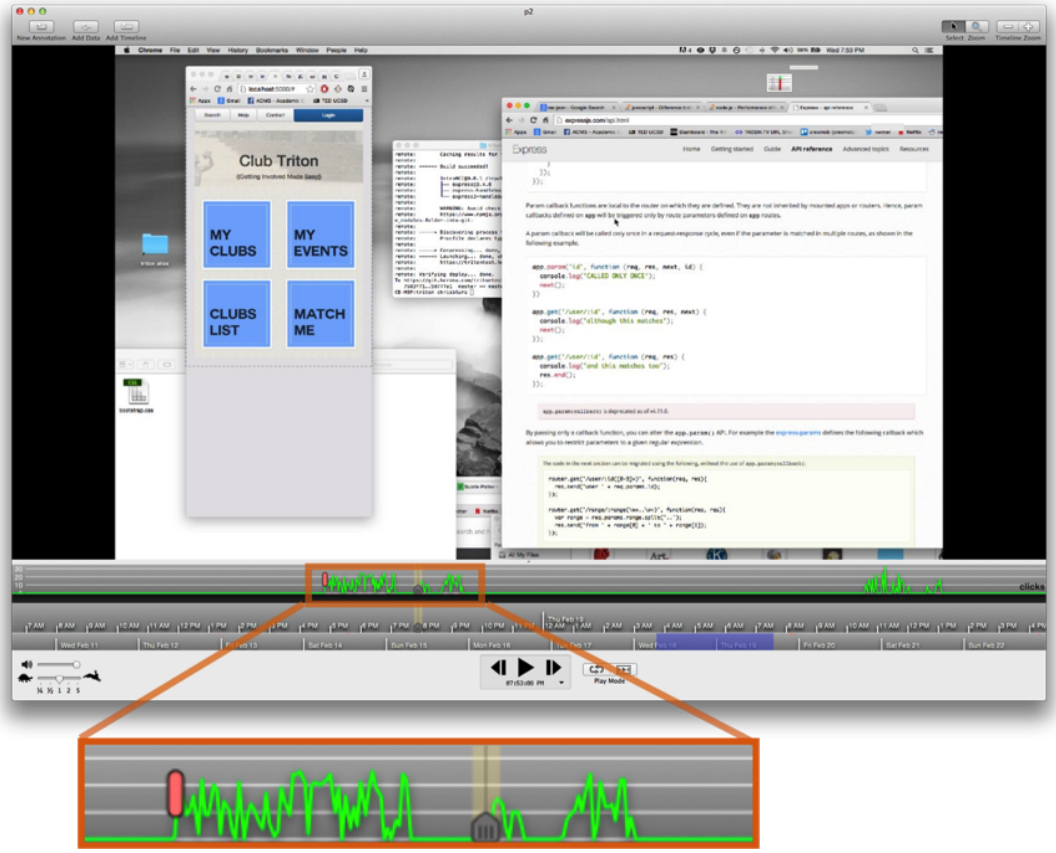

Figure 6: Data recorded in Traces [23] can be visualized in Chronoviz [5] to allow participants to navigate their recordings based on moments marked with an experience sample (red bar) or changes in level of general computer activity (green line).

4.1.3. Methods and Materials

I propose using a similar method to study how data analysts conduct and explain their exploratory data analyses. Twelve or more participants will be recruited from laboratories around UC San Diego that use Jupyter Notebooks to conduct their research, with multiple participants coming from each lab. Potential participating labs include the Knight Lab which studies microbiomes, the Frazer Lab which studies pathological genomics, and the Voytek Lab which studies neural oscillations, all of which conduct and share their analyses using Jupyter Notebooks. Participants may be motivated by the desire to better understand their analytical practice and contribute to the development of tools that make it easier to

conduct and share exploratory data analyses. Participants will be enrolled and consented at their lab's weekly research meeting.

Participants will be asked to record their everyday computer activity for a period of one week using Traces and a Jupyter Notebook extension designed to track notebook interactions in more detail. While the final state of notebooks can be easily reviewed, it is difficult to tell how notebooks were created or modified over time. For example, what cells were edited, re-run, or deleted? How were notebooks curated by splitting, deleting, or rearranging cells during and after analysis to support later recall and communication of reasoning or steps? While the time-lapse videos from the Traces recording could be used to answer these questions, reviewing them would be tedious and any quantitative summary of notebook use would have to come from hand-coding videos. Instead, we propose developing an extension to Jupyter Notebook that tracks the full history of a notebook's use including the creation, editing, and deletion of each cell. To this end I have developed a preliminary Jupyter Notebook extension that tracks a full history of each cell's inputs and outputs. More development effort is needed to track rearrangement, splitting, and deletion of cells. However, once recorded, combining this notebook-level history with computer-level recording will provide unique insight into the way data analysts conduct their work in and outside their notebooks.

In addition to tracking their computer activity, participants will be observed at moments of interest during the recording period. These will include moments of individual data analysis, one-on-one meetings with other analysts, and presentation of analytical results to colleagues in lab meetings. These observations will provide context for the activities we observe in the computer activity recordings.

After the recording period we will interview participants about the tools and techniques they used to conduct and share their exploratory analyses. These will be scheduled as pair interviews in which two analysts from the same lab will be interviewed simultaneously. While there is a risk of group-think when interviewing multiple people at the same time, pair interviews will allow participants to engage in a constructive interaction [32] in which they can explain, question, and clarify their own and each other's work practices. Moreover, if we are unable to directly observe many one-on-one meetings between analysts, we may use a portion of the pair-interview time to have analysts select a recent analysis and explain it to their co-interviewee. These interviews will be aided by auto-confrontation in which participants will be able to review and discuss portions of their recorded activity using Chronoviz (Figure 6).

4.1.4. Analysis and Measures

The recordings, observations, and interviews will produce a large amount of data. This will include 1) click, keystroke, window, and screenshot data from Traces, 2) full notebooks and notebook history from the Jupyter Notebook extension, 3) notes from in-person observations, and 4) videos and transcripts from participants interviews at the end of recording. These diverse data will be used to triangulate an understanding of the tools and techniques analysts use to conduct and share exploratory data analyses.

While our focus will be on the sharing and communication of analyses, it will be important to situate these practices within the larger context of the analytical activity. For this reason, the workstation activity

recordings will provide a useful backdrop for understanding how the tools and techniques used to share data analyses compare with those used to conduct the analysis in the first place.

A first round of quantitative and qualitative analysis will identify the tools and techniques analyst use to share their analyses. Tools may include handwritten notes, whiteboards, emails, Jupyter Notebooks, or other information management software. Techniques may include sketching on whiteboards, splitting cells or notebooks, deleting "dead-end" analyses, or manually documenting key results in Word documents. Qualitative methods including grounded theory will be used to summarize the artifacts and techniques described in observation notes and interview transcripts. Quantitative methods will be used to analyze Traces and notebook logs to identify the prevalence and ordering of tool and technique use.

A second round of analysis will examine 2-3 episodes of data analysis and sharing in detail. This more detailed examination will focus on synthesis rather than analysis, considering how diverse tools and techniques are assembled to complete the sharing of an analysis. For example, an episode may show that the analyst first attempts to make sense of their analysis by drawing a diagram of their process on a scratchpad or whiteboard, translates this understanding into a linear arrangement of their Jupyter Notebook cells, and then employes Github file hosting, email, and a brief face-to-face meeting to share the analysis with their manager.

4.1.5. Expected Results

In this first study, we expect to find that:

1. The production and sharing of data analyses involves numerous tools, but centers on computational notebooks
2. Analysts devote substantial time and effort to tracking and curating their process to make it easy to recall and communicate, though much of this effort occurs after the initial analysis
3. Analysts demonstrate a number of workarounds for tracking and curating their process both during analysis and afterwards when preparing a notebook or other artifacts for sharing or archiving
4. Social practice, more so than than documentation, is used to convey where and how professional judgement was employed

We expect the tools, techniques, and workarounds we identify to guide the participatory design of notebooks in the third study.

4.2. Study 2: Artifact Analysis of Computational Notebooks Shared Online

4.2.1. Research Question

The second study will also address the first research question:

> *What tools and techniques do analysts use to generate and share clear explanations of their exploratory data analyses?*

In contrast to the first study, which seeks to understand the process of conducting and sharing analyses within the rich social structure of an academic laboratory, this study will examine how analyses are shared online, stripped of the social practices available in a physical lab.

4.2.2. Sharing Jupyter Notebooks on Github

Github.com is a popular website for storing and sharing computer files, particularly software files. Millions of people use it to store copies of their software, aided by the built-in version control which records changes to files over time. This version control enables users to track changes to their files, revert to prior versions, and merge contributions from multiple people. These files are stored in repositories, many of which are publicly available online as Github does not charge users to host their files so long as they are shared publicly.

Github is increasingly being used to store the scripts and results of data analysis, particularly in the form of Jupyter Notebooks. As of July 2015, GitHub hosted over 200,000 Jupyter Notebooks [26]. In addition to providing free file hosting for the notebooks, GitHub also renders the notebooks on their website so that visitors can see the notebook as it would appear in the Jupyter program without installing or running Jupyter itself. Many of the participants in our initial interviews used this feature to share their analyses with colleagues and advisors.

The large number of notebooks publicly available on GitHub provides a unique opportunity for research. While the full process of analysis is not apparent in these notebooks, notebooks stored on GitHub are documents that analysts use to share their analytical process, both with their future selves and others. Individual notebooks may be inspected for their structure and the version control history may be used to partially examine how these notebooks change over time.

4.2.3. Methods and Materials

The primary method of this study will be programmatic artifact analysis in which a large number of Jupyter Notebooks are analyzed for their content and structure. Initially we will use GitHub's API [37] to sample about 1,000 Jupyter Notebooks from user's public repositories. Each notebook will be downloaded along with its full version history. Should this analysis produce useful results, we may elect to analyze a larger sample of notebooks, up to the full population of publicly available notebooks on GitHub. This larger analysis will likely require supercomputing resources, but the initial analysis of 1,000 notebooks should be feasible on a high-power desktop computer.

4.2.4. Analysis and Measures

This proposed study is itself an exploratory data analysis, so it is difficult to predict what aspects of the notebook's content or structure will be most informative. While intermediate results will inform what we analyse in the end, initially we plan to investigate:

- number of cells in each notebook
- type (code or markup) and ordering of cell inputs
- type (textual, graphical) and ordering of cell outputs
- total length, in rendered pixels, of each notebook
- number of previous versions of each notebook

- whether each notebook is stored by itself, or in a repository with other notebooks

4.2.5. Expected Results

It is difficult to predict what we will find with this exploratory analysis. However, based on our pilot interviews we expect:

1. Notebooks will tend to be either very short, or very long demonstrating different styles of analysis
2. Short notebooks will be more likely to be stored in repositories with other notebooks than longer ones
3. Notebooks, and especially long notebooks, will tend to have relatively few markup cells with explanatory text and instead consist primarily of code and output
4. Most notebooks will have only one version saved

The results of this analysis will demonstrate the diversity of ways analysts use Jupyter Notebooks to share their analyses. While there are a number of examples of clear and elegantly structure notebooks on GitHub [34], we expect this to find this is the exception rather than the norm. Instead, we expect that notebooks shared on GitHub will reflect the messy, iterative, and exploratory data analysis used to produce them.

4.3 Study 3: Participatory Design of Computational Notebooks

4.3.1. Research Question

The first two studies will examine how analysts perform and share exploratory data analyses, first within academic laboratories, and then within the broader community of GitHub users online. We expect to find that, unmodified, the byproducts of data analysis such as notebooks, code files, and output files do not provide a clear explanation of the steps or reasoning used to perform an analysis. Instead, we expect to find that analysts spend considerable time curating notebooks, reports, and other artifacts to make the process of their analysis clear to others, or even their future selves. Especially within a physical lab, we expect that social practices will play a role in how these artifacts are curated and shared, particularly when conveying where and how professional judgement was employed.

The final proposed study will ask:

> *How might computational notebooks better support the generation and sharing of clear explanations of exploratory data analyses?*

While lab policies, social pressures, and conventions might have the greatest impact on how analysts perform and share their work, we believe that information technology has the potential to shape the way analysts work. In particular, the design of computational notebooks and other data analysis software might enable or hinder data exploration or process explanation. Since many data analysts use Jupyter Notebooks to conduct and share their analyses, this study will involve iteratively developing and deploying Jupyter Notebook extensions with data analysts.

4.3.2. Extending Jupyter Notebook

Jupyter Notebook is open source and extensible software. Being open source, the software can be rewritten to include new features or operate in an entirely different way. Jupyter Notebook can also be modified without changing the source code by writing and installing Javascript extensions. These extensions can change the look and operation of the notebook, for example changing the way cells are rendered on the screen or calling an additional function whenever a cell is executed. These extensions enable the program to be modified without needing to be fully re-installed, and users can have multiple extensions running at the same time. This extensibility provides a unique opportunity to study how changes to the tools analysts use might affect their ability to perform and share analyses.

4.3.3. Methods and Materials

The primary method of this study will be participatory design [17]. Participants will be recruited from laboratories around UC San Diego that use Jupyter Notebooks to conduct their research. For example the Knight Lab which studies microbiomes, the Frazer Lab which studies the genomics of disease, and the Voytek Lab which studies neural oscillations all conduct and share their analyses using Jupyter Notebooks. Participants may be motivated to participate by the desire to contribute to the development of notebook extensions that make it easier to track, review, and share their analyses.

Participants will be asked to participate in iterative rounds of prototyping and testing notebook extensions. Participatory design sessions may involve brainstorming, paper prototyping, video prototyping, review of mockups, or initial testing of fully functional prototypes. The prototyping and development of notebook extensions will be guided initially by the needs observed during the computational ethnography. In particular we will focus on reducing the tension between data exploration and process explanation, as well as conveying where and how professional judgement were employed. Later, extension design will be guided by feedback from the participatory design sessions.

While the selection and design of extensions will be guided by the results of the first two studies and feedback during the design process, a few initial ideas demonstrate the type and scope of modifications we expect to build and test.

One extension might make it easier for analysts to track alternative analyses. Analysts frequently need to test many alternatives, for example testing slightly different model parameter settings. Currently, analysts will either create separate cells to test each setting, cluttering the notebook, or repeatedly test different parameters in one cell, overwriting the history and obscuring the fact that a decision was made about parameter settings that could significantly impact the final finding. One way to more fully document this process without cluttering the notebook is to automatically capture a history of analyses performed in each cell and then let analysts curate that history to only include alternatives that they think are useful to demonstrate the decision. This cell-level history need not be visible at all times, but may be accessed when a cell is active.

Another extension could aim to encourage more consistent documentation of reasoning by presenting a cell that floats on top of the notebook in which analysts can write their current goal, method, assumptions, or interpretation of results. While analysts may not want to pause their analysis to fully document their work, they may be willing to write short phrases in this cell, especially if it can be accessed via a keyboard shortcut. Having this cell pervasively visible can remind analysts of their current goal, and

analysts could use the stream of short comments generated throughout the analysis to go back and document their process more fully at a later time.

As highlighted in the Visual Analytics literature on provenance, a key challenge with tracking the methods and reasoning used to conduct an analysis is to get users to conduct or document their work in ways that will be easy to understand later. This is difficult as analysts may work "at-the-speed-of-thought" and want to perform the analysis in the easiest and fastest way that answers their question, rather than take the time to document or structure it. This is a logical approach as many intermediate results are "dead-ends", and taking time to document these slows the analysis and could reduce the number of hypotheses they can test. A key focus of our design process will be modifying the notebook so that the easy way of working leads to a more understandable notebook down the road, or that taking time to document or structure one's work also provides benefits in the near term.

4.3.4. Analysis and Measures

Once extensions are fully developed, participants will be asked to use them for an extended period of time during their everyday work, and then later interviewed about their experiences using them. We will also recruit additional participants who were not involved in the design of the extensions to test them and provide feedback. The primary results from this study will be a set of working Jupyter extensions and evidence that they enable analysts to perform and share their analytical process more easily. Evidence for the extensions' effectiveness will come mainly from interviews with analysts, and logs demonstrating consistent use of the extensions in their everyday work.

Should we desire more evidence beyond the usage log and interviews, we may run a lab study, such as those performed by [3] or [22], to test if analysts can recall more of their goals, insights, and methods while reviewing notebooks created using notebooks modified with our extensions as opposed to unmodified notebooks. The decision on whether to run these additional analyses will be based on how often analysts end up using the extensions we develop. The impact of an infrequently used extension may be best demonstrated with a lab study.

4.3.5. Expected Results

We expect to find that analysts prefer extensions that help them generate and test a wide range of hypotheses, but then succinctly summarize their process and results. We also expect to find that analysts prefer to use extensions that provide an immediate benefit, and not just aid long-term understanding of process. For example, and extension making it easier to manage multiple alternative analyses may be more widely used if it aids analysts with the immediate task of navigating their codebase by making it easy to expand and collapse whole sections of the notebook. We also expect analysts to benefit from more clearly identifying and stating where professional judgement was employed, such as when selecting how to remove outliers from a dataset.

5. CONCLUSION

Data analysis is increasingly vital to the work of many individuals and organizations. It is an iterative and exploratory process of generating and testing hypotheses. When reviewing, replicating, or revising these analyses, it is often necessary to inspect the steps and reasoning used to produce them. However the tools and techniques analysts use to perform and share their work often fall short of helping them make sense

of this process. This confusion is a consequence of the tension between supporting exploratory analysis and generating clear explanations of the analytical process.

This thesis leverages the advent of computational notebooks as a unique opportunity to study this tension, and investigate how information technology can support the clear communication of analytical process. The proposed research involves studying how analysts conduct and share their work within an academic lab, and online. It also includes analysts in the redesign of tools they use to perform and share their work.

This research will have three main contributions. First, it will characterize how data analysts perform and share their work, both within a laboratory setting and online as well as enumerate the tools and techniques they use to generate clear explanations of their work. Second, it will demonstrate if and how computational notebooks might be redesigned to support clearer explanation of analytical process. It will wrestle with the tension between exploration and explanation as well as that between providing immediate benefit to the analysis or long-term benefit when reflecting on the process later. Insights into how to navigate these tensions may have implications for the design of "computational notebooks" in other domains, such as electronic medical records, and even provenance tracking in data visualization software. The final contribution of this research will be the Jupyter Notebook extensions produced through participatory design with analysts, which we will make available to the 3 million strong Jupyter Notebook community for use in their everyday analyses.

References

1. Bavoil L, Callahan SP, Scheidegger CE, et al. (2005). VisTrails: Enabling Interactive Multiple-View Visualizations. IEEE Visualization.
2. Brodlie K, Poon A, Wright H, et al. (1993). GRASPARC-A Problem Solving Environment Integrating Computation and Visualization. (pp 102–109). IEEE Comput. Soc. Press.
3. Dou W, Jeong DH, Stukes F, et al (2009). Recovering reasoning process from user interactions. IEEE Computer Graphics & Applications, 29(3): 52-61.
4. Dunne C, Riche NH, Lee B, et al. (2012). GraphTrail: Analyzing Large Multivariate, Heterogeneous Networks While Supporting Exploration History. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1663–1672). ACM.
5. Fouse A, Weibel N, Hutchins E, and Hollan JD. (2011). ChronoViz: A System for Supporting Navigation of Time-Coded Data. In CHI'11 Extended Abstracts on Human Factors in Computing Systems (pp. 299–304). ACM.
6. Freire J, Silva CT, Callahan SP, et al. (2006). Managing rapidly-evolving scientific workflows. In International Provenance and Annotation Workshop (pp. 10-18).
7. Gotz D and Zhou M. (2009). Characterizing Users' Visual Analytic Activity for Insight Provenance. Information Visualization 8(1): 42–55.
8. Granger B and Grout J. JupyterLab: Building Blocks for Interactive Computing. [Video] Retrieved from http://scipy2016.scipy.org/
9. Groth DP and Streefkerk K. (2006). Provenance and annotation for visual exploration systems. IEEE Transactions on Visualization and Computer Graphics, 12(6): 1500-1510.
10. Guo PJ. (2012). Software tools to facilitate research programming (Doctoral dissertation, Stanford University).
11. Guo PJ and Seltzer M. (2012). BURRITO: Wrapping Your Lab Notebook in Computational Infrastructure. TaPP, 12, 7-7.
12. Harper R and Sellen, A. (1995). Collaborative tools and the practicalities of professional work at the international monetary fund. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 122-129). ACM.
13. Heer J, Mackinlay J, Stolte C, and Agrawala M. (2008). Graphical histories for visualization: Supporting analysis, communication, and evaluation. IEEE transactions on visualization and computer graphics, 14(6): 1189-1196.
14. Kandel S, Paepcke, A, Hellerstein JM, and Heer, J. (2012). Enterprise data analysis and visualization: An interview study. IEEE Transactions on Visualization and Computer Graphics, 18(12): 2917-2926.
15. Kidd A. (1994). The marks are on the knowledge worker. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 186-191). ACM.
16. Manyika J, Chui M, Brown B, et al. (2011). Big data: The next frontier for innovation, competition, and productivity.
17. Muller, MJ and Kuhn S. (1993). Participatory design. Communications of the ACM, 36(6): 24-28.

18. Nguyen PK, Xu K, Bardill A, et al. (2016). SenseMap: Supporting Browser-Based Online Sensemaking through Analytic Provenance. In IEEE Conference on Visual Analytics Science and Technology.
19. Parker SG and Johnson CR. (1995). SCIRun: A Scientific Programming Environment for Computational Steering. In Proceedings of the 1995 ACM/IEEE Conference on Supercomputing (pp 52). ACM
20. Pike, William A., Richard May, Bob Baddeley, et al. 2007 Scalable Visual Reasoning: Supporting Collaboration through Distributed Analysis. In Collaborative Technologies and Systems, 2007. CTS 2007. International Symposium on Pp. 24–32. IEEE.
21. Ragan ED, Endert A, Sanyal J, and Chen J. (2016). Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes. IEEE Transactions on Visualization and Computer Graphics 22(1): 31–40.
22. Ragan ED & Goodall JR. (2014). Evaluation Methodology for Comparing Memory and Communication of Analytic Processes in Visual Analytics. In Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization (pp. 27–34).
23. Rule A, Tabard A, and Hollan J. (2016). Traces: A Flexible, Open-Source Activity Tracker for Workplace Studies. Computer supported cooperative work workshop on the quantified workplace.
24. Russell DM, Stefik MJ, Pirolli P and Card SK. (1993). The cost structure of sensemaking. In Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems (pp. 269-276). ACM.
25. Scaffidi C, Shaw M & Myers B. (2005). Estimating the numbers of end users and end user programmers. In 2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05) (pp. 207-214). IEEE.
26. Shirokov S. (2015 May 7). GitHub + Jupyter Notebooks = <3. Retrieved from https://github.com/blog/1995-github-jupyter-notebooks-3
27. Shrinivasan YB, & van Wijk JJ. (2008). Supporting the Analytical Reasoning Process in Information Visualization. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1237–1246). ACM.
28. Suchman, L. A. (1983). Office procedure as practical action: models of work and system design. ACM Transactions on Information Systems (TOIS), 1(4): 320-328.
29. Tabard A. (2009). Supporting lightweight reflection on familiar information (Doctoral dissertation, Université de Paris-Sud. Faculté des Sciences d'Orsay (Essonne)).
30. Tabard A, Mackay WE & Eastmond E. (2008). From individual to collaborative: the evolution of prism, a hybrid laboratory notebook. In Proceedings of the 2008 ACM conference on Computer supported cooperative work (pp. 569-578). ACM.
31. Tukey, J. W. (1977). Exploratory data analysis.
32. Van den Haak MJ, de Jong MD & Schellens PJ. (2004). Employing think-aloud protocols and constructive interaction to test the usability of online library catalogues: a methodological comparison. Interacting with Computers, 16(6): 1153-1170.
33. Walker R, Slingsby A, Dykes J, et al. (2013). An Extensible Framework for Provenance in Human Terrain Visual Analytics. IEEE Transactions on Visualization and Computer Graphics 19(12): 2139–2148.

34. Wang W. (2017 January 6) Retrieved from https://github.com/ipython/ipython/wiki/A-gallery-of-interesting-IPython-Notebooks
35. Xu K, Attfield S, Jankun-Kelly TJ, et al (2015). Analytic provenance for sensemaking: A research agenda. IEEE computer graphics and applications, 35(3): 56-64.
36. Zheng K, Hanauer DA, Weibel N, and Agha Z. (2015). Computational Ethnography: Automated and Unobtrusive Means for Collecting Data In Situ for Human–Computer Interaction Evaluation Studies. In Cognitive Informatics for Biomedicine (pp. 111-140). Springer International Publishing.
37. API. Retrieved from https://developer.github.com/v3/
38. RNA-Popgen-Notebook from http://github.com/gocarli/RNA-Popgen-Notebook/blob/master/Population_Genetics.ipynb