Code Code Evolution: Understanding How People Change Data Science Notebooks Over Time

Deepthi Raghunandan
University of Maryland
College Park, Maryland, USA
draghun1@umd.edu

Aayushi Roy
University of Maryland
College Park, Maryland, USA
aroy2530@umd.edu

Shenzhi Shi
University of Maryland
College Park, Maryland, USA
sshi1234@umd.edu

Niklas Elmqvist
University of Maryland
College Park, Maryland, USA
elm@umd.edu

Leilani Battle
University of Washington
Seattle, Washington, USA
leibatt@cs.washington.edu

ABSTRACT

Sensemaking is the iterative process of identifying, extracting, and explaining insights from data, where each iteration is referred to as the "*sensemaking loop*." However, little is known about how sensemaking behavior evolves from exploration and explanation during this process. This gap limits our ability to understand the full scope of sensemaking, which in turn inhibits the design of tools that support the process. We contribute the first mixed-method to characterize how sensemaking evolves within computational notebooks. We study 2,574 Jupyter notebooks mined from GitHub by identifying data science notebooks that have undergone significant iterations, presenting a regression model that automatically characterizes sensemaking activity, and using this regression model to calculate and analyze shifts in activity across GitHub versions. Our results show that notebook authors participate in various sensemaking tasks over time, such as annotation, branching analysis, and documentation. We use our insights to recommend extensions to current notebook environments.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; **Empirical studies in HCI**; • **Mathematics of computing** → **Exploratory data analysis**.

KEYWORDS

Computational notebooks, machine learning, sensemaking, data science, data exploration, analysis.

ACM Reference Format:
Deepthi Raghunandan, Aayushi Roy, Shenzhi Shi, Niklas Elmqvist, and Leilani Battle. 2023. Code Code Evolution: Understanding How People Change Data Science Notebooks Over Time. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3544548.3580997

CHI '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9421-5/23/04.
https://doi.org/10.1145/3544548.3580997

1 INTRODUCTION

Sensemaking is "the process of searching for a representation and encoding data in that representation to answer task-specific questions" [34]. In each iteration of this "sensemaking loop" [28], data scientists refine their code, visualizations, and annotations in pursuit of a deeper understanding of their data [36]. Pirolli and Card posit that data scientists often oscillate between *exploring* the data and *explaining* what they have learned (to themselves or stakeholders) during sensemaking [28], leading to more of "spiral" of activity than a true "loop". Rule et al. observe similar sensemaking patterns among computational notebook users [33]. Furthermore, sensemaking is considered an *iterative* process, where the explanation or insight of each oscillation informs the next [28, 33].

Computational notebooks such as Jupyter [17], R Markdown, or Observable are especially popular for documenting the complexities of the sensemaking process given the ease with which code can be interleaved with descriptive text and illustrative images [16, 33]. However, notebooks still fall short of the ideal for sensemaking, particularly in tracking changes to notebooks over time [13], frustrating many notebook users [4].

In order to improve notebooks for sensemaking we must first characterize users' common interaction patterns so that we can (re)design notebook environments to better support them [13]. However, the evolving nature of sensemaking suggests that these patterns may vary depending on where users are within the progression between *exploration* and *explanation* [33]. Thus, we need to determine where a user is along this exploration-explanation spectrum before we can design appropriate solutions. Recent work posits that we can infer where a user is within the exploration-explanation spectrum directly from computational notebooks [16, 33, 42]. However, these prior works rely on small-scale user studies to investigate sensemaking within notebooks. Furthermore, they treat notebooks as static outputs of sensemaking rather than a core medium for iteration. This limits our understanding of notebooks as living documents of scientific inquiry. Without more rigorous validation, it is still unclear whether current theory can accurately detect sensemaking within real-world notebook environments.

This paper proposes a new approach to analyzing how computational notebooks are revised over time. The key idea is that many analysts already track their notebook iterations using public version control infrastructure such as GitHub. We contribute a pipeline to collect, model, and quantify exploration and explanation

across GitHub commits. This pipeline allows us to (1) characterize observed shifts in sensemaking behaviors within notebooks, such as whether notebooks become more explanatory or exploratory over time, and to (2) understand why these shifts occur. To do this, we randomly sampled and downloaded 2,574 Jupyter notebooks stored on GitHub's public repositories. We report on their content, revision history, and evolution. Our analysis has three parts:

(1) **Identifying Relevant Notebooks** – finding the data science notebooks that were actively refined overtime on GitHub, as well as quantitative metrics to analyze them;
(2) **Measuring Exploration vs. Explanation** – leveraging prior work [16, 33] to distinguish between the exploratory vs. explanatory nature of data science notebooks; and
(3) **Measuring Evolution** – drawing on GitHub revision history to understand how notebooks, and in turn authors' positions in the sensemaking loop, shifted over time.

We acknowledge that these quantitative approximations may not reflect the author's complete process. This discrepancy is due in part to authors' selective reporting as well as to limitations inherent to notebook platforms themselves [4, 16, 21, 24, 33, 39, 42]. In other words, the use of GitHub as a data source likely biases the type of notebooks we collect for our sample. Regardless, we believe that measuring notebooks as they publicly change over time still provides a unique perspective on the sensemaking process that qualitative analyses of singular notebook versions cannot.

We make the following contributions in this paper: (1) we develop a rubric to show how to quantify the explanatory or exploratory nature of a Jupyter Notebook, enabling us to analyze data-science notebooks at scale; (2) we track the evolution of notebooks over time by calculating our quantitative measure across multiple notebook versions; (3) we characterize the way analysts iterate on their notebooks; and (4) we use these insights to make design recommendations to better support the different notebook-based sensemaking behavior we observed. More broadly, we contribute a more nuanced view of the data science process that brings notebook analysis methodology closer alignment with established theories of sensemaking and data exploration. This quantitative approach to understanding the analytical process can be directly applied towards teaching, guiding, and developing tools for promoting best practices in data science.

Beyond the overview of our method and results presented in this paper, we have also provided supplementary material with the full details in the following OSF repository : https://osf.io/9q4wp/?view_only=61e6f58d29194742a0aaed328afdea4d

2 RELATED WORK

In this section, we introduce key concepts and terminology that we use in our work to map signs of exploration and explanation in computational notebooks to corresponding shifts within the sensemaking loop.

2.1 Computational Notebooks

As an embodiment of the literate programming paradigm [18], where traditional source code is embedded in descriptive natural language, *computational notebooks* are an ideal medium for studying the sensemaking process [7, 17, 26, 29, 35, 44]. A notebook is a linear sequence of executable code that perfectly captures the procedural nature of sensemaking. The ability to inspect intermediate results by generating visualizations and tables scaffolds the exploratory process. The rich annotation features scaffold the pivoting of data representations towards explanation. Notebooks also allow for easy sharing of data, code, and analyses all in one [4, 33]. As a result, computational notebooks have quickly become an essential part of conducting data science [16, 17, 33, 39].

Data scientists utilize computational notebooks—specifically their flexible cell structure—to iterate on different branches of exploration and create narratives surrounding their analyses [4, 5, 11, 14, 16, 19, 27, 31, 33, 40–43].

2.2 Sensemaking in Computational Notebooks

Pirolli and Card describe the sensemaking loop as cumulative iterations by which analysts develop an understanding of the data [28]. Each iteration informs the next. Our work enriches this definition of sensemaking with notebook-oriented notions of exploration and explanation from the literature [16, 33]. Specifically, we define the "sensemaking spectrum" as a two-dimensional representation of the sensemaking loop from early-stage exploration to late-stage explanation [28].

Qualitative studies from Rule et al.[33], Kery et al. [16], and Wang et al. [42] show that we can observe sensemaking within computational notebooks in the form of exploration and explanation. For example, Kery et al. observed that some analysts created many small code cells while performing exploratory data analysis to optimize iteration, and later grouped code into individual "logical units" to communicate analytical steps [16]. Rule et al. noted that analysts place text that serve different purposes in different parts of the notebook [33]. They found that nearly all code comments help explain the methods employed by code, headers labeled the analyses, and most non-header text explained analytical steps. Wang et al. extended this finding by showing that highly readable (explanatory) notebooks use a variety of descriptors to attract a broader audience [40]. Based on these findings, it seems evident that different notebook characteristics, such as types of documentation or distribution of code across cells, can indicate an analyst's current position within the sensemaking spectrum between exploration and explanation.

Some previous work contribute to our understanding of how notebooks are used but do not identify these steps in the context of sensemaking. For example, Dong et al. find that code cleaning is an integral part of sharing a notebook [5]. They characterize cleaning as renaming variables, generating functions, reordering code cells, adding pertinent annotations, moving content between files, and removing extraneous content. We leverage Dong et al.'s work to construct a comprehensive model of sensemaking in notebooks.

2.3 From Exploration to Explanation

In data *exploration*, analysts seek to profile their data, define their goals, and become comfortable with potential analytic methods [8, 16, 27]. As Alspaugh et al. explain, data analysis exists within a spectrum between "exploratory" and "directed" analysis, wherein the nature of analysis changes as goals become more concrete [1]. Analysts seek to understand their dataset, look for exciting patterns,

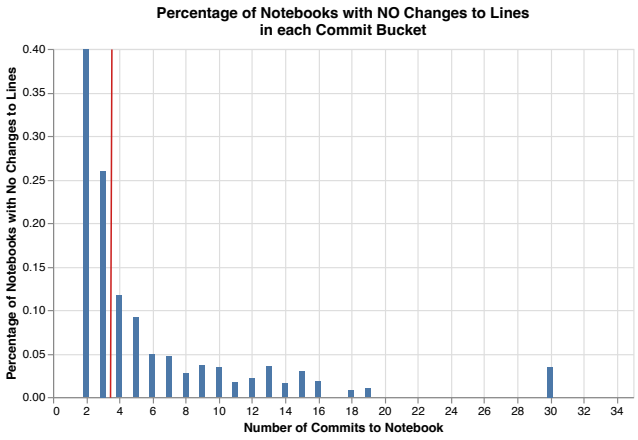

Figure 1: Thresholding on Number of Notebook Commits. We grouped notebooks by the number of times they were committed to GitHub. We note the percentage of notebooks within each commit group wherein the number of lines changed between the first and last commits. As the vertical red line indicates, we filter out notebooks in commit groups where more than a quarter of the notebooks showed zero changes to lines (i.e. notebooks with fewer than 4 commits).

and identify assumptions as a means to inform next steps [1, 12, 25, 45]. The ultimate objective of this process is to inform decisions.

The process of *explaining* data insights entails shaping explorations into a narrative to communicate the process and results [16]. This type of explanation provides clarity on how the analysis process yielded particular insights to an audience (including oneself). Analysts can describe their analyses and findings in varying levels of detail and clarity, ranging from reporting all avenues of exploration and ensuing insights to saving only the most critical decisions and findings [16, 22]. The level of detail they choose depends mainly on the audience. When the audience is oneself or fellow technical team members, the analyst focuses on retaining code and branches of exploration and formatting them in a comprehensible manner [16, 33]. When presenting results to a broader, perhaps non-technical, audience, analysts may remove details that appear confusing or uninteresting and add more explanatory text—shifting the focus from the code to the narrative [16, 21, 33].

Our definitions are also grounded in literature on notebook reproducibility—a common motivation for authoring notebooks [4, 27, 31, 42, 43]. Like explanatory notebooks, reproducible notebooks allow for communication, reuse and reproduction—enabling a clear linear structure [31, 43] and presenting clean code [5, 27].

3 DATASET

Given our aim to use revision histories to measure how data science notebooks evolve, we chose to analyze publicly available Jupyter Notebooks found on GitHub for the following reasons: 1) it provides an extensive repository of Jupyter Notebook documents, and 2) they are particularly amenable to meta-analysis due to the ease of accessing their underlying JSON metadata structure. Since GitHub contains a large variety of notebooks in terms of quality and purpose, we anticipated that many notebooks would not be applicable [19, 33] for this analysis. Thus, we sought to programmatically filter for data science notebooks to automatically scale our analysis to any sample size.

In July 2019, we identified 4.7 million notebooks on GitHub, and randomly sampled approximately 10% of this dataset. Of the approximately 400,000 notebooks selected, 27.4% were eliminated due to dead links and an additional 57.6% were eliminated due to inaccessible commit data. We queried for GitHub commit information to ensure we could examine all versions of notebooks. Of the remaining 59,887 notebooks, we selected 2,574 notebooks for further analysis using the criteria we outline below.

3.1 Data Collection Method

To mine Jupyter Notebooks from GitHub, we used Rule et al.'s approach [33]. We first downloaded and accessed the 59,887 notebooks remaining after the original sampling and filtering discussed above. For the sake of project feasibility, we chose to observe only Python notebooks annotated in English. Python and R are the most popular languages used for data science, but we made this choice on the basis that Python is significantly more common in Jupyter than R (more than 96% of all Jupyter Notebooks are written in Python [33]). Using these criteria, we programmatically eliminated 3,642 notebooks from our sample.

To further select notebooks suitable for our analysis, we defined a standard to identify Jupyter notebooks that use **data science** and were **stored on GitHub**. To meet our standards: (1) notebooks must demonstrate data analysis activity, (2) some subset of changes must be observable across multiple versions found in the GitHub repository, and (3) changes to the notebook must be made by the original owner of the notebook. We briefly outline our filtering criteria below; further details about our criteria and our methods can be found in our supplemental material.

Data Science Notebooks. First, we defined data science notebooks as ones that deal with data in any capacity, ranging from loading a dataset to executing numerical operations on the dataset, all the way to developing predictive models. We used the number of popular data science Python libraries as a heuristic to determine whether the notebooks were data science-oriented. We manually derived a list of data science libraries by reviewing 28 online Python tutorials. The full list contained the following packages, in order of popularity: `numpy`, `scipy`, `pandas`, `scikit-learn`, `matplotlib`, `pytorch`, and `tensorflow` [27].

We select notebooks that contain more than two of these libraries and at least 15 function calls to these libraries. We added this last criterion to ensure that data science libraries were imported as well as utilized. The thresholds to these measures were determined using qualitative analysis described in the supplemental materials. We acknowledge that this heuristic has potential for eliminating data science notebooks that use libraries not included in this list. However, this method yielded a definite sample of data science notebooks from the dataset.

Versioned Notebooks. Second, we used the number of notebook versions within the repositories and the number of changes within

them as a heuristic to measure whether changes to data analysis were observable. Initially, we considered notebooks with at least two versions. Through qualitative analysis, we found that a threshold requiring at least four notebook versions ensured that we would observe substantive changes to notebook content 1.

Among notebooks with at least four versions, we found that selecting notebooks with at least two changes to the number of cells and 20 changes to the number of lines ensured that both notebook content and structure were changed. We did not set an upper bound on any of these thresholds and our goal was to ensure the data science notebooks generally demonstrated changes across versions. We identified these specific thresholds using the qualitative approach described in the supplemental work. A total of 5,082 were eliminated for not meeting this criterion.

Original Content. After observing duplicate notebooks and empty class templates in our sample, we restricted the analysis to versions made only by the original repository owner. This did not eliminate notebooks with multiple contributors and ensured that we considered original work across repositories. An additional 12,350 notebooks were eliminated based on this criterion.

3.2 Summary & Dataset Considerations

A total of 2,574 notebooks remained after the process described above had filtered out dead links, inaccessible data, non-analysis code, and insufficient commit histories. A majority of these notebooks were data-science oriented, contained a rich amount of data science activity, and were written in English.

We acknowledge that our quantitative approach may produce many false negatives (i.e., ignore some valid data science notebooks), providing opportunities to develop looser filters that are still accurate in future research. However, given the millions of notebooks we have access to, we believe false negatives are only a minor concern compared to false positives, i.e., notebooks that include no data science conducted over time, since false positives could pollute the corpus with irrelevant data points that would be difficult to detect automatically. For this reason, we chose rigorous criteria to minimize the number of notebooks in the sample that do not demonstrate traditional sensemaking. We believe that our strict inclusion criteria ensure that very few of the selected notebooks are false positives, making our corpus suitable for large-scale quantitative analyses.

We also acknowledge that not all analysts may commit all their iterations to GitHub repositories. However, given the relatively large size of our dataset (2,574 notebooks and 26,474 notebook versions), the noise in our dataset is significantly diminished.

4 MEASURING EXPLORATION VS. EXPLANATION

We first seek to answer the following research question: *Can we apply previous findings to quantitatively measure exploration and explanation in computational notebooks?* To answer this question, we first manually curated a reference dataset of 244 notebooks (10% of our sample) using a manual rubric that maps notebook characteristics to points within the sensemaking spectrum. This rubric scores builds directly upon findings of previous work regarding how sensemaking manifests within computational notebooks. We then used our manual reference dataset to develop and validate a model to automate the manual classification.

4.1 Constructing the Reference Dataset

We acknowledge that in the absence of feedback from the original notebook authors, it is difficult—and in some cases, impossible—to synthesize an *absolute* measure of the exploratory-explanatory nature of a notebook. We cannot nor should we claim that a specific notebook is forever explanatory or exploratory. However, our interest in this work is to understand how notebooks *evolve* over time, suggesting that a *relative* measure of notebook iteration could be a viable approach to analyzing notebook histories. In other words, with a substantial history of notebook iteration, we can quantify how sensemaking *shifts* over time, rather than trying to pinpoint exactly where sensemaking begins or ends. Given our extensive filtering strategy in the previous section, we know that our corpus provides rich records of notebook revisions, enabling a comparison-based analysis approach.

With this in mind, we introduce a measure of the exploratory or explanatory nature of an individual notebook version, which we will use later to perform a before-and-after analysis across notebook versions. In this section, we describe a new rubric to score data science notebooks according to their position along the sensemaking spectrum, which we aim to automate. The goal of this analysis was to determine whether each notebook appears to be more exploratory or explanatory in nature, but again, in service of our larger goal of analyzing notebook *shifts* over time.

Our rubric development was guided by existing evaluations of sensemaking within computational notebooks. For example, as observed in prior work [16, 33, 40], notebooks with good narrative structure generally tell a compelling story of both the data analysis process and the insights derived from this process. These notebooks clearly communicate the analyst's motivations and insights, and can appeal to a wide audience via instructional text or explanations of field-specific terminology [1, 4]. Therefore, the better the narrative structure of a notebook, the higher the exploration-explanation score it should receive.

Based on the literature, one of the authors developed a rubric for scoring notebooks. Two other authors provided feedback on the rubric between coding iterations. All coders were knowledgeable in data science. Inspired by methods for reaching agreement on qualitative codes [23], our scoring process involved three iterations with two coders to converge to scores consistent with our rubric. The iterations included a preliminary scoring iteration where scores were assigned based on an initial rubric, a second one where scores were refined in parallel with the rubric, and a final one where scores were reviewed for consistency with the final rubric. Coders reached a Krippendorff's Alpha inter-rater reliability score of 0.88 after converging on this rubric.

We represented the positions of notebooks on the sensemaking spectrum using a score between 0.1 and 1.0, where 0.1 represents the most exploratory notebooks and 1.0 the most explanatory ones. Scores were assigned in increments of 0.1. We wanted our range of scores to be evenly distributed such that the first five scores (0.1-0.5) characterize mostly exploratory notebooks and the last five scores (0.6-1.0) characterize mostly explanatory notebooks.

Score	Stage	Concrete Examples
0.1	Understanding the data [1, 12, 25, 28, 45]	Just code (Few unorganized cells [16, 33], duplicated code [11], no output [27].)
0.2	Iterative data wrangling [1, 8, 12, 25, 28, 45]	Just code, some output from exploration [27] (disjoint, duplicated code cells; lots of code cells with individual lines of code [4, 9, 16, 33]).
0.3	Defining goals using iterative exploratory analysis [1, 12, 28, 45]	Lots of code [16], some output from exploration (some code cells are grouped by functionality, text headers and code comments are used to label groups) [4, 5, 9, 16, 33].
0.4	Beginning goal-oriented exploratory analysis [1, 12, 28, 45]	Code and visual output address some goals (some code cells are linearly grouped by text headers and code comments are used to label and annotate analysis) [5, 9, 16, 31, 33, 42, 43].
0.5	Exploratory analysis with clear goals [1, 12, 28, 45]	Code and visual output address goals (majority of code cells are linearly grouped by text headers and code comments) [5, 9, 16, 31, 33, 42, 43].
0.6	Analytical steps are communicated [1, 8, 12, 28, 45]	Code and code output are interwoven with text headers and code comments for the sake of outlining the logical steps which were taken. [9, 16, 33, 40].
0.7	Some insights of analysis tracked and communicated [1, 8, 12, 28, 45]	Analysts are also using text to briefly annotate their code with insights from individual logical steps. [16, 33, 40].
0.8	Some motivations and insights tracked and communicated [1, 8, 12, 28, 45]	Analysts are explaining their motivations behind individual analytical steps, and more thoroughly annotating their logical steps with insights. [16, 33, 40].
0.9	Motivations and insights of analysis clearly communicated [1, 8, 12, 28, 45]	Analysts introduce their analytical reasoning behind the work overall. In addition, they are illustrating links between logical steps using text in the form of headers, insights and motivations. Together the text forms a narrative of the methods and the results. [16, 33, 40].
1.0	Analysis workflow communicated to a wide audience [28]	Text may provide instructions on how to interact with the notebook, provide context behind the work, motivations on the methodology, insights from individual logical steps and insights from the entire exercise [32]. If code and code output are present, they align with the narrative being outlined by the text. [16, 33, 40].

Table 1: Notebook scoring rubric. This rubric leverages existing observations from the literature to characterize notebooks along the sensemaking spectrum. For example, Tukey's definition of exploratory data analysis motivates our definition of stages 0.1 - 0.5 [38], and we defined stages 0.6 - 1.0 using existing definitions of narrative structure [33] and types of descriptors [40]. Our supplementary material (https://osf.io/9q4wp/?view_only=61e6f58d29194742a0aaed328afdea4d) includes the full rubric.

To form an impression of the notebooks, coders considered the following criteria:

- Code abstraction methods such as functions, classes, and code distribution within cells;
- The clarity of code based variable names and in-line code comments;
- The use of markdown headers to create sections;
- The cohesiveness of the analytical workflow; and
- The types of descriptors (analytical, procedural, and context) included in the document.

We identify exploratory notebooks as ones which leverage code to explore data. These notebooks place little focus on explaining insights, reasoning, or analytical methods, and instead focus on fast iteration resulting in duplicated, messy code [4, 16, 19]. In explanatory notebooks, on the other hand, the intent to outline, document, or explain previous data exploration is clear [1, 4, 33, 40]. The rubric in Table 1 specifies how each notebook was evaluated.

4.2 Automating the Scoring Process

To scale up our analysis, we needed a way to programmatically calculate a notebook's exploration-explanation score. We observed characteristics of notebooks from our reference dataset to understand how they contributed to a notebook's position on the sensemaking spectrum. We chose to observe particularly quantitative characteristics which were highlighted by previous literature on sensemaking in notebooks and used our own observations to understand how other metrics correlated with our manually assigned exploration-explanation scores.

4.2.1 Analyzing Notebook Characteristics. Prior work suggests that authors change many different aspects of a notebook throughout the sensemaking process. Specifically:

- The amount of code in a notebook tends to increase as analysts explore their dataset [4, 5, 9, 11, 16, 19, 27, 33].
- Data science notebooks often contain content beyond just code. This content is of particular interest, because they are

Focus	Code Related Measures	Non-Code Related Measures
Output		*Number of tables produced by code cells* *Number of visualizations produced by code cells* *Number of text outputs produced by code cells*
Organization	*Number of code cells* Number of lines of code across code cells	*Number of spacing characters across markdown cells* Number of lines of text across markdown cells
Output & Organization		*Number of markdown cells*
Other	Number of spacing characters across code cells Number of individual code comments within code cells	

Table 2: Summary of metrics. Our Hybrid combination included the above measures in italics. To ensure that different quantitative measures could be compared fairly across notebooks, we normalized each measure with respect to individual notebooks. Our normalization process translates each measure to a domain of 0.0 to 1.0. We normalized cell counts by dividing them by the total number of cells found in the notebook (e.g., number of code cells divided by the number of all cells). We normalized the output (text, table, and visualization) counts by dividing them by the total number of outputs in the notebook. We normalized the number of code comments by the number of lines found within the code cell. Finally, we normalized the number of spaces for a given cell type by dividing by the total spaces across all notebook cells.

primarily explanation-oriented, and thus increase a notebook's exploration-explanation scores [16, 32, 33, 41].
- Negative space can have a profound impact on how information is organized and presented for communication [37]. Rule et al. suggest that the number of spacing characters in text and code cells could point towards more of an explanation focus for a notebook [33].

For these reasons, we analyzed all parts of a notebook, including code, non-code, and whitespace, when developing and applying the rubric.

4.2.2 Combining Measures. We combined a subset of these measures into three groups named "Output-Focused," "Organization-Focused," and "Hybrid."

- The "Output-Focused" group focused on the outputs generated by a notebook, which may indicate a more explanatory notebook.
- The "Organization-Focused" group of measures gauged the proportions of different cell types and their structure, where notebooks with more markdown cells and/or better organized cells were likely to be more explanatory.
- It is possible that cell outputs and cell organization together play important roles in assessing the exploration-explanation scores of a notebook. In response, we formulated a new "Hybrid" combination, incorporating measures from both of the above combinations. We refer to this combination as "Hybrid."

We provide an itemized list of measures and their organization in Table 2.

4.2.3 Comparing Combinations of Quantitative Measures. We used each combination of measures as parameters in a multi-linear regression analysis against the manually assigned exploration-explanation scores. We leveraged a k-fold cross-validation technique to ensure the strength of each model. The models were trained in 5 folds on 20% of the data, and tested on the rest. R^2 values were calculated for each model, within each fold. A mean and median R^2 value were generated for each model. Median R^2 values were compared to assess correlations.

4.2.4 Hybrid-Focused Combination Performance. We found that our Output-Focused and Organization-Focused combinations correlate positively with increases in exploration-explanation score. The Organization-Focused combination has a higher correlation value than our Output-Focused combination.

However, it is unclear whether cell outputs and cell organization measure redundant information, or are complementary. To assess this relationship, we performed the same analysis with our "Hybrid" combination, which produces a multi-linear regression model with a correlation value ($R^2 = 0.591$): $Y = 0.426 \times \texttt{totalMarkdownCells} + 0.145 \times \texttt{totalMarkdownSpace} - 0.077 \times \texttt{totalCodeCells} + 0.176 \times \texttt{totalVisualizations} + 0.125 \times \texttt{totalTextOutputs} + 0.172 \times \texttt{totalTableOutputs} + 0.395$.

Thus, it seems that cell types, cell outputs, and cell organization capture separate but complementary facets of a notebook author's sensemaking process. For this reason, **we use the hybrid combination for all subsequent analyses** in this paper. The results for the "Hybrid" combination are provided in Figure 2, where the y-axis represents the range of automated exploration-explanation scores, and the x-axis the manually assigned exploration-explanation scores.

4.3 Results and Takeaways

In this study, we used prior observations of how analysts interact with the sensemaking process as a whole [1, 12, 16, 20, 28, 34] and computational notebooks in particular [33] to derive a new rubric for assigning a exploration-explanation score to an individual notebook. Our scoring mechanism aims to establish sufficient placement along the sensemaking spectrum such that we can observe shifts in

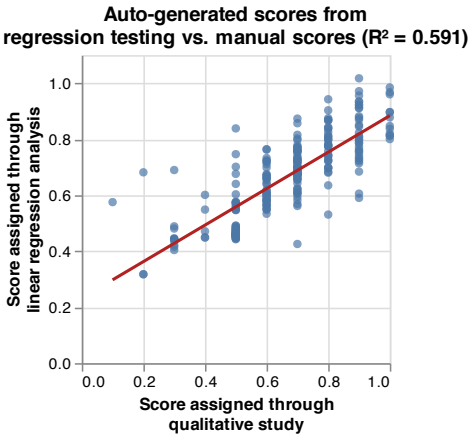

Figure 2: Score comparison. Comparison of our manually assigned exploration-explanation scores (x-axis) and hybrid automated scores (y-axis), using a combination of quantitative measures. These data points are drawn from our reference dataset containing 244 notebooks. (Multi-linear regression model, $R^2 = 0.591$.)

the spectrum over time. Although not exact, our computed scores still provide a valuable signal for studying notebook evolution.

To scale up the application of our rubric, we first formed a manually-labeled reference dataset containing 244 notebooks. These were used to develop our regression models. We then analyzed relevant quantitative measures that may predict these exploration-explanation scores. We found a strong correlation between increases in exploration-explanation scores and increases in organization-focused measures, such as having more lines or more spacing in markdown cells. We also found a positive correlation between output-focused measures and exploration-explanation score (i.e., more explanatory notebooks). Thus, the content and structure of a notebook may be indicative of the sensemaking goals of the notebook's author.

Analysis Limitations. We acknowledge that it is unrealistic to extract exact quantitative measures for exploration-explanation scores and that many combinations of notebook attributes could ultimately predict sensemaking behavior. Furthermore, although we believe that our R-Squared value indicates that the metrics we considered can be used to measure sensemaking activity in notebooks, there is room for improvement towards *modeling* sensemaking. For example, modeling cell relationships could help capture the narrative structure of the notebook. We could derive these relationships by implementing more semantic analysis techniques that track the execution order and the inter-dependence between code cells. Furthermore, natural language techniques could be used to identify the depth and breadth of documentation as well as their relevance to the techniques attempted.

That being said, these techniques and capabilities are considered in retrospect. At first, it was unclear what techniques could be applied appropriately to this data, and to what degree, since few if any works have taken this approach to quantifying the sensemaking process in computational notebooks. Thus, we position this paper as an exploratory "first look" into how these kinds of analyses can be conducted programmatically, and leave it to future work to generalize and extend them. We encourage the community to extend our initial feature set with new attributes, and we hope that our findings can inform future goals for developing accurate and realistic semantic models in the future (see Section 6).

5 MEASURING NOTEBOOK EVOLUTION

Rule et al. observe that "the process used to collect, explore, and model data has a significant impact on the sense made." In other words, the *process* of authoring a notebook affects the insights derived. Given that a single snapshot of a notebook represents only one point within this process, it stands to reason that analyzing only one version of a notebook is insufficient to fully comprehend the sensemaking process behind it. For example, it is impossible to know from a single notebook version whether a user's analysis *shifted* from exploration towards explanation, as hypothesized in prior work, or followed a different path.

However, a more complete view of the user's sensemaking process could be gained by considering how the notebook has changed over time, i.e., across multiple git versions. To this end, we analyze how our exploration-explanation scores from Section 4 change across notebook versions by treating them as individual time series. We seek to answer the following research question through this analysis: *How does the exploration-explanation score of a notebook change over time, and what factors (if any) may explain any observed changes in the score?*

5.1 Measuring Exploration-Explanation Scores Across Versions

To understand how notebooks change over time, we chose to characterize notebooks by their respective and available versions. This was done in two steps.

First, we used public GitHub commits as a proxy for notebook versions. We downloaded all available GitHub versions for each notebook. For each version, we generated the notebook metrics needed to apply the "hybrid-focused" formula from Section 4.2.4. We used these metrics to calculate the exploration-explanation score of each version, compiling a list of scores for each notebook. This transformation allowed us to view each notebook as a time-series of exploration-explanation scores (i.e., a series of notebook scores ordered based on the time of each commit). For example, we would represent a notebook with 5 versions with a list of 5 numbers, each ranging from 0.1 to 1.0. Each number indicated the position of each version within the sensemaking spectrum. We viewed changes in the time-series to indicate the evolution of a notebook across versions.

Second, we normalize the time-series data to enable comparison across notebooks. The number of versions and thus the length of our representations varied across notebooks, ranging from 4 to 94 versions. To do this, we generated a simple, best-fit linear regression model for each notebook representing points as a linear relationship between version numbers and exploration-explanation scores. A linear model is an appropriate choice because we focus on general shifts across entire notebook histories, which is a noisy time series.

Linear Models. Many time series exhibit different patterns at different levels of granularity [10], where some of the observed variation may be due to noise [2]. The stock market is a classic example. The gyrations of the stock market vary non-linearly at a granular level, but a linear model can overcome the effects of noise to reveal overall trends of stock market prices over time, e.g., market booms and busts and phenomena such as "regression to the mean" [2]. A linear model is simple but still appropriate for assessing these kinds of trends in noisy time series [6].

We also attempted to analyze this data using more sophisticated time series analysis methods such as dynamic time warping. However, we soon realized these methods were unsuccessful due to noise; example time series are shown in Figure 3. We observed consistent overall shifts across time series, but no consistent patterns between consecutive pairs of commits. Hence we adopted a more traditional time series analysis method, i.e., a linear model [6].

5.2 Grouping Time-Series

Now that we had a means of comparing notebook time-series, we chose to group notebooks by major shifts in exploration-explanation score as a way to identify common user behaviors. We wanted to asses whether these behaviors matched our current understanding of the sensemaking spectrum. For example, if users generally follow the pattern hypothesized in prior work [5, 9, 13, 16, 33], then we would expect to see notebooks shifting upward from exploration towards explanation. However, the notion of a sensemaking loop suggests that users might also do the reverse, corresponding to shifts from explanation toward exploration. In the remainder of this section, we describe our process for grouping the time series and qualitatively analyzing each group, and discuss the major shifts that users tended to make along the sensemaking spectrum.

Grouping Methods. We focus our analysis on how notebooks *shift* along the sensemaking spectrum, represented by three variables: initial score (i.e., time-series starting with high/low exploration-explanation scores), final score (i.e., time-series ending with high/low exploration-explanation scores), and direction of slope from respective linear regression model (i.e., increasing or decreasing scores). Using the rubric established in section 4, we labeled scores ≤ 0.5 as exploratory and > 0.5 as explanatory.

We identified four main groups of notebook shifts: exploration to exploration, exploration to explanation, explanation to explanation, and explanation to exploration. For example, time-series that began and ended with exploratory notebook versions were grouped as 'exploration to exploration.'

Qualitative Analysis Methods. Three of the authors qualitatively examined a random sample of 5% of all notebooks (142 total) and their version histories using the following guidelines:

(1) We analyzed the first version to form a hypothesis for the analyst's initial intent in creating the notebook.
(2) We observed changes in the type of text, code, and visualizations across individual version deltas and how these changes contributed to the notebook's narrative.
(3) We paid special attention to changes in the structure of the notebook across versions—e.g., markdown, comments, or visualizations demarcating different analytical steps.
(4) The frequency of commits, the commit window, and the commit messages gave our coders clues into how authors leveraged GitHub to meet their analysis goals.

We derived qualitative codes (words or short phrases) to describe our observations with respect to these guidelines. We used these codes to identify broader behavioral themes within each of the four sensemaking groups. Themes focus on structural elements commonly used to track the narrative and flow of sensemaking, including code comments, objectives, sections, templates, and cleaning [33]. Details are provided in our supplementary material.

5.3 Results

Here we discuss our observations for each group of sensemaking shifts, summarized in Table 3: exploration to exploration, exploration to explanation, exploration to explanation, and explanation to exploration.

5.3.1 Exploration to Exploration. 22.6% of our sample contains notebooks that begin as exploratory (0.31-0.49) and remain exploratory after subsequent changes (scores of 0.31-0.49). Notebooks in this group tend to have a relatively flat slope, suggesting "slow" progress along the sensemaking spectrum. Although they remain exploratory, we still observe both positive (towards explanation) and negative (towards exploration) shifts within this group.

Edit Behavior. Authors of these notebooks often depend on *code comments* to organize, annotate and save code [13, 16, 33]. Code is commented as a way to control the flow of the analysis [13, 16, 33]. Authors also add text within code comments to label analyses and describe insights. These notebooks organize code based on their purpose. Code in loops and functions are sometimes found in separate cells from code that outputs text, tables, or visualizations. Code that outputs are generally found in smaller sections to facilitate quick iteration [16]. The edit behavior within negatively and positively sloping notebooks are the same. Negatively sloping notebooks often capture the removal of visualizations and positively sloping notebooks their additions.

5.3.2 Exploration to Explanation. 15.1% of our sample contains notebooks that begin as exploratory (scores of 0.31-0.49) and become explanatory (0.50-0.91). These notebooks had a relatively steep slope, which could be interpreted as "rapid" shifts in sensemaking. They tend to start within a narrow range of exploration scores and end within a wider range of explanatory scores.

Edit Behavior. The first versions of these notebooks typically contain just code or code and visualizations. In subsequent versions, there are two main methods of iteration authors employ. The first method consists of adding markdown and code in tandem, such as including annotations and headers into sections as they create and edit code cells. In the second method, authors focus on code iteration first, and add markdown and headers in their last few commits [5, 16, 33]. Some authors explicitly label a *cleaning phase* within their GitHub versions where they prep their notebook for communication purposes. This cleaning phase often involves reordering, splitting, and reformatting code cells as well as adding observations within markdown cells [5, 9].

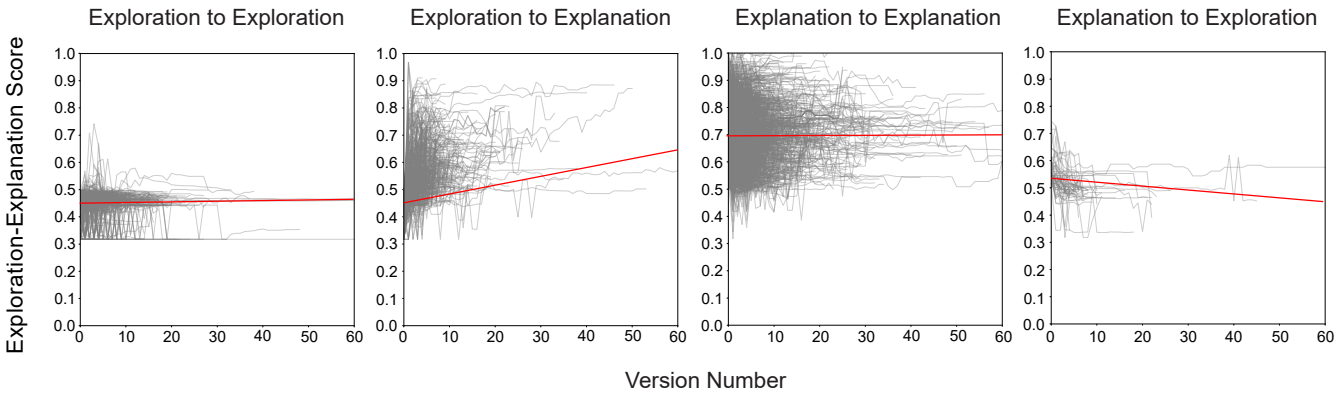

Figure 3: Visual summary of notebook time-series. Summary of trends within each notebook group, in order: exploration to exploration, exploration to explanation, explanation to explanation, and exploration to exploration. The red line represents the average linear trend.

5.3.3 Explanation to Explanation. 60.1% of our sample contains notebooks that begin as explanatory notebooks (scores of 0.50-1.00) and remain explanatory after changes (0.50-1.00). Although both positive and negative shifts are observed within this group, these notebooks tend to have relatively flat slopes, similar to the 'exploration to exploration' group. This is by far the largest group observed, suggesting that data scientists may prioritize clarity and reproducibility during sensemaking within GitHub notebooks, consistent with prior work [5, 9, 16, 27, 33].

Edit Behavior. Notebooks that became less explanatory (sloped negatively) often began with a *template, to-do list, or a statement of objective* at the top of the page. In other words, they started as highly explanatory, which we see reflected in these notebooks' first scores, averaging 0.7. Notebooks that became more explanatory (sloped positively) lacked an explicit statement of objectives. Objectives, implicit (in positively sloping notebooks) or explicit (in negatively sloping notebooks), seemed to drive the construction of the rest of the notebook. For example, if a template specified three goals, we observed authors attempt each goal sequentially across versions. Some authors even added commit messages about the goal being achieved. When authors implemented each goal, they often added annotations to describe and explain their workflow as it progressed. For example, if authors added code and visualizations to the end of the notebook, they also added markdown text to describe their process and results.

5.3.4 Explanation to Exploration. Perhaps not surprisingly, only 2.09% of notebooks started explanatory (scores of 0.50-0.74) and became exploratory (0.33-0.49). Relative to other groups, these notebooks shift negatively from within a narrow explanatory range to a narrow exploratory range.

Edit Behavior. These notebooks progress towards exploration through the removal of explanatory elements. For example, several notebook authors commented code producing visualizations and deleted markdown cells in later versions. This reduction of visualizations and markdown in favor of code may be indicative of authors preparing for new iterations of sensemaking with existing (and likely duplicate) code as a starting point [13, 19].

5.4 Summary

Our qualitative findings suggest that GitHub commits can capture shifts in notebook editing behaviors over time, which we successfully mapped to corresponding shifts in the authors' sensemaking. Thus, our results support the idea that one can *automatically* detect a variety of sensemaking activities within computational notebooks.

Although we do see the shift from exploration to explanation emphasized in prior work [5, 9, 16, 33], our analysis also reveals a variety of shifts along the entire sensemaking spectrum which were previously unobserved.

- It appears data scientists explain their findings *in tandem* while exploring their data, as seen through our analysis of the "explanation to explanation" group of notebooks.
- Furthermore, the "exploration to exploration" group shows that some notebooks have yet to reach the explanatory stage, suggesting that some authors are content to keep certain analyses or notebooks exploratory.
- We also observed shifts *away* from explanation towards exploration. Though previously unobserved (and in some ways, counter-intuitive), this result is consistent with our understanding of the sensemaking spectrum. We speculate that this behavior demonstrates the beginning of a new sensemaking iteration.
- The fact that some notebooks start with explicit objectives suggests that authors begin these notebooks with prior knowledge and analysis goals, and likely leverage them to streamline their analysis of the data. Put another way, analysis experience may allow notebook authors to "short-circuit" the traditional sensemaking loop.

6 DISCUSSION

We have presented an analysis of 60,000 Jupyter Notebooks and their respective GitHub histories. With this corpus, we isolate 2,574 notebooks that appear to be data science-oriented, characterize

	Explore-Explore	Explain-Explain	Explore-Explain	Explain-Explore
# of notebooks	582	1549	390	54
% of sample	22.6	60.1	15.1	2.00
avg # of versions	9	10	10	11
avg first score	0.438	0.683	0.438	0.567
avg last score	0.453	0.695	0.618	0.469
avg slope value	0.002	0.0015	0.021	-0.012
% positively sloping	58.4	54.4	96.9	5.55
% negatively sloping	38.3	45.5	3.00	94.4
% neutral sloping	3.20	.06	0	0

Table 3: General statistics of each notebook group. Explore signifies that the score (first-last) in the time-series corresponding to the notebooks are in the exploratory side of the sensemaking spectrum. Explain signifies that the score is in explanatory side of the spectrum.

their organization and structure, quantitatively measure various properties to situate them within the overall sensemaking process [28], and observe how sensemaking within these notebooks shifts across GitHub commits.

6.1 Explaining and Generalizing the Results

First, our results demonstrate that we can apply qualitative observations from the literature (e.g., [5, 9, 16, 33, 40, 41]) to *automatically* measure sensemaking within Jupyter Notebooks. We found that a linear combination of quantitative measures involving both output types (e.g., how many visualizations are generated?) and organization (e.g., how much negative space is incorporated?) correlated with the scores in our reference set. These findings suggest that the presence of descriptive outputs such as visualizations and text, as well as text formatting with negative space, are signals for sensemaking in notebooks.

Second, by taking a mixed-methods approach to measuring how each data science notebook evolves across multiple GitHub commits, we showed that we can estimate how notebooks change over time. We can automatically detect a variety of sensemaking activities within computational notebooks: sustained exploration, shifting from exploration to explanation, sustained explanation, and shifting from explanation to further exploration. We validated our observations of quantitative score shifts through qualitative observations of the corresponding notebook edits, which reveal consistent patterns of notebook editing behaviors associated with these shifts. As Pirolli and Card describe, analysts appear to exhibit a cycle of sensemaking activities.

Third, our findings also reveal a range of distinct notebook edit behaviors. During exploration, notebook authors leverage code comments to control, organize, and annotate their code flow. Cells are leveraged to enable rapid iteration and create a separation between functionally different snippets. For example, when developing explanatory notebooks, authors can choose to add explanatory elements such as markdown and visualization in tandem or during a "cleaning" phase. These explanatory elements often point to either implicit or explicit goals being set for the analysis. Some notebook authors choose to remove explanatory elements like markdown cells or visualizations as they iterate. This may be suggestive of a change in the authors' objectives.

These behaviors align with existing observations of data exploration behaviors using visualization tools [3]. Observed parallels between notebook editing behaviors and visual analysis behaviors suggest that there are core patterns to sensemaking that transcend particular tools and environments. As a result, our work opens the door to gaining a deeper quantitative understanding of the sensemaking loop itself through the lens of data science tools and practices. For example, our findings could inform the design of new features within not only alternative notebook platforms such as Google Colab but also popular exploratory visual analysis tools such as Tableau Desktop [3].

6.2 Implications for Data Science Tool Design

Our findings show that authors often use structural aspects of the notebook to track and manage the evolution of their analysis (Section 5). For example, notebook authors often use markdown cells to label sections of code and describe their analysis objectives. However, given that these structural elements are subject to change during analysis, we believe that our finding highlights a need for tools that help data scientists manage their goals *while* they analyze data within a notebook [9].

Generate Relevant Recommendations. Using the techniques we have demonstrated, notebook platforms can automatically calculate the position of a notebook document within the sensemaking spectrum *while it is being edited*. Platforms could use this information to support, teach, or even enforce best practices. For example, having detected that the author is in the exploratory phase of analysis, the platform may choose to automatically version the document to comprehensively capture competing branches of exploration.

We believe this information can be particularly pertinent to data science engines that wish to guide analysts with recommendations on analysis tools and techniques. For example, having detected that an author is performing exploratory analysis, a recommendation engine can cull recommendations from a group of curated *exploratory* notebooks. Our ideas can direct how recent work, such as by Yan et al. [46] and by Raghunandan et al. [30], generate recommendations found in Jupyter Notebooks. They can use the context of a notebook to provide more targeted data science recommendations to authors.

Provide Best-Practice Templates. Based on our observations in Section 5, it seems that people learning data science, i.e., authors explicitly leveraging notebook templates, are being taught to conduct their analyses in a goal- or objective-driven manner. In contrast, we did not observe templates corresponding to open-ended data analysis practices. This suggests that existing pedagogy provides direct *infrastructure* (i.e., templates) for more directed analysis [1], but not necessarily for open-ended exploration. While we cannot discount the possibility that authors engaging in open-ended exploration may also be using templates unobserved in our analysis, the templates that we did observe do not seem to teach open-ended exploration. We suggest that stronger guidance and infrastructure can be provided to facilitate best practices in open-ended data exploration such as through new systems and tools. With more training

and practice in open-ended exploration, infrastructure and standards for conducting open-ended exploration within computational notebooks will hopefully evolve [39], which in turn can enhance our ability to quantify its use in the real world.

Track Multiple Analysis Paths. GitHub versioning of computational notebooks does not help to track what data scientists do. For example, an analyst may pursue a particular line of inquiry, realize that a few analysis steps were dead ends, and backtrack to an earlier point to continue their analysis—introducing an alternative branch of investigation. It is hard to represent this non-linear flow with GitHub commits. We need mechanisms that track the actual non-linear and iterative practices of data scientists [9, 13, 15]. We suggest that an extension to current computational notebooks could remedy this problem—an extension that versions and manages cell dependencies. This enhancement would enable notebook users to better track their sensemaking processes and enable researchers to study sensemaking (and its evolution) in notebook environments.

6.3 Limitations and Future Work

Although our techniques produce a relatively small sample compared to the original corpus, our study is still one of the largest analyses of Jupyter Notebooks from GitHub (e.g., compared to [5, 40]). Part of the problem is the inconsistent notebook quality on GitHub [40]. We combat this challenge by proposing a method to identify data science notebooks suitable for quantitative analysis. This methodology could easily be extended to collect larger notebook corpora in the future; for example, by curating data science notebooks from all of the millions of notebooks on GitHub.

We approached our dataset with an understanding that many authors selectively report their analysis [21]. As our findings indicate, many notebooks on GitHub are skewed towards the explanatory side of the spectrum, suggesting that some authors may wait until later in the sensemaking process to share their notebooks. Coupled with a lack of ground truth for the mental models of the notebook authors, our ability to infer user intent was limited. We note that this is a fundamental limitation of surveying computational notebooks stored in a public repository such as GitHub, but that the benefits of getting the kind of insight demonstrated here far outweighs this drawback. We address this limitation in part through a mixed-methods analysis strategy in Section 4 and Section 5.

Nevertheless, it would be interesting to develop new strategies for collecting richer notebook metadata to fill observed gaps in GitHub histories and to infer user intent from this metadata. We view our work in this paper as the first of many to explore mixed methods towards understanding sensemaking in computational notebooks.

ACKNOWLEDGMENTS

This work was partially supported by the U.S. National Science Foundation CRII award IIS-1850115 and by the VMWare early career faculty grant. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. We would also like to thank Dr. Andrew Head and Dr. Michelle Mazurek for their guidance and mentorship. We gratefully acknowledge the classic 1998 Konami game *Dance Dance Revolution* for inspiring our paper title. Allow us to close with the following immortal words drawn from this game: "*You're a rockstar. You're the one they came to see. I'm crying, buckets of tears.*"

REFERENCES

[1] Sara Alspaugh, Nava Zokaei, Andrea Liu, Cindy Jin, and Marti A Hearst. 2018. Futzing and moseying: Interviews with professional data analysts on exploration practices. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2018), 22–31. https://doi.org/10.1109/tvcg.2018.2865040

[2] Adrian G. Barnett, Jolieke C. Van Der Pols, and Annette J. Dobson. 2005. Regression to the mean: what it is and how to deal with it. *International Journal of Epidemiology* 34, 1 (2005), 215–220. https://doi.org/10.1093/ije/dyh299

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

[4] Souti Chattopadhyay, Ishita Prasad, Austin Z. Henley, Anita Sarma, and Titus Barik. 2020. What's Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376729

[5] Helen Dong, Shurui Zhou, Jin L. C. Guo, and Christian Kästner. 2021. Splitting, renaming, removing: A study of common cleaning activities in Jupyter notebooks. In *Proceedings of the IEEE/ACM Conference on Automated Software Engineering Workshops*. IEEE, Piscataway, NJ, USA, 114–119. https://doi.org/10.1109/ASEW52652.2021.00032

[6] William R. Foster, Fred Collopy, and Lyle H. Ungar. 1992. Neural network forecasting of short, noisy time series. *Computers & Chemical Engineering* 16, 4 (1992), 293–297. https://doi.org/10.1016/0098-1354(92)80049-F

[7] Google. 2019. Colaboratory. https://colab.research.google.com/.

[8] Philip Jia Guo. 2012. *Software Tools to Facilitate Research Programming*. Ph. D. Dissertation. Stanford University.

[9] Andrew Head, Fred Hohman, Titus Barik, Steven M. Drucker, and Robert DeLine. 2019. Managing Messes in Computational Notebooks. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, Article 270, 12 pages. https://doi.org/10.1145/3290605.3300500

[10] Jeffrey Heer and Maneesh Agrawala. 2006. Multi-scale banking to 45 degrees. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 701–708. https://doi.org/10.1109/TVCG.2006.163

[11] Malin Källén and Tobias Wrigstad. 2021. Jupyter Notebooks on GitHub: Characteristics and Code Clones. *The Art, Science, and Engineering of Programming* 5, 3, Article 15 (2021), 31 pages. arXiv:2007.10146 https://arxiv.org/abs/2007.10146

[12] Sean Kandel, Andreas Paepcke, Joseph M. Hellerstein, and Jeffrey Heer. 2012. Enterprise Data Analysis and Visualization: An Interview Study. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2917–2926. https://doi.org/10.1109/TVCG.2012.219

[13] Mary Beth Kery, Amber Horvath, and Brad A Myers. 2017. Variolite: Supporting Exploratory Programming by Data Scientists. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1265–1276. https://doi.org/10.1145/3025453.3025626

[14] Mary Beth Kery, Bonnie E John, Patrick O'Flaherty, Amber Horvath, and Brad A Myers. 2019. Towards effective foraging by data scientists to find past analysis choices. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–13.

[15] Mary Beth Kery and Brad A. Myers. 2018. Interactions for Untangling Messy History in a Computational Notebook. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing*. IEEE, Piscataway, NJ, USA, 147–155. https://doi.org/10.1109/VLHCC.2018.8506576

[16] Mary Beth Kery, Marissa Radensky, Mahima Arya, Bonnie E. John, and Brad A. Myers. 2018. The Story in the Notebook: Exploratory Data Science using a Literate Programming Tool. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 174:1–174:11. https://doi.org/10.1145/3173574.3173748

[17] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E. Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica B. Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. 2016. Jupyter Notebooks – A publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. IOS Press, Amsterdam, Netherlands, 87–90. https://doi.org/10.3233/978-1-61499-649-1-87

[18] Donald E. Knuth. 1984. Literate Programming. *Comput. J.* 27, 2 (1984), 97–111. https://doi.org/10.1093/comjnl/27.2.97

[19] Andreas P. Koenzen, Neil A. Ernst, and Margaret-Anne D. Storey. 2020. Code duplication and reuse in Jupyter notebooks. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing*. IEEE, Piscataway, NJ, USA, 1–9. https://doi.org/10.1109/VL/HCC50065.2020.9127202

[20] Sean Kross and Philip J. Guo. 2019. Practitioners teaching data science in industry and academia: Expectations, workflows, and challenges. In *Proceedings of the*

ACM Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300493

[21] Yang Liu, Tim Althoff, and Jeffrey Heer. 2020. Paths explored, paths omitted, paths obscured: Decision points & selective reporting in end-to-end data analysis. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–14.

[22] Andreas Mathisen, Tom Horak, Clemens Nylandsted Klokmose, Kaj Grønbæk, and Niklas Elmqvist. 2019. InsideInsights: Integrating Data-Driven Reporting in Collaborative Visual Analytics. *Computer Graphics Forum* 38, 3 (2019), 649–661. https://doi.org/10.1111/cgf.13717

[23] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. 2019. Reliability and Inter-rater Reliability in Qualitative Research: Norms and Guidelines for CSCW and HCI Practice. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–23. https://doi.org/10.1145/3359174

[24] Michael Muller, Melanie Feinberg, Timothy George, Steven J. Jackson, Bonnie E. John, Mary Beth Kery, and Samir Passi. 2019. Human-centered study of data science work practices. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–8. https://doi.org/10.1145/3290607.3299018

[25] Michael Muller, Ingrid Lange, Dakuo Wang, David Piorkowski, Jason Tsay, Q. Vera Liao, Casey Dugan, and Thomas Erickson. 2019. How data science workers work with data: Discovery, capture, curation, design, creation. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–15. https://doi.org/10.1145/3290605.3300356

[26] ObservableHQ. 2011. Observable. https://observablehq.com.

[27] Joao Felipe Pimentel, Leonardo Murta, Vanessa Braganholo, and Juliana Freire. 2019. A large-scale study about quality and reproducibility of jupyter notebooks. In *Proceedings of the IEEE/ACM Conference on Mining Software Repositories*. IEEE, Piscataway, NJ, USA, 507–517. https://doi.org/10.1109/MSR.2019.00077

[28] Peter Pirolli and Stuart Card. 2005. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of the International Conference on Intelligence Analysis*, Vol. 5. The MITRE Corporation, McLean, VA, USA, 2–4.

[29] Roman Rädle, Midas Nouwens, Kristian Antonsen, James R. Eagan, and Clemens N. Klokmose. 2017. Codestrates: Literate Computing with Webstrates. In *Proceedings of the ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 715–725. https://doi.org/10.1145/3126594.3126642

[30] Deepthi Raghunandan, Zhe Cui, Kartik Krishnan, Segen Tirfe, Shenzhi Shi, Tejaswi Darshan Shrestha, Leilani Battle, and Niklas Elmqvist. 2021. Lodestar: Supporting Independent Learning and Rapid Experimentation Through Data-Drive Analysis Recommendations. In *Proceedings of the IEEE Symposium on Viusalization in Data Science*. IEEE, Piscataway, NJ, USA, 8 pages.

[31] Mohammed Suhail Rehman. 2019. Towards understanding data analysis workflows using a large notebook corpus. In *Proceedings of the ACM Conference on Management of Data*. ACM, New York, NY, USA, 1841–1843. https://doi.org/10.1145/3299869.3300107

[32] Adam Rule, Amanda Birmingham, Cristal Zuniga, Ilkay Altintas, Shih-Cheng Huang, Rob Knight, Niema Moshiri, Mai H. Nguyen, Sara Brin Rosenthal, Fernando Pérez, and Peter W. Rose. 2019. Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks. *PLoS Computational Biology* 15, 7 (2019), 8 pages. https://doi.org/10.1371/journal.pcbi.1007007

[33] Adam Rule, Aurélien Tabard, and James D. Hollan. 2018. Exploration and Explanation in Computational Notebooks. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 32:1–32:12. https://doi.org/10.1145/3173574.3173606

[34] Daniel M. Russell, Mark J. Stefik, Peter Pirolli, and Stuart K. Card. 1993. The cost structure of sensemaking. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 269–276. https://doi.org/10.1145/169059.169209

[35] Aurélien Tabard, Wendy E. Mackay, and Evelyn Eastmond. 2008. From individual to collaborative: the evolution of Prism, a hybrid laboratory notebook. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*. ACM, New York, NY, USA, 569–578. https://doi.org/10.1145/1460563.1460653

[36] James J. Thomas and Kristin A. Cook. 2005. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society Press, Los Alamitos, CA, USA.

[37] Edward R. Tufte. 2001. *The Visual Display of Quantitative Information*. Vol. 2. Graphics Press, Cheshire, CT, USA.

[38] John W. Tukey. 1977. *Exploratory Data Analysis*. Addison-Wesley, Reading, MA, USA.

[39] April Yi Wang, Anant Mittal, Christopher Brooks, and Steve Oney. 2019. How Data Scientists Use Computational Notebooks for Real-Time Collaboration. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (Nov. 2019), 39:1–39:30. https://doi.org/10.1145/3359141

[40] April Yi Wang, Dakuo Wang, Jaimie Drozdal, Xuye Liu, Soya Park, Steve Oney, and Christopher Brooks. 2021. What makes a well-documented notebook? A case study of data scientists' documentation practices in Kaggle. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–7. https://doi.org/10.1145/3411763.3451617

[41] April Yi Wang, Dakuo Wang, Jaimie Drozdal, Michael Muller, Soya Park, Justin D. Weisz, Xuye Liu, Lingfei Wu, and Casey Dugan. 2022. Documentation Matters: Human-Centered AI System to Assist Data Science Code Documentation in Computational Notebooks. *ACM Transactions on Computer-Human Interaction* 29, 2 (2022), 1–33. https://doi.org/10.1145/3489465

[42] Jiawei Wang, Kuo Tzu-Yang, Li Li, and Andreas Zeller. 2020. Assessing and restoring reproducibility of Jupyter notebooks. In *Proceedings of the IEEE/ACM Conference on Automated Software Engineering*. IEEE, Piscataway, NJ, USA, 138–149. https://doi.org/10.1145/3324884.3416585

[43] John Wenskovitch, Jian Zhao, Scott Carter, Matthew Cooper, and Chris North. 2019. Albireo: An interactive tool for visually summarizing computational notebook structure. In *Proceedings of the IEEE Symposium on Visualization in Data Science*. IEEE, Piscataway, NJ, USA, 1–10. https://doi.org/10.1109/VDS48975.2019.8973385

[44] Hadley Wickham and Garrett Grolemund. 2017. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. O'Reilly Media, Sebastopol, CA, USA.

[45] Kanit Wongsuphasawat, Yang Liu, and Jeffrey Heer. 2019. Goals, Process, and Challenges of Exploratory Data Analysis: An Interview Study. *CoRR* abs/1911.00568 (2019), 11 pages. arXiv:1911.00568 http://arxiv.org/abs/1911.00568

[46] Cong Yan and Yeye He. 2020. Auto-Suggest: Learning-to-Recommend Data Preparation Steps Using Data Science Notebooks. In *Proceedings of the ACM Conference on Management of Data*. ACM, New York, NY, USA, 1539–1554. https://doi.org/10.1145/3318464.3389738