Notebooks in the Wild: Classifying Applications, Structure, and Data Transformations in Jupyter Notebooks

Jacob Yim [1] **and Qinshi Zhang** [1]

[1] *University of California, San Diego*

Abstract

Jupyter Notebooks are popular tools for working with data. In order to better understand how to support notebook users, we conduct a corpus study of 30 real-world, open source Jupyter notebooks. We classify notebooks based on application, structure, and types of data transformations, finding that notebooks are used in diverse ways. We identify opportunities to support notebook users in using notebooks with nonlinear structure, and in applying notebooks for education and scripting.

Keywords: Jupyter Notebooks. Computational Notebooks.

1 Introduction

Jupyter Notebooks are among the most popular modern tools for working with data [1]. However, despite their popularity, prior work has identified a variety of usability issues with notebooks. Among them are reproducibility problems associated with hidden state, challenges with managing dependencies, and difficulties writing modular, reusable, and shareable code within the notebook ecosystem [2]. While many prior studies focus on understanding pain points faced by notebook developers, comparatively few have attempted to classify the tasks performed in notebooks and the practices employed by notebook users. A better understanding of how notebooks are currently used could help inform future tool-building directions and uncover groups of users unsupported by existing efforts. In this preliminary study, we conduct an exploratory analysis on a corpus of 30 real-world Jupyter notebooks. In particular, we focus on classifying **applications**, **structure**, and **data transformations** within these samples. We aim to answer three research questions through this analysis:

- **RQ1**. What types of tasks do real users apply Jupyter Notebooks to?
- **RQ2**. How do users structure their notebooks to accomplish their tasks?
- **RQ3**. What data transformations do users perform in notebooks to accomplish their tasks?

Through open coding and content analysis methodologies, we find that users employ notebooks for a wide variety of tasks, including use cases not directly related to data analysis, like education and scripting, and utilize a variety of data transformations. Additionally, a significant fraction of notebooks involve nonlinear structure. These findings suggest new design opportunities for improving computational notebooks for broader groups of users.

2 Method

To begin classifying notebook applications, structure, and data transformations, we first collected a corpus of 30 publicly available Jupyter Notebooks via The Stack [3], a 3.1 TB dataset of open-source code from GitHub. This sample was collected randomly and without prior knowledge of notebook contents.

We then performed open coding on the notebooks in this dataset, manually assigning codes to notebooks and building a codebook iteratively. During this process, we treated notebooks as qualitative data, and assigned to each notebook a set of codes relating to its use case, structure, and the data transformations within. Specifically, to address **RQ1** and **RQ2**, we assigned each notebook classifications for its broader use case, type of data, and linear/parallel structure. Meanwhile, for **RQ3**, codes for data transformations were applied at the level of individual blocks of code. As a result, each notebook was associated with only one each of use case, data type, and structure type, but could be assigned multiple data transformation codes. We constructed these classifications iteratively, frequently adding new codes and revisiting previously coded documents during this process.

PLATEAU
13th Annual Workshop at the Intersection of PL and HCI

DOI: 10.35699/1983-3652.yyyy.nnnnn

Organizers:
Sarah Chasins, Elena Glassman, and Joshua Sunshine

Following open coding, we analyzed the results using an informal content analysis methodology. For the codes pertaining to **RQ1** and **RQ2**, we produced total counts of notebooks in the dataset exhibiting each code. For the codes related to **RQ3**, we decided not to conduct quantitative content analysis, as we found comprehensively classifying data transformations to be difficult given the variation in code and sheer number of possible translations.

3 Results

3.1 **RQ1**: Types of Tasks Performed in Notebooks

To address **RQ1**, we assigned each notebook to categories corresponding to the broad type of task performed within, as well as the type of data worked with inside the notebook. During open coding, we identified four broad categories of notebooks in our dataset:

- **Modeling**: Notebooks in this category provided data to a model, ostensibly to make predictions or perform analysis.
- **Data Analysis**: Tended to show properties of an existing dataset, including relationships between features.
- **File Manipulation/Scripting**: Used to automate some process or transformation, typically modifying files on the system.
- **Educational**: Used to demonstrate concepts using toy data, or were class assignments with tasks to be performed by students.
- **N/A**: There was only one notebook in this category, which was only a few cells long and lacked enough information to assign to a task.

The total counts of notebooks in each of the above categories are displayed in Table 1.

Category	**Count**
Modeling	11
Educational	11
Data Analysis	4
File Manipulation/Scripting	3
N/A	1

Table 1. Counts of notebooks in each task category.

In addition to categorizing broad groups of tasks, we also designed codes for types of data involved in each notebook:

- **Numerical**: Notebooks in this category dealt primarily with numerical data, e.g. integers or floating point numbers.
- **Text**: Dealt primarily with text data, e.g. strings or tokenized text.
- **Mixed**: Dealt with data that was potentially categorical and numerical, quantitative and qualitative.
- **Other Files**: Worked primarily with files with a specified structure, e.g. TIF files.
- **N/A**: Notebooks in this category did not work with data, or contained few meaningful data transformations.

The counts of notebooks with each data type are displayed in Table 2.

3.2 **RQ2**: Structural Patterns in Notebooks

Based off of our observations during the coding phase, we chose to focus on the degree of sequentiality in notebook operations for our analysis of notebook structure. We identified two broad structural patterns:

- **Linear**: Computational steps in these notebooks tended to follow from one another sequentially.
- **Parallel**: These notebooks tended to incorporate some element of comparison between different approaches, often with cells sharing similar code. Operations in these notebooks tended not

Category	Count
Numerical	11
Text	8
Mixed	4
Other Files	3
N/A	4

Table 2. Counts of notebooks with each data type.

to follow directly from one another in order, but instead performed computations on different "branches" of notebook state. [1]

The counts of notebooks exhibiting each structural pattern are displayed in Table 3.

Category	Count
Linear	22
Parallel	8

Table 3. Counts of notebooks with each structural pattern.

3.3 **RQ3**: Data Transformations in Notebooks

During open coding, we also attempted to construct a rough classification of data transformations. We identified 22 different codes, which we organized into five different categories: string operations, column/array transformations, row transformations, dataframe transformations, and encoding. The full codebook is displayed in the appendix.

Data transformations took on a wide range of forms, as evidenced by the various codes and categories we identified during open coding. While we did not perform quantitative analysis of our codes, the codebook we produced demonstrates some of this variation.

4 Discussion

Our investigation of **RQ1** yielded several surprising results. While Jupyter Notebooks are typically known as tools for data analysis and exploration [1], our analysis provides evidence that notebooks may be commonly used for other purposes as well. Educational notebooks made up a large proportion of our sample (11/30, about 36.7% of total)–however, this figure may be artificially inflated by our sampling strategy, as education-related notebooks may be more likely to be publicly posted on GitHub and distributed or duplicated by students. Meanwhile, data analysis notebooks made up a comparatively small portion of the sample–only 4/30 notebooks, about 13.3% of the total. This may have been affected by the definitions of our codes–notebooks classified as modeling, for instance, may have had some overlap with data analysis, but notebooks involving training and evaluating models were generally always classified as modeling notebooks. Finally, we found file manipulation scripts to be a surprising use case for Jupyter Notebooks. 3/30 notebooks, or 10% of the total, were classified as file manipulation/scripting notebooks. These notebooks performed tasks that other programmers might use a shell script for: one was simply used to move files between directories, and another systematically modified a group of files in the TIF format. It would be interesting to investigate further why some users choose Jupyter over other options, including regular Python scripts, for these types of tasks.

For **RQ2**, we found the frequency of notebooks with nonlinear structure to be a particularly interesting result–8/30 notebooks were categorized as having parallel structure, or about 26.7% of

1 In contrast to previous classifications, many notebooks in the corpus exhibited both linear and parallel traits; in these cases, we tended to classify notebooks as parallel. Therefore, the Parallel classification should be interpreted as including all notebooks containing *some* notion of branching or comparative analysis.

the sample. These branching organizational structures may be a reflection of the exploratory nature of notebooks and data science as a whole; programmers may write non-sequential code in the process of exploring different approaches to working their data. This matches with our other observations that "parallel" notebooks tended to have blocks of code with a high degree of similarity; for instance, users performing modeling tasks may have wanted to compare results between different models or different parameters to find the best approach. For some users working with data, notebooks may serve as a tool for thinking about branching options and alternate paths; these users may require new tools that match their nonlinear thinking. Additionally, the fact that these notebooks in our sample contained so much duplicate code could imply that these users need better tooling for modularity and for comparing different versions of notebook code and cells.

Our analysis of **RQ3** reflected a theme also indicated by our answers to **RQ1** and **RQ2**: notebook users work in a variety of ways. Notebooks are now used for various tasks and types of data, and may not even be limited to the domain of data analysis. Notebook users, accordingly, employ diverse approaches and transformations to reach their goals.

5 Limitations

One major limitation of our corpus-based approach to studying notebooks is that we cannot fully assess user intent from only their code. As a result, our classifications represent subjective interpretations of the code. This applies particularly to the analysis of types of tasks performed in the notebooks, which is closely related to users' purpose and intent–without real users, we can only approximate this underlying meaning.

In our analysis of **RQ2**, we chose to focus on the degree of sequentiality in notebook operations to form our classifications. This represents only a single facet of notebook structure–we chose this classification based off of the patterns in our dataset, but other axes of notebook structure remain unexplored by our analysis.

Finally, our codes for **RQ3** have not been formally analyzed (e.g. using thematic or content analysis), making it difficult to draw conclusions relating to **RQ3**. Consequently, our results for this research question are limited.

6 Future Work

Our study uncovers several open questions, especially relating to the design of future tools. We identified a number of use cases for notebooks outside of their traditional use for data analysis. For future work, we ask: how can we better support users of notebooks for education and scripting tasks? Additionally, we identified nonlinear structure in notebooks; now, how can we support notebook users in working nonlinearly and comparing between approaches? In particular, we would like to explore the design space of notebooks that are not limited to linear, one-dimensional execution: what would a "branching" notebook look like?

7 Conclusion

To assess how Jupyter notebooks are used in practice, we conducted a corpus study of 30 open-source notebooks. Using open coding and content analysis, we classified notebooks based on use case, structure, and data transformations within. Ultimately, we find that notebooks are used for a wide variety of tasks in diverse ways. We identify groups of users who may be underlooked, including users of notebooks for education and scripting. We also find that some notebooks are structured in nonlinear ways, holding implications for new notebook designs. We hope that these results may be used to inform the development of future tools.

8 Appendix

A Data Transformation Codes

Category	Code
String operations	▪ Removing pattern ▪ Replacing pattern ▪ Convert to lowercase ▪ Convert to uppercase ▪ Split ▪ Strip ▪ Filter strings based on property (length, pattern, etc.)
Column/array transformations	▪ Convert column type ▪ Drop column ▪ Add new column ▪ Apply function to column ▪ Replace missing values ▪ Select columns
Row transformations	▪ Filter rows ▪ Add rows ▪ Drop rows ▪ Select rows ▪ Slice rows
Dataframe transformations	▪ Groupby ▪ Join tables ▪ Get dataframe shape
Encoding	▪ Encoding methods

Table 4. Data transformation codebook.

References

[1] J. M. Perkel, "Why jupyter is data scientists' computational notebook of choice," *Nature*, vol. 563, no. 7732, pp. 145–147, 2018.

[2] S. Chattopadhyay, I. Prasad, A. Z. Henley, A. Sarma, and T. Barik, "What's wrong with computational notebooks? pain points, needs, and design opportunities," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20, Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–12, ISBN: 9781450367080. DOI: 10.1145/3313831.3376729. [Online]. Available: https://doi.org/10.1145/3313831.3376729.

[3] D. Kocetkov, R. Li, L. B. Allal, *et al.*, *The stack: 3 tb of permissively licensed source code*, 2022. arXiv: 2211.15533 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2211.15533.