Title

Understanding Interactive and Reproducible Computing With Jupyter Tools at Facilities

Authors

Paine, Drew

Ramakrishnan, Lavanya

Publication Date

2023-12-05

Peer reviewed

Understanding Interactive and Reproducible Computing With Jupyter Tools at Facilities

Drew Paine and Lavanya Ramakrishnan
Data Science and Technology Department
Lawrence Berkeley National Laboratory
{pained, lramakrishnan}@lbl.gov

October 2020

Abstract

Increasingly Jupyter tools are being adopted and incorporated into High Performance Computing (HPC) and scientific user facilities. Adopting Jupyter tools enables more interactive and reproducible computational work at facilities across data life cycles. As the volume, variety, and scope of data grow, scientists need to be able to analyze and share results in user friendly ways. Human-centered research highlights design challenges around computational notebooks, and our qualitative user study shifts focus to better characterize how Jupyter tools are being used in HPC and science user facilities today. We conducted twenty-nine interviews, and obtained 103 survey responses from NERSC Jupyter users, to better understand the increasing role of interactive computing tools in DOE sponsored scientific work. We examine a range of issues that emerge using and supporting Jupyter in HPC ecosystems, including: how Jupyter is being used by scientists in HPC and user facility ecosystems; how facilities are purposefully supporting Jupyter in their ecosystems; feedback NERSC users have about the facility's deployment, and, discuss features NERSC indicated would be helpful. We offer a variety of takeaways for staff supporting Jupyter at facilities, Project Jupyter and related open source communities, and funding agencies supporting interactive computing work.

Keywords — Jupyter, HPC user facilities, science user facilities, qualitative research
Report Number: LBNL-2001355

1 Introduction

High Performance Computing (HPC) and science user facilities are integral components of data-intensive research in the United States. Many facilities in the US are funded by the Department of Energy, as well as other entities like the National Science Foundation or National Institutes of Health. HPC facilities sustain dynamic ecosystems with advanced computing systems, large amounts of data storage, and high speed networking that enable scientists to complete data-intensive work. The National Energy Research Scientific Computing Center (NERSC), located at Berkeley Lab, is one such facility. Science user facilities provide large-scale resources for running experiments and collecting data, as well as computing resources for accessing and using the generated datasets. Examples include: the Advanced Light Source (ALS) synchrotron at Berkeley Lab [1], the Vera C. Rubin Observatory [27], or the multi-lab Atmospheric Radiation Measurement (ARM) facility [2]. Each type of facility provides essential scientific resources, and their science users increasingly work interactively using computational notebooks.

Computational otebooks, such as Jupyter Notebooks[1], enable users to write code for exploratory analyses, plot data, and share results interactively in a single document. Notebooks enable scientists to explore and process massive datasets in computational environments ranging from personal computers to the commercial cloud. Scientists using HPC and science facilities are already incorporating Notebooks into their work in these ecosystems. Increasingly, facilities are working to better provision and support Notebooks, and other interactive tools, in their computing ecosystems using products from Project Jupyter [21]. This is essential as scientists undertake human-in-the-loop or experiment-in-the-loop decision making to shape unfolding data collection or analysis with massive quantities of data. Researchers require the ability to explore and analyze these products in reasonable amounts of time, and frequently need the ability to use advanced machine learning tools when accomplishing these tasks. We consequently need to better understand how Jupyter Notebooks, and other tools from open source ecosystems, are provisioned and supported in HPC and science user facilities. Improving our understanding of the dynamics of researchers, tools, and ecosystems will help us identify potential improvements to support the continued advancement of scientific work.

Addressing this gap in our knowledge, we conducted two qualitative user research studies in 2019 and 2020. These studies investigate 1) how scientists use Jupyter Notebooks at facilities, and 2) how HPC and science user facilities provide and support Jupyter setups for their users. Qualitative user research enables us to learn about the ways people do work, while exploring the dynamic cultures formed by varied scientific groups and facilities. Our studies explored the ways facilities are supporting their user's scientific work with Jupyter products, how scientists are using Jupyter Notebooks to accomplish different research activities, and identifies gaps in the alignment of Jupyter products and the systems and cultures of facilities. First, we conducted a semi-structured interview study of stakeholders at both types of facilities in the United States, as well as members of Project Jupyter, to understand how they engage with Jupyter products, and understand the work they support in depth. Second, we used a descriptive survey to obtain feedback from HPC users who utilize the NERSC Jupyter installation. This descriptive survey provides a higher level view of Jupyter usage in the NERSC ecosystem. Our semi-structured interviews yield nuanced, in-depth examinations of work at NERSC, and other facilities around the United States.

Our findings show that a variety of HPC and science user facilities are provisioning and supporting Jupyter products. This is typically accomplished by adapting and running JupyterHub to fit with the technologies and policies of a given facility's ecosystem. Scientists who use facility resources rely on Jupyter for everything from exploratory data analysis to machine learning workflows, visualization of results, and even just simpler, easy access to an HPC system. Project Jupyter's products are designed to work in different computing environments, and we see that varied facilities are able to successfully adapt and customize these resources for their needs. Challenges still emerge that make it difficult for facilities, and their users, to readily adopt JupyterLab Extensions that can facilitate smoother interactive workflows, or to share Notebooks. Ameliorating these issues, and improving user experiences, is necessary as HPC systems and scientific instruments at user facilities steadily support, and require, interactive workflows to address research problems.

[1] In this report, when referring to Jupyter Notebooks specifically we capitalize the term.

2 Background

In this section we situate our qualitative research findings by first examining the open source Project Jupyter ecosystem. We then provide a brief description of the NERSC HPC facility, and its systems, including a Jupyter setup. Finally, we discuss human-centered research on computational notebooks, and explain how our research focuses beyond notebooks to investigate large scientific computing and data ecosystems.

2.1 Project Jupyter

Project Jupyter is an open source undertaking producing software while developing standards for data science tools [21]. The project fosters and sustains a vibrant community spanning academia and industry. Products being built include: the classic Jupyter Notebook; JupyterLab, a newer web-based interactive development environment; and, JupyterHub which facilitates and orchestrates the execution of Notebooks for multiple users on shared computing systems. Project Jupyter as a community also supports a variety of related tools from throughout its ecosystem, including nbdime [20] for version controlling notebooks and the Binder Project [22] for supporting reproducible computing environments.

Jupyter Notebooks are designed to enable interactive computing with data everywhere from personal computers to the cloud. Notebooks rely on underlying kernels to handle computation while a web-based interface allows users to work on their documents. HPC users can install and run Notebooks on system nodes if a facility's policies allow. JupyterHub is the project's resource for providing and managing notebooks for multiple users in an environment, including providing an extensible user authentication framework and a kernel spawner. The JupyterHub spawner launches single user kernels for Notebooks while providing access to varied resources (e.g. different computing systems, filesystems, etc.). The project commonly illustrates the utility of JupyterHub with use cases involving cloud computing or private clusters. Individual facilities deploying JupyterHub adapt it by customizing the spawner and authentication setups to work with their particular policies and resources. Finally, JupyterLab is the project's latest user interface. The JupyterLab interface is an interactive development environment that integrates classic Notebooks with file browsers, terminals, and more through JupyterLab Extensions. JupyterLab's extensibility enables projects or facilities to customize the user experience by provisioning default sets of JupyterLab Extensions that support different types of tasks. Adopting JupyterLab in combination with JupyterHub enables facilities, or other entities, to provide a more cohesive environment for interactive data exploration and analysis.

Drawing upon JupyterHub, the Binder Project offers a toolset that with one-click takes Notebooks and their environment specifications from a version control repository, then spins up an ephemeral container in the commercial cloud. Users can play with the analyses captured in the Notebook with everything necessary to start up the kernel already installed. Binder is particularly useful for workshops where users are asked to try out a new library, or learn how to accomplish a data science analysis. Binder's reliance upon the commercial cloud means it currently does not work in HPC ecosystems without adaptation.

2.2 Jupyter at NERSC

The National Energy Research Scientific Computing Center (NERSC) is an HPC facility at Berkeley Lab [17]. NERSC provides its users with access to HPC systems, storage for massive quantities of data, and high speed networking in connection with ESnet [7]. The facility currently offers the Cori HPC system, a Cray XC40. Cori is composed of 2,388 Haswell nodes and 9,668 KNL nodes for up to 30 petaflops of scientific computing [16]. Cori also has a subset of nodes provisioned with GPUs in advance of NERSC's next system, Perlmutter. Perlmutter will be coming online in late 2020 and will offer a variety of new features [18]. NERSC also provides Spin, a container-based platform for deploying services, science gateways, workflow managers, and so on [19]. For data storage NERSC offers various filesystems for global storage, machine specific local storage, and a tape-based backup system.

NERSC has been providing users with a Jupyter setup, for more than three years, that helps users work on their HPC systems (currently Cori but soon Perlmutter as well), as well as the Spin service platform. At the time of our study three of the Cori login nodes were dedicated to the Jupyter installation, which is an expansion from one node originally. The facility uses JupyterHub to present users with a landing page where they can select a computing system to spawn a Notebook server. This Notebook server will be

connected to all of the available filesystems and displayed through JupyterLab. With this setup the facility makes it simple for users to run Jupyter Notebooks, and includes customizations to help users navigate the various filesystems.

2.3 Human-Centered Studies of Computational Notebooks

With the advent and adoption of computational notebooks in data science, human-centered research in the fields of Human Computer Interaction (HCI) and Computer Supported Cooperative Work (CSCW) has increasingly examined the intertwined social and technical (sociotechnical) facets to the design and use of these tools. Researchers in these fields investigate computational notebooks (including Jupyter Notebooks, among others) as increasingly common data science tools that have a variety of design and usability challenges [4,9,13–15,25]. Broadly, this work focuses on how the design of notebooks as flexible tools influences the ways researchers use them to do work, along with an emphasis on how notebooks are or are not shared. Our report complements this past research by shifting focus, exploring the work facilities undertake to make Notebooks more usable in their ecosystems.

The first common theme across human-centered research is the examination and critique of the fundamental design of notebooks, specifically as they are used in data scientist's practices. Research emphasizes that the notebooks scientists produce are often "messy" because they are primarily employed for exploratory data analysis [9, 14, 24, 25]. The open-ended design of computational notebooks makes it easy for them to become long unwinding narratives of work in practice. Rule et al. [25] found users prefer messy notebooks for personal exploratory work, and then undertake additional work to produce a cleaned up version when sharing results with collaborators. Rule et al. [24] subsequently developed and evaluated an extension for Jupyter Notebooks to facilitate this cleanup work. This extension enables a user to "fold" cells and produce a simplified Notebook for sharing with colleagues. Similarly, Head et al. [9] designed and prototyped a tool for taking messy notebooks and exporting a streamlined version that is more shareable, and suitable for documenting a research process. Understanding that notebooks often end up being disorganized, collectively this research demonstrates a need for care and concerted effort when users are sharing, or publishing, these artifacts for review and use by other individuals.

A related vein of research focuses on the ways computational notebooks help data scientists produce narratives in their work. This research underscores how key decisions leading to final analyses are not typically captured, losing important provenance information. Kery and colleagues [13–15] completed multiple studies examining how data scientists think through problems using computational notebooks. Framing notebook work as storytelling, the authors find that notebooks are limited in their ability to help produce detailed records of the decisions made over the course of exploratory work, e.g. provenance being captured is minimal [15]. To address this issue, Kery et al. [14] designed and implemented a tool to capture this type of information so that data scientists could leverage it when revisiting their work at a later time.

Finally, Randles et al. [23] briefly examined how astronomers are or are not citing notebooks in their publications. Of 91 publications examined they found only 37 linked to openly accessible notebooks and 54 more mentioned notebooks but did not provide access to them. The authors motivation was to specifically explore how Jupyter notebooks can embody the FAIR [29] principles for reproducible open science. This work aligns with efforts to facilitate open communication and an emphasis on notebooks as executable artifacts associated with publications.

These human-centered investigations are primarily scoped around the computational notebook as an artifact being used in data science practices. Left under explored is the larger sociotechnical context to using Jupyter Notebooks. These interactive documents require access to sufficient computing systems, and relevant data, to be usable and useful for data science work. Our investigations address this gap by examining the complexities that emerge with supporting Notebooks in HPC and science user facility ecosystems. Our findings illustrate facets to this key sociotechnical context that help us look beyond the design of Notebooks alone.

3 Research Methods

We interviewed thirty-two researchers, computer scientists, and open source developers between September and December 2019. Subsequently, we surveyed recent users of NERSC's Jupyter installation between March - April 2020 (receiving 103 responses from over 900 users). Our interviews were designed to capture the how and why of individual's work with Jupyter tools in HPC environments. Our descriptive survey developed a high-level view of the way NERSC's Jupyter setup is being used today. Here we provide an overview of our two approaches. Additional details can be found in the appendices.

3.1 Exploratory Semi-Structured Interviews

Interviewing is a technique for gathering nuanced insights from individuals about their work, practices, and collaborations. Semi-structured interviews enable researchers to have consistent conversations when interviewing multiple people, while still probing more deeply in areas of relevance with particular individuals [28]. We interviewed thirty-two individuals between September and December 2019 through twenty-nine semi-structured interviews. One interview was conducted jointly with two HPC facility staff members present. A second was conducted simultaneously with three postdoctoral researchers from a scientific user facility. Classifying the interviewees, eleven were domain scientists (who may also contribute to open source science tools), fifteen were computer scientists and practitioners building scientific software (including members of HPC facilities or large science projects), five were Project Jupyter team members, and one individual was a developer of an open source software tool. We gathered more than twenty-six hours of recorded data (avg. 55 minutes per interview). The recordings were professionally transcribed then cleaned by the first author for analysis (see Appendix A.1.3).

We designed our exploratory interview study with two complementary protocols (see Appendix A.1). The first protocol was designed to learn about a participants work, details about their use of Jupyter tools including how this has changed over time, and the impact of Notebooks on the reproducibility of their work. For individuals in development or HPC facility roles we also asked how they are supporting Jupyter in HPC environments, and how they have customized these tools for that ecosystem. The second protocol was designed to learn more about work on these tools by Jupyter team members, or other open source developers. We inquired about an individual's history with the project, motivations for contributing, what they think is unique about these tools, how they believe their products fit in different computing ecosystems, and their experiences engaging with Jupyter users.

3.2 NERSC Jupyter User Survey

Surveys provide higher-level qualitative insights about a group of users. We designed our descriptive survey to complement the initial findings from our interviews by gathering data from active users of NERSC's Jupyter installation. This survey was open for one month between March - April 2020 and advertised widely to this user community.

We designed our survey with questions about the ways scientists generally use Jupyter in their work, as well as their use of Jupyter at NERSC specifically (see Appendix B). Questions explored how long individuals have been using Jupyter overall and at NERSC, how they share Notebooks with colleagues, how often they use Jupyter in a variety of computing environments (e.g., desktop, HPC, cloud), the research activities they use Jupyter for (e.g., data exploration, analysis, prototyping code, Machine Learning work, etc.), and what does and does not work well when using Jupyter at NERSC.

4 Findings

Our datasets produced a variety of insights about the use of Jupyter, challenges users and facilities face, and opportunities for further work. We see that Jupyter tools are frequently used on personal computers and at HPC facilities, along with private clusters and the commercial cloud, depending on the resources scientists, or their collaborations, have the ability to access. Facilities providing Jupyter installations are doing so by adapting and customizing the project's JupyterHub and JupyterLab products to suit the policies

and features of their environment. For HPC facility users, these setups are appreciated and make working with large computing systems easier. Challenges still emerge around outages, performance, documentation, and the ability to make customizations.

4.1 Jupyter Usage in HPC Ecosystems

Our interviews and survey broadly examined where and how participants use Jupyter Notebooks, since facilities manage JupyterHub and JupyterLab for their users, as well as the ways individuals are sharing Notebooks with colleagues. This high-level contextual information about Jupyter usage on HPC systems, and beyond, provides us with views across the varied ecosystems where these tools are being utilized for science. Knowing where and broadly how these products are used, we can then examine how different HPC and science user facilities provide and manage these tools.

Computing environments where Jupyter Notebooks are being used. Researchers interviewed use Jupyter Notebooks in a variety of environments, from their personal computers to private clusters, HPC systems, and the commercial cloud. NERSC users indicated that when using Notebooks they rely on their own personal computers and NERSC HPC systems most frequently. Some survey respondents also work with other HPC facilities, private clusters, or the commercial cloud at different frequencies. Interestingly, many survey respondents indicated that they never use the commercial cloud, private clusters, other HPC systems, or other systems, see Figure 1.

Takeaways.

- *Participants use Jupyter Notebooks and tools in a variety of computing environments. Usage depends on what resources are available to them through their projects or organizations.*
- *Feedback from this sample of NERSC users suggests they are able to achieve their work using NERSC's Jupyter setup or their local machines.*

How Jupyter Notebooks are being used in HPC ecosystems. Knowing where participants are using Jupyter Notebooks, we wanted to know what types of work they use them for, particularly when working in HPC ecosystems. Analyzing our interview and survey data, we find a variety of tasks that researchers accomplish using Jupyter Notebooks in one way or another (summarized in Figure 2). These insights illustrate the breadth of work that can be accomplished using Notebooks with HPC resources.

Overall study participants are using Jupyter Notebooks to analyze their datasets large and small. Participants noted they frequently use Jupyter Notebooks to prototype code for analysis workflows, or eventual software packages. Visualizing data that has been processed within Notebooks is common as well. Interviewees emphasized that JupyterLab makes it easy to quickly access HPC system resources so that they can browse data and work in Notebooks in one interface. Jupyter users at NERSC are using this setup to explore data on the various HPC filesystems, share results or code with collaborators, or taking advantage of the processing power or GPU nodes NERSC provides to run Machine Learning workflows. A few survey respondents indicated that they are using Jupyter Notebooks to execute their batch processing jobs. Some respondents use parallel processing tools (e.g., Dask [5] or ipyparallel [10]), even as they commented on the difficulty of using these tools on NERSC systems.

Takeaways.

- *Our participants use Jupyter Notebooks for common data science tasks including data exploration, prototyping analysis codes, data analysis, and visualization.*
- *Some NERSC users utilize Jupyter Notebooks for machine learning work. Studying these workflows, and their use of interactive computing tools, as new resources like GPU nodes are made available, may produce useful insights about the adoption of different tools and hardware.*
- *Further investigating how scientists are trying to use Notebooks to launch, manage, and monitor batch processing jobs may illuminate how these resources augment HPC practices. This may help facilities identify areas for further development and integration of interactive computing tools to make HPC systems more usable to a variety of users.*

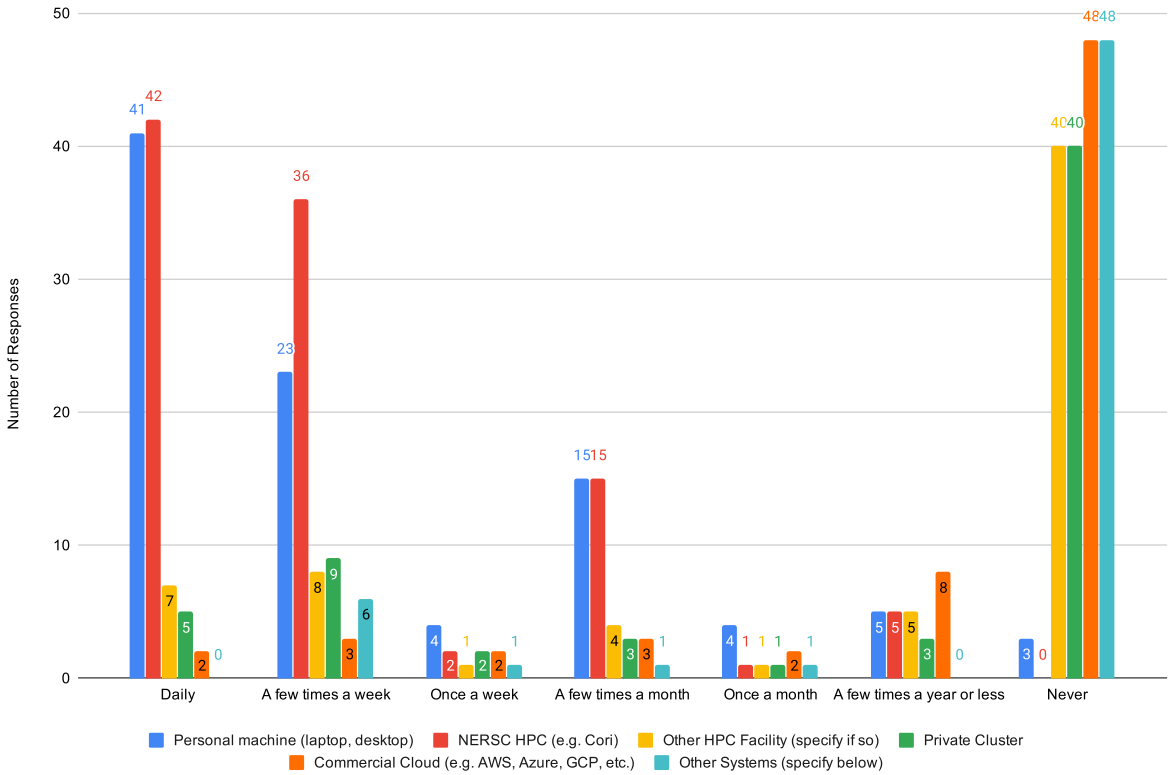

Figure 1: Responses indicating how frequently (daily, a few times a week, once a week, etc.) survey respondents use Jupyter Notebooks in different contexts (personal machine, NERSC HPC, commercial cloud, etc.). Respondents could check multiple options for each frequency and computing context. Results capture how many responses were provided for each combination.

How Jupyter Notebooks are being shared in HPC ecosystems. Our studies examined how users share their work with colleagues using Notebooks. All of our interviewees indicated they share Notebooks in some way. They typically do so with collaborators, and at times the public or their community. 71 of our 103 (69%) survey respondents indicated they share Notebooks. The remaining thirty-two respondents answered that they do not share notebooks, and they did not elaborate on why they chose this answer. Studying motivations for sharing or not sharing Notebooks in more depth would be worthwhile. This would help us better understand whether this is a lack of need, desire, or an inability to successfully share due to a computing challenge, or some other reason. Understanding these motivations, or hindrances, is important since the ability to share or not share Notebooks could influence the potential reproducibility of the research.

Among individuals sharing Notebooks, mechanisms for accomplishing this currently include email, version control repositories, shared file systems like NERSC's Community File System (CFS), and even Slack, see Figure 3. Four survey respondents mentioned that they use Google's CoLab [8] service for sharing and working in Notebooks in real time with collaborators. CoLab is a fork of Jupyter tools, creating a closed ecosystem that adds collaborative features like Google Docs to Notebooks. These features are not interoperable with HPC or other ecosystems. An individual commented that they have "gotten quite addicted to the ease of use of CoLab. Being able to share and run code without any worry about the installation/etc makes the sharing process much faster." This ease of use is possible due to the closed environment and CoLab making a Notebook's environment available to anyone the file is shared with in the cloud. This same individual also explained how their scientific collaboration has a convoluted process for sharing Notebooks with the necessary environment information for them to be able to run these files on different computing systems. This response underscores that sharing Notebooks is about more than just sharing the file itself. Users must also be able to share the environment configuration, something the Binder Project is building

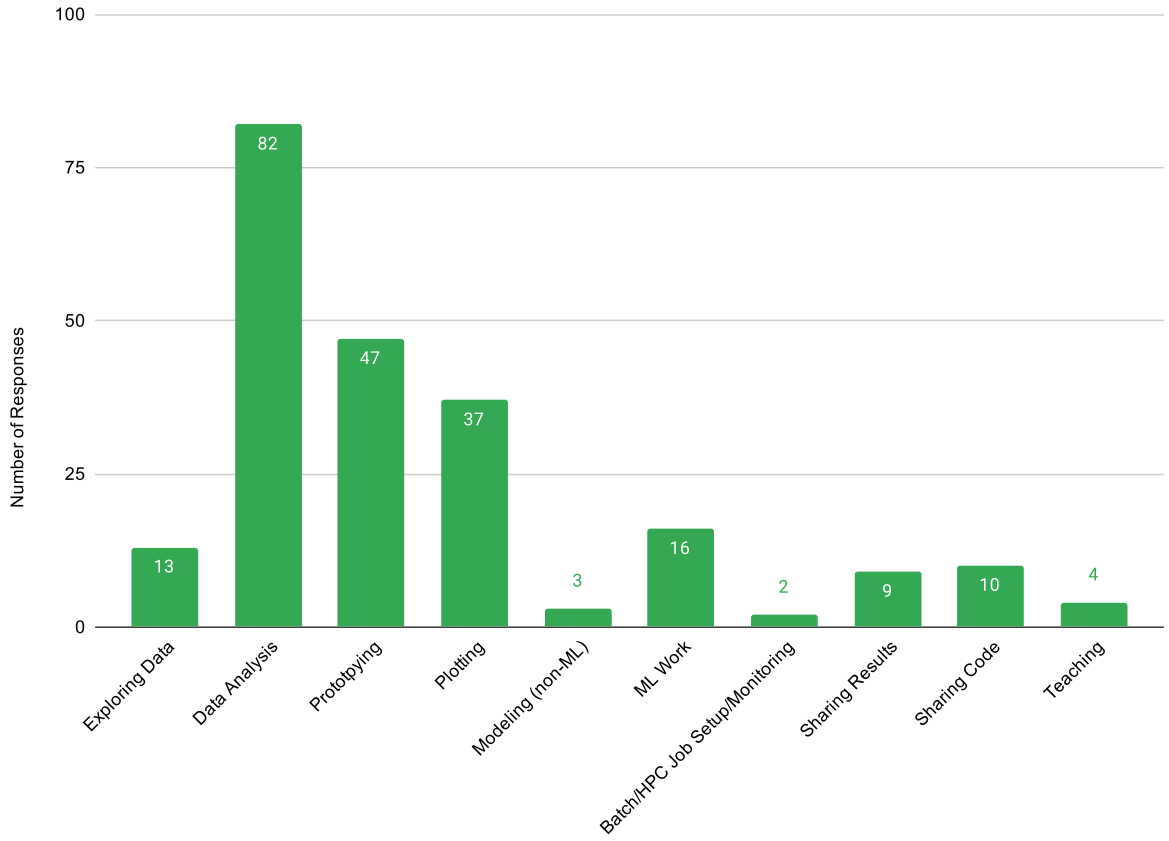

Figure 2: Survey Responses from NERSC users indicating how they use Jupyter Notebooks in their work. Free responses were open coded by the first author, see Appendix B, and categorized by the topics presented along the x-axis. The number of responses for each category make up the y-axis.

tools to facilitate regardless of the computing environment in question.

Takeaways.

- *Scientists are most commonly sharing Notebooks through a shared file system or a version control solution. Sharing through the shared file system on an HPC system is easy but relies on ad-hoc methods to maintaining versions and provenance. However, version control systems add an extra overhead for the user.*
- *Sharing Notebooks requires sharing the artifact and the specification for the computational environment. Users are not always aware of the best strategies for accomplishing this task.*
- *Many survey respondents are not sharing Notebooks. Investigating whether this is due to a lack of need or technical challenges that could be addressed by NERSC may improve user's experiences and the reproducibility of their work.*

4.2 Supporting Jupyter at HPC and Science User Facilities

We designed our interviews to explore how HPC facilities and science user facilities support Jupyter in their ecosystems. Scientific users can run Jupyter Notebooks on HPC systems manually, but do so without necessarily receiving the support of a facility's extensive staff. Staff from facilities, as well as those working on large projects building data analysis infrastructures for teams, mentioned they receive requests from scientific users to directly support Jupyter on HPC systems. These individuals also noted that some interest

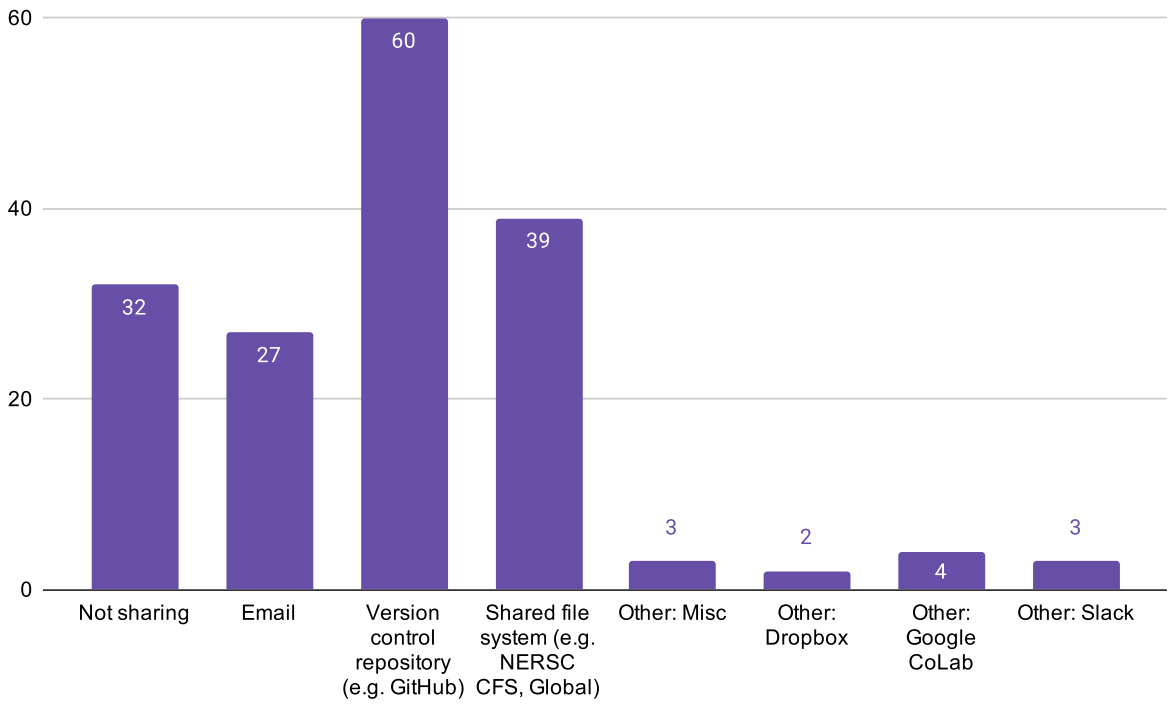

Figure 3: Multiple choice survey responses for how NERSC users share their Notebooks with colleagues. Respondents could select as many options as applicable from the four provided choices, and use the Other field to list any additional mechanisms that they use to share. Results were open coded and categorized by the first author, see Appendix B.

about Jupyter came up in conversations with their organization's senior management who have heard about Jupyter and other interactive tools from their different science constituents. Facilities are answering these requests and supporting work with Jupyter tools by adopting JupyterHub and JupyterLab to manage the user experience. Our findings illustrate this is being accomplished today with concerted effort to adapt these open sources tools to particular HPC ecosystems. This enables facilities and their users to take advantage of the various resources within the ecosystem, while meeting organizational policies associated with these resources. We examine how these adaptations are accomplished, and offer key takeaways relevant to other facilities contemplating adopting these tools.

Adapting JupyterHub to facility ecosystems. HPC and science user facilities are incorporating JupyterHub to support users who want to work in Notebooks across varied computing resources. We see that the flexibility of JupyterHub, and other Jupyter tools, is essential to its adoption in HPC and science user facility contexts. Our interviewees emphasized that the ability to successfully adapt and incorporate JupyterHub is due to its extensible design, collaborative efforts with colleagues at other facilities, and support from members of the Jupyter team to ensure customizations are built and maintained. Continuing to foster community among facilities, and share adaptations, is essential for these entities as they continue to incorporate this tool in their environments, particularly since their systems and policies present unique constraints.

JupyterHub's modular design helps facilities support multiple users running Notebooks on shared computing hardware. Interviewees from various HPC facilities explained that at their facility they setup JupyterHub to run on a dedicated infrastructure node with a landing page. A user can select a particular HPC system from the landing page, and JupyterHub is then responsible for orchestrating the user's Notebooks, and associated kernels, to run on the selected system. As part of configuring JupyterHub this way, facilities are also enabling automatic access to shared file systems, and user home directories, when they spin up a Jupyter Notebook — as opposed to user's manually launching Notebooks on an HPC system, then SSH tunneling to access it remotely. NERSC's Jupyter setup enables users with appropriate permissions to run Notebooks on

the Cori HPC, including high memory and GPU nodes, or the Spin service platform. Staff from other HPC facilities described similar setups. Multiple individuals noted that they have extended pre-existing JupyterHub authentication routines, and Jupyter kernel spawners, by customizing JupyterHub plugins for their unique configurations. For example, HPC facilities providing both unclassified and classified computing resources must address unique challenges. Users need to be able to run Notebooks in each type of environment, assuming their account has the appropriate permissions, so staff have to ensure complex security policies are met. In environments with resources dedicated to classified work, HPC facility staff need to be able to get Jupyter tools through security evaluations. Interviewees working with all types of HPC systems emphasized that security policies can be difficult for interactive computing tools since they often spin up web services. JupyterHub's extensibility helps ameliorate such challenges since staff can adapt the tools as needed for the environment, including scoping the parts of systems Jupyter is configured to access.

Staff at science user facilities explained how they typically construct data analysis infrastructures for their users. These individuals indicated they frequently provision a JupyterHub setup using virtual machines or containers on the resources their facility has available. These setups are built with a stack of software common to their user community's work already installed. Creating customized JupyterHub setups in this way ensures that users all have access to a consistent set of software, and less need to install different things manually. A staff member leading a project at a beamline user facility explained how they worked with the Project Jupyter team to customize their JupyterHub to spawn notebook kernels over SSH. Doing this enabled them to run the kernels on a centralized virtual machine setup, which makes it easier for staff to manage systems for end users. A developer from an astronomy observatory likewise explained that their project constructs data infrastructure using JupyterHub run in containers. This container-based JupyterHub configuration ensures that their diverse user base can run the project's software stack in environments ranging from HPC centers to private clusters, or the commercial cloud. Multiple individuals from science user facilities also explained how they leverage JupyterHub when running workshops, or other educational or outreach efforts. Using JupyterHub they create a custom environment so that attendees don't have to install full software stacks, but have the ability to create their own notebooks and interactively engage with the material being taught.

Across our discussions, interviewees from facilities noted that Project Jupyter presents JupyterHub, and other tools, using cloud computing oriented examples. Our interviewees emphasized that the different organization and operation of HPC systems can introduce challenges, especially compared with private clusters or the commercial cloud. This has necessitated the HPC community work together to build plugins and customizations for these types of computing environments (e.g., spawners and authenticators noted above). Various workshops have been held to bring members of the HPC and Jupyter communities together to identify and address problems, including one hosted by NERSC [12]. In addition, an openly accessible community forum [11] is also providing a space to share information and raise issues. These efforts illustrate how the HPC and science user facility community is leveraging the Jupyter community's openness, and the modular, flexible design of Jupyter tools in general.

Takeaways.

- *Customizing JupyterHub enables HPC and science user facility staff to ensure scientists using a facility's resources can smoothly run Jupyter Notebooks.*
- *Science user facilities, and large multi-institutional projects, can provision a core set of software for their community using a JupyterHub setup to make everything easily accessible in Notebooks.*
- *Stakeholders customizing JupyterHub by building plugins should continue growing a community through information and tool sharing at workshops, conversations on open source repositories through issues, and with existing forums. These efforts should continue engaging the Project Jupyter community, helping to ensure this open source effort has visibility into the unique constraints of HPC and science user facility environments. Additional workshops that bring facility stakeholders and members of Project Jupyter together can help identify emerging challenges and foster more growth of this sub-community.*

Providing JupyterLab as a default user interface. Configuring JupyterLab as a default interface presents a more user friendly entry point to HPC systems, and enables easier data browsing and terminal access, through a single web-based environment. Among our interviewees JupyterLab was discussed with a mix of

positive as well as ambivalent feelings. Multiple scientists felt that Notebooks were enough of a user interface for how they're doing work, and they were not quite sure what they really gained from JupyterLab as an individual researcher. Interviewees in developer roles, at times, negatively perceived JupyterLab as trying to be a more fully fledged development environment, without offering features that an IDE commonly offers. Interestingly, members of the Jupyter team noted their intention is not to create an integrated development environment. They are focused on building an interactive data exploration environment, and emphasized that fully-fledged software development is better conducted in other tools. Even with some stakeholder ambivalence or hesitation about JupyterLab, many staff members and scientists working at facilities noted that this interface is useful. When working on a complex HPC system with large datasets, the ability to explore data interactively through JupyterLab, when setup with the correct JupyterLab Extensions, helps them as they work without needing to shift between different tools. Interviewees building data analysis infrastructures for a project, or community, commented that being able to install JupyterLab Extensions helps them simplify their user's experience across computing systems. This continuity helps the scientists focus on the research at hand, rather than deal with variations in computing resources.

Facility staff and scientists both explained that they find JupyterLab Extensions helpful, particularly file browser customizations and viewers for different data formats. At the same time, each group noted challenges around the installation and management of JupyterLab Extensions in a shared JupyterHub environment. Currently a facility must install a JupyterLab Extension for all users when providing access through JupyterHub. Consequently, individual users cannot install a JupyterLab Extension on their own, or just for their own use. Facility staff noted that at times JupyterLab Extensions can conflict with each other when run in the same environment, due to their implementation in one JavaScript namespace. Furthermore, at least one facility staff member explained how hard it can be to find and hire someone with sufficient JavaScript expertise to customize or manage extensions. Multiple interviewees, and respondents to our NERSC survey, mentioned they would like to be able to customize the JupyterLab extensions available in their environment, but could not to their knowledge. It was unclear whether there were any processes at a particular facility for users to request a particular JupyterLab Extension, or to install one on their own. HPC or science user facilities could decide to provide particular projects, or teams, with customized JupyterHub plus JupyterLab installations. This would help these entities customize the entire setup for the collaboration in question, however no one interviewed in our study was doing this at the time. Some large projects teams working across individual facilities were however constructing their own setups, but this was not the common way of supporting work in our data.

Takeaways.

- *The design of JupyterLab provides a simplified environment for HPC users to access resources while working in Notebooks, including having access to file browsers, terminals, and JupyterLab Extensions.*
- *Facilities should develop and communicate processes for users to request the installation or management of JupyterLab Extensions.*
- *Facilities should incorporate user feedback in any processes for selecting JupyterLab Extensions to be installed, perhaps allowing individuals to vote for options they would find useful.*
- *HPC facilities might consider provisioning custom JupyterHub plus JupyterLab setups for large collaborations. This would help these projects run specialized instances with a pre-installed software stack for their community. This would require sufficient human resources to manage in a sustainable manner.*
- *The Jupyter project could consider whether it is possible to isolate JupyterLab Extensions installed on one JupyterHub. This might be a pathway to allowing individual users to determine which JupyterLab Extensions they would like to install and run.*

4.3 User feedback on Jupyter at NERSC

The Jupyter installation at NERSC supports a variety of users at the facility. We gathered feedback about this installation during our interviews and more broadly through our survey. Here we examine what users find does and does not work well with Jupyter at NERSC.

4.3.1 What works well with NERSC's Jupyter Service

Interviewees were glad to have this Jupyter setup available as they do their work, even when it comes with some challenges. Feedback gathered through our survey included praise and constructive comments, as well as frustrations and complaints. Furthermore, NERSC users were at times unaware of some available features with the Jupyter installation in this ecosystem.

Managing a Jupyter installation. Participants were happy that Jupyter is setup at NERSC and being actively managed and supported. Having the JupyterHub setup managing access to various systems, rather than having to manually spin up a Notebook and kernel, then SSH to a system node, was valued. Many participants also find that Jupyter Notebooks and/or JupyterLab make accessing an HPC like Cori, with its varied powerful nodes, easier due to the relatively simple user interfaces these tools offer. In particular, since the JupyterLab user interface lets people have a file browser, multiple Notebooks, and other artifacts open in one window, users find it easier to access various items while doing work.

Filesystem and database access. Participants were most frequently happy that NERSC's Jupyter setup provides a simple, friendly way to access the various file systems and databases available at the facility. Users find it helpful to be able to access and explore large datasets hosted at NERSC through the JupyterLab file browser. Respondents noted they would not be able to store such large datasets on their personal machines, so exploring and working with them through NERSC's Jupyter setup makes their work possible. Respondents also mentioned the ease of accessing various hosted databases through the Jupyter setup. For individuals working as part of collaborations with large datasets, having NERSC as a single place that hosts the data, then makes it accessible via Jupyter, makes it easier for their colleagues to do work and contribute to results. Finally, individuals noted that plotting was also easier in the NERSC Jupyter environment since the data was all accessible within Notebooks.

Pre-built environments and software stacks. Respondents appreciated the various pre-built Jupyter environments that NERSC provides with its installation. These pre-built environments make it easy to focus on data work, rather than setup and configuration work for a task. Respondents working as part of large collaborations appreciated having easy access to the software stacks that these entities produce and provide at NERSC. Not having to install and maintain the stack on their own machines, combined with access to the large datasets stored on the file systems, makes their use of collaboration resources simpler.

Documentation and support from NERSC staff. A few responses specifically noted that support from the NERSC staff managing Jupyter has been helpful as they try to do their work. The documentation available was also mentioned by a few users as a helpful aspect of NERSC's Jupyter installation.

Access to GPU and large memory nodes. A few respondents mentioned that access to GPU or large memory nodes via Jupyter at NERSC is helpful for their work. Some users indicated they wished they had access to GPUs at NERSC, suggesting expanding this service will be helpful to the community.

Batch job submission. A few respondents indicated that the ability to submit batch jobs via the NERSC Jupyter setup was helpful.

Takeaways.

- *Participants find NERSC's Jupyter setup helpful since it is an easy way to interact with HPC systems, and large datasets, while leveraging pre-built software stacks and environments.*
- *Participants specifically mentioned support from NERSC staff, including documentation, as a positive aspect of the facility's Jupyter service.*
- *Pre-built and customized environments ease the complexity of software stacks for scientific users of all backgrounds.*

4.3.2 Challenges and feature requests for Jupyter at NERSC

Along with positive feedback, various challenges emerged that users face with NERSC and Jupyter tools. Some challenges were a result of larger NERSC issues, some from the design of Jupyter tools, and some the

intersection of the two. Users commonly noted feature requests that would be helpful if trying to solve some of these challenges.

Outages, crashes, and speed of the systems. The most frequently mentioned challenges from users focused on the outages, crashes, and slowdowns they face with using NERSC's Jupyter setup. Even with three Cori login nodes dedicated to Jupyter, respondents indicated that their work could easily crash or slow down if one user overloaded the system by using up all of the memory, or not properly shutting down their tasks. Some individuals also noted that lags with NERSC's different file systems can cause problems when they are trying to do interactive work. Crashes and slowdowns are inherent to shared systems, but become very visible in interactive computing where limited parts of an HPC system are made available.

Aside from crashes or slow downs users also found the outages caused by NERSC maintenance windows, or unexpected events, frustrating. These events are beyond the scope of the Jupyter service alone, but do point to a challenge for scientists expecting to be able to do work whenever needed on production systems that do have downtime.

Documentation for environments and parallel computing. Documentation is an essential aspect of a user's experiences, and their ability to effectively use systems. Users raised challenges with the documentation available today for Jupyter at NERSC. For example, it was not always clear how, or when, users should change the pre-configured environments for use with Notebooks. Some NERSC users indicated that they would benefit from more documentation describing "best practices" for using Jupyter tools at the facility. Example requests included: general advice on structuring a Notebook for data analysis when using NERSC database services; how to install or create custom environments; and how to easily launch and manage batch processing jobs.

The most common challenge noted was a lack of documentation, or guidance, on setting up and running parallel computing tools with Notebooks, or JupyterLab, in the NERSC environment. Some users were uncertain how to run Dask parallel computing tools, or comparable products, through the NERSC JupyterHub setup so that they can spawn jobs on Cori compute nodes. A helpful form of documentation for some respondents would be pre-configured Notebooks or environments illustrating how they could run parallel tools (e.g., mpi4py or Dask) interactively from Jupyter. These tutorial Notebooks would be a chance for NERSC to illustrate best practices for unfamiliar users. In addition, increased integration with existing schedulers (e.g. SLURM) was mentioned as a desirable feature.

GPU node access. Some survey respondents were not aware that, on request, NERSC currently provides access to Cori GPU nodes. Consequently, a couple of individuals specifically noted that access to GPUs would be great for their work. It was unclear if these individuals were unaware of the Cori GPU nodes, or simply didn't have access at the time. This is likely a short-term issue since the Perlmutter system will provide increased access to this type of resource, however it illustrates that an HPC facility's roll out of novel features can easily, and likely unintentionally, exclude interested users.

Quirks with JupyterLab workspaces and user interfaces. A few issues and quirks with the JupyterLab user interface and Notebooks emerged from our data. Some respondents noted that they use multiple workspaces across different projects. Switching among these workspaces has to be accomplished currently by manually changing the URL in a web browser. Individuals noted they would find an interface dedicated to managing and switching between workspaces helpful. In addition, various requests related to the JupyterLab user interface included: the ability to search recently accessed directory paths, and better keyboard shortcut support in JupyterLab. For example, users reported that Ctrl+C doesn't work in the JupyterLab terminal.

Version controlling Notebooks. Participants sharing their Notebooks frequently mentioned using a version control system to accomplish this task. Many interviewees expressed frustration with trying to version control Notebooks. Since Notebook files are a mix of plain text and binary data, gibberish is generated when calculating differences between two versions of a Notebook. Interviewees were often unaware of the Jupyter Project's tool *nbdime* for diffing and merging Notebook files. This tool produces more useful outputs by accounting for the varied types of data encapsulated in Notebooks. While this is not a challenge caused by NERSC, it is one where making this tool visible as part of the facility's service could ameliorate some user frustrations. Increasing awareness could be accomplished through incorporation in documentation or

tutorials, or by having a JupyterLab Extension installed for all users.

Sharing Notebooks plus their environments. Individuals sharing Notebook files also need to share a functioning environment with specifications declaring the packages necessary to spin up and run a kernel. A few respondents specifically noted that it is not straightforward to share a Notebook's environment with other users in the NERSC ecosystem. It is unclear what approach these individuals were taking, but facilities could provide a tutorial as part of their documentation. This type of tutorial could lay out the best way of constructing a Notebook and environment, so that they can be shared among users at the facility, and with users in other computing ecosystems.

Today, the Binder tool facilitates Notebook sharing by providing users with one click execution of a version controlled notebook using a cloud-based container. HPC facilities do not yet have a similar offering based on our discussions. This is an area for further development, with one prototype being developed by our team [26]. Adapting and customizing Binder so that users can share Notebooks, and run them with one click at a particular facility, could improve scientist's ability to share their work with colleagues. Ideally the data underlying these Notebooks would be accessible as well, since this is a key reason why users find HPC Jupyter installations helpful. This would facilitate collaboration among colleagues with minimal setup and configuration hassles. This would also be beneficial for collaborations where some members wish to interactively work with a data analysis workflow, but not engage in detailed development or setup work.

Challenges sharing large datasets along with Notebooks. Notebook files can be shared in a variety of ways, as we've seen our respondents are already doing. The larger challenge scientists face is sharing access to the large datasets needed to execute a Notebook, particularly with users from different institutions. Having access to large datasets on shared filesystems (which users already appreciate) is not sufficient when more complex modes of Notebook sharing come into play.

Survey participants valued the easy access to the large datasets stored on NERSC filesystems. At a facility like NERSC, Notebooks with the location of datasets encoded can be shared fairly easily by teams or projects working primarily, or solely, with that one facility's resources. Projects with members who do not have access to a particular facility's resources face challenges sharing Notebooks, and the data they require. NERSC users find sharing Notebooks and the large quantities of data they rely upon at NERSC with colleagues from other institutions challenging if those individuals do not have NERSC accounts (or the ability or desire to acquire one). Sometimes sharing the Notebook alone may be helpful, but for the most effective collaboration these researchers have to find a way to make the Notebook, its environment specification, and the necessary data available outside of NERSC's ecosystem. One bioinformaticist explained that, rather than work to obtain NERSC access for a collaborator, they found it easier to setup a JupyterHub and host data in the commercial cloud. Creating this setup, they could easily collaborate with a colleague by sharing Notebooks with full access to a required dataset, in an environment they completely control. This approach is not feasible for all researchers, or projects, but demonstrates the lengths some will go to when trying to facilitate smoother collaboration with colleagues across institutions.

Takeaways.

- *Various challenges identified here (e.g., crashes and the speed of the Jupyter service, varying awareness of Cori GPU nodes, etc.) highlight the balancing act facilities must do provisioning resources for shared interactive computing systems. In the case of NERSC, the number of Cori login nodes dedicated to Jupyter has expanded over time. At the same time access to Cori GPU nodes is not universal. For facilities procuring new, or adjusting existing, systems to support more interactive computing, it is worth discussing how much of the available computing infrastructure different services should be allocated. Developing mechanisms to dynamically adjust based on user demands may also be a way to help scale interactive services, like JupyterHub, in real time as users increase.*
- *Facilities should provide sufficient documentation about nuances of their Jupyter, or other interactive computing, setups. This will help users understand what may be different from using Jupyter setups in other environments. Documentation may partially take the form of Notebooks as interactive tutorials. These Notebooks could explain how to do HPC specific work in Jupyter, for example how to manage parallel computing jobs from within a Notebook.*

- *Sharing Notebooks that rely upon large datasets is complicated if collaborators do not have access to the same computing resources or facilities. The HPC and science user facility communities could collaboratively dedicate efforts to adapt Binder, or similar tools, to make one click Notebook sharing simpler to achieve in HPC ecosystems. Funding agencies could sponsor these adaptations to ensure reproducible, shareable, interactive data science is easier to accomplish with HPC systems.*
- *The variety of products emerging from complex ecosystems can be challenging to keep track of for new (and existing) users who are focused on their own work. The challenges we identified for users version controlling Notebooks surfaced how variable awareness of the nbdime tool is among scientists. HPC and science user facilities, as well as scientific open source projects, need to help scientific users become aware of the various tools available and how to effectively use them in their own work. Scientists may need to be made aware that a tool exists and can solve a problem for them, or how to install and integrate it into their ecosystem. Outreach and communication efforts among scientific open source communities, HPC and science user facilities, and particular scientific communities are one way to educate each constituency about needs and opportunities to better support data science work.*

5 Summary

Our user research engaged with individuals who use or support Jupyter at HPC and science user facilities. Our focus was on better understanding scientific needs and requirements for interactive and reproducible computational work with large datasets in these ecosystems. We interviewed thirty-two researchers, computer scientists, and Project Jupyter developers in 2019, and subsequently surveyed NERSC Jupyter users between March - April 2020 (receiving 103 responses from over 900 users contacted). Prior human-centered research examined the ways data scientists work with Notebooks, and the design challenges that come with creating usable, supportive research tools. Our study shifted focus, and delved into more overarching issues with the ways different facilities are supporting their scientific user's work with Jupyter tools, how scientists are using Jupyter Notebooks to accomplish data science work, and illustrated gaps in the alignment of Jupyter tools and the ecosystems created and sustained by HPC and science user facilities.

Our findings affirm that individuals in HPC and user facility ecosystems are using Jupyter Notebooks for interactive analysis of data, exploratory code development, and visualization of results. Facilities providing a JupyterLab interface, with a JupyterHub backend, are making it easier for users to leverage HPC resources, access large datasets on shared file systems, and get work done in a single place. NERSC survey respondents are using Jupyter frequently, with 41% reporting they use NERSC's service daily, and 35% a few times a week. Interviewees responsible for supporting interactive computing services at facilities underscored the utility of JupyterHub and JupyterLab, whether at an HPC or science user facility, or as part of a large multi-institutional project. These open source tools enable these individuals to build and provision interactive data analysis infrastructures, all while meeting facility or project policies with minimal impact to users.

The takeaways throughout this report illustrate many opportunities for facilities, and open source science software projects, to collaboratively work to improve scientist's ability to do work with these important resources. Facilities and open source projects are creating and sustaining vibrant, complex ecosystems of tools and resources. Supporting scientists, with varying computing backgrounds, requires more than just making common tools available. Facilities, funding agencies, and communities need to be ensuring resources and tools are as visible to scientists as possible. Learning that individuals struggle with a problem (like version controlling notebooks) and are unaware of a community-supported solution underscores the need to continue outreach and awareness efforts. Challenges particularly remain as users work with larger datasets collaboratively, especially when scientists require the ability to share Notebooks with colleagues who work in different environments. Existing practices and tools can sometimes get the job done, but as the scale and scope of work expands cracks are emerging. Addressing these issues will require concerted effort among facilities, teams, and funding agencies if Notebooks are to truly enable more shareable reproducible scientific work. This is an area for further research that can be informed by the variety of takeaways offered in this report, and in part illustrate the complexity of bringing together diverse ecosystems.

6 Acknowledgements

The authors wish to thank our study participants for their insights and feedback, as well as the members of the Usable Data Abstractions team, including Fernando Perez, Shreyas Cholia, Devarshi Ghoshal, Matthew Henderson, and Lindsey Heagy. This work is supported by the U.S. Department of Energy, Office of Science and Office of Advanced Scientific Computing Research (ASCR) under Contract No. DE-AC02-05CH11231.

References

[1] Advanced Light Source. ALS - Advanced Light Source. `https://als.lbl.gov/`. pages 2

[2] Atmospheric Radiation Measurement User Facility. ARM Research Facility. `https://www.arm.gov/`. pages 2

[3] Charmaz, K. *Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis*, 2nd ed. Sage, 2014. pages 19, 23

[4] Chattopadhyay, S., Prasad, I., Henley, A. Z., Sarma, A., and Barik, T. 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* (New York, NY, USA, 2020), CHI '20, Association for Computing Machinery, p. 1–12. pages 4

[5] Dask core developers. Dask: Scalable analytics in Python. `https://dask.org/`. pages 6

[6] Emerson, R. M., Fretz, R. I., and Shaw, L. L. *Writing Ethnographic Fieldnotes*. The University of Chicago Press, Chicago, IL, 1995. pages 19, 23

[7] ESnet. Energy Sciences Network Home. `https://www.es.net/`. pages 3

[8] Google. Welcome To Colaboratory. `https://colab.research.google.com/`. pages 7

[9] Head, A., Hohman, F., Barik, T., Drucker, S. M., and DeLine, R. Managing messes in computational notebooks. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2019), CHI '19, Association for Computing Machinery. pages 4

[10] IPython Development Team. Interactive Parallel Computing with IPython. `https://github.com/ipython/ipyparallel`. pages 6

[11] Jupyter at Research Facilities. Jupyter at Research Facilities. `https://groups.google.com/g/jupyter-research-facilities`. pages 10

[12] Jupyter Community Workshop. Jupyter Community Workshop. `https://jupyter-workshop-2019.lbl.gov/`. pages 10

[13] Kery, M. B., John, B. E., O'Flaherty, P., Horvath, A., and Myers, B. A. Towards effective foraging by data scientists to find past analysis choices. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2019), CHI '19, Association for Computing Machinery. pages 4

[14] Kery, M. B., and Myers, B. A. Interactions for untangling messy history in a computational notebook. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (2018), pp. 147–155. pages 4

[15] Kery, M. B., Radensky, M., Arya, M., John, B. E., and Myers, B. A. The story in the notebook: Exploratory data science using a literate programming tool. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2018), CHI '18, Association for Computing Machinery. pages 4

[16] National Energy Research Scientific Computing Center. Cori. `https://docs.nersc.gov/systems/cori/`. pages 3

[17] National Energy Research Scientific Computing Center. National Energy Research Scientific Computing Center. https://www.nersc.gov/. pages 3

[18] National Energy Research Scientific Computing Center. Perlmutter. https://www.nersc.gov/systems/perlmutter/. pages 3

[19] National Energy Research Scientific Computing Center. Spin. https://www.nersc.gov/systems/spin/. pages 3

[20] Project Jupyter. nbdime Jupyter Notebook Diff and Merge tools. https://github.com/jupyter/nbdime. pages 3

[21] Project Jupyter. Project Jupyter Home. https://jupyter.org/. pages 2, 3

[22] Project Jupyter. The Binder Project. https://jupyter.org/binder. pages 3

[23] Randles, B. M., Pasquetto, I. V., Golshan, M. S., and Borgman, C. L. Using the jupyter notebook as a tool for open science: An empirical study. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)* (2017), pp. 1–2. pages 4

[24] Rule, A., Drosos, I., Tabard, A., and Hollan, J. D. Aiding collaborative reuse of computational notebooks with annotated cell folding. *Proc. ACM Hum.-Comput. Interact. 2*, CSCW (Nov. 2018). pages 4

[25] Rule, A., Tabard, A., and Hollan, J. D. Exploration and explanation in computational notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2018), CHI '18, Association for Computing Machinery. pages 4

[26] Usable Data Abstractions Team. Scaling and Sharing Analysis Work from Desktops to HPC Systems with Jupyter. https://dst.lbl.gov/projects/uda/#/jupyter. pages 14

[27] Vera Rubin Observatory. Rubin Observatory. https://www.lsst.org/. pages 2

[28] Weiss, R. S. *Learning From Strangers: The Art and Method of Qualitative Interview Studies*. The Free Press, New York, NY, 1995. pages 5

[29] Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., Gonzalez-Beltran, A., Gray, A. J. G., Groth, P., Goble, C., Grethe, J. S., Heringa, J., 't Hoen, P. A. C., Hooft, R., Kuhn, T., Kok, R., Kok, J., Lusher, S. J., Martone, M. E., Mons, A., Packer, A. L., Persson, B., Rocca-Serra, P., Roos, M., van Schaik, R., Sansone, S.-A., Schultes, E., Sengstag, T., Slater, T., Strawn, G., Swertz, M. A., Thompson, M., van der Lei, J., van Mulligen, E., Velterop, J., Waagmeester, A., Wittenburg, P., Wolstencroft, K., Zhao, J., and Mons, B. The fair guiding principles for scientific data management and stewardship. *Scientific Data 3* (2016), 160018. pages 4

Appendix A Interview Methods

A.1 Interview Protocols

A.1.1 Protocol A for Scientists and Computing for Science Individuals

Thank you for taking time to talk. Our project is examining the different ways scientific researchers and developers use Project Jupyter tools with different computing platforms. To begin,

1. Can you tell me a bit about your research/work overall?
   (a) What would you consider your particular community?
   (b) **DEV:** What kinds of systems/tools do you work on generally?
2. **SCIENCE:** Can you tell me a bit about the data that you use?
   (a) Where does it come from?
   (b) Where is it stored typically?
   (c) Is this storage site separate from the compute resources you have?
3. How did you first start using Jupyter?
   (a) What kinds of problems did it help you solve?
   (b) Why do you continue to use Jupyter?
4. **SCIENCE:** How common is Jupyter in your community?
   (a) How is Jupyter supporting your community?
5. Are you using Jupyter on clusters, the cloud, or HPCs?
   (a) How are you doing this?
   (b) How does your Jupyter usage vary among these environments?
   (c) How do you move work between these environments?
   (d) What challenges emerge in this work?
6. **DEV:** How are you supporting Jupyter in HPC environments?
   (a) What challenges are emerging?
7. **DEV:** How have you customized Jupyter for your HPC environment?
   (a) What tools have you leveraged? Built?
8. What do you wish Jupyter could help you do on HPCs?
9. How have your Jupyter needs and use changed over time?
10. How do you see Jupyter's impact on the reproducibility of your work?
11. Where do you see the Jupyter ecosystem going in the next few years?
    (a) How do you think this will impact your own work?
12. Is there anything else you would like us to know?

A.1.2 Protocol B for Project Jupyter team members

1. Can you tell me a bit about your research/work overall?
   (a) What kinds of systems/tools do you work on generally?
2. Can you tell me how you started contributing to Jupyter?
   (a) What attracted you to the project?
   (b) What motivates you to continue beining involved/contributing?
3. What do you think is unique about Jupyter?
   (a) What kinds of work do you think Jupyter is best suited to?
   (b) What kinds of work do you think Jupyter doesn't work for?
4. How do you think Jupyter fits in different ecosystems currently?
5. What experiences have you had engaging with Jupyter users?
   (a) What has worked well?
   (b) What difficulties have they faced?
6. Where do you see the Jupyter ecosystem going in the next few years?
   (a) What do you hope to contribute in the future?
7. Is there anything else you would like us to know?

A.1.3 Analysis Process

Analysis of our interview data used a qualitative open coding approach where the first author read each transcript and identified segments of conversation to apply codes to which capture an idea or concept emerging from this data [6]. Through this open coding process and constant comparison themes were identified and pulled out to write explanatory memos about in a grounded theory-like process [3]. Our qualitative coding resulted in 84 unique codes that were gathered into eight overarching themes. Memos were written about codes and themes to produce our understanding of the ways Jupyter tools are used and incorporated into HPC environments. The insights abstracted from these codes and themes provide the basis for the findings presented in this report and informed the development of our descriptive survey.

Theme	*Example Codes*	*Description*
Cloud Computing (2 Codes)	• Economic concern w/ potential usage • Usage today	Theme capturing particular discussion points about commercial cloud usage and challenges
Data (4 Codes)	• Provenance • Quantity using w/in Jupyter • File format oddities	Discussion points related to work with data of different types
HPC (5 Codes)	• Facility connection w/ Jupyter project • Facility's user needs changed or not • Compute nodes / batch processing w/ Jupyter	Codes raising HPC facility specific issues or concerns

Theme	*Example Codes*	*Description*
Jupyter (25 Codes)	• Usage as IDE • Collaboration features/potentials • History/future trajectories • How users collaborate around notebooks • Utility for teaching &/or reproducibility • Sharing notebook + environment	Codes raising specific issues with Jupyter ecosystem tools, issues with particular features or ways of working, social & political challenges with the project or funding
Misc. (7 Codes)	• Citizen science • Shift in hardware types • Sustainability of tools/projects	Miscellaneous codes covering topics surrounding Jupyter usage, open science projects, or interviewee's work
NERSC (3 Codes)	• Jupyter history/setup • Kale project • Perlmutter anticipation	Codes resulting from issues specific to the NERSC facility
Protocol Questions (36 Codes)	• Challenges supporting Jupyter installs • Data - where its stored • Dev - how Jupyter fits different ecosystems currently • First start using Jupyter • How customized - tools leveraged	Codes representing the questions from the interview protocols that surface particular issues examined and complement other themes and their codes
Z Meta/Misc (2 Codes)	• Juicy • Useful quote	Two meta or miscellaneous codes used by the first author to specifically identify interesting question or issues alongside other focused codes

Appendix B Survey Methods

B.1 Process Overview

Our sixteen question survey was split across four sections and used a variety of free response and multiple choice questions. The survey was distributed as a Google Form that was open to responses from March 26 - April 26 2020 ($\sim$ four weeks). The survey was advertised through three mechanisms. First, the NERSC Jupyter administrator directly emailed 964 recent active users of their Jupyter installation on March 26, 2020. Second, an advertisement for the survey was included in the NERSC users weekly email on March 30th and April 6th, 13th, and 20th that reaches the 7,000 plus registered users of NERSC resources. Third, the NERSC Jupyter administrator added a banner to this setup from April 6 - 26 that was visible when users logged in. This banner is commonly used to display messages about upcoming maintenance events to users.

B.2 Survey Questions

Section 1: Project Description & Human Subjects Agreement

- Question 1: Checkbox to affirm agreement with human subjects terms
  - Required to proceed with survey

Section 2: General Jupyter Usage

- Question 2: How long you used Jupyter? (days, months, years)
  - Short answer free response
- Question 3: How do you use Jupyter in your work and what influences you to use these tools? (E.g. data analysis with notebooks provided by a collaboration, developing & testing machine learning models, etc.)
  - Long answer free response
- Question 4: How do your organize your work when you're using Jupyter notebooks? (Please select all that apply)
  - Multiple choice with following options:
    - One notebook per analysis
    - One notebook for multiple analyses
    - Multiple notebooks for one analysis
    - Multiple notebooks for multiple analyses
    - Other
- Question 5: How are you sharing notebooks with colleagues? (Please select all that apply)
  - Multiple choice with following options:
    - Not sharing
    - Email
    - Version control repository (e.g. GitHub)
    - Shared file system (e.g. NERSC CFS, Global)
    - Other
- Question 6: How often do you use Jupyter in each of these contexts? (please check all that apply)
  - Checkbox grid with following options:
    - Frequency in rows:
      - Daily
      - A few times a week
      - Once a week
      - A few times a month
      - Once a month
      - A few times a year or less
      - Never
    - Computing environments in columns:
      - Personal machine (laptop, desktop)
      - NERSC HPC (e.g. Cori)
      - Other HPC Facility (specify if so)
      - Private Cluster
      - Commercial Cloud (e.g. AWS, Azure, GCP, etc.)
      - Other Systems (specify below)
- Question 7: If you selected an answer for "Other HPC Facility" or "Other System" in the previous question please explain what the additional contexts are where you use Jupyter.
- Question 8: Do you use Jupyter across multiple systems? (e.g. running a notebook at NERSC then connecting to other systems, running different notebooks on multiple systems at once). If yes, describe how.
  - Short answer free response

Section 3 – Jupyter Usage at NERSC

- Question 9: How long have you used Jupyter at NERSC? (days, months, years)
  - Short answer free response
- Question 10: How often do you use Jupyter at NERSC for these activities?
  - Checkbox grid with following options:
    - Frequency in rows:
      - Daily
      - A few times a week
      - Once a week
      - A few times a month
      - Once a month
      - A few times a year or less
      - Never
    - Activities in columns:
      - Exploring data stored on one of the filesystems
      - Analyzing data interactively
      - Developing a software package
      - Machine Learning workflows
      - Executing batch processing jobs
      - Other (specify below)
- Question 11: If you answered Other in the previous question, please explain what additional activities you use Jupyter for at NERSC.
  - Short answer free response
- Question 12: How long do you run your notebooks at NERSC? (e.g. just while exploring data, while a full analysis runs, for extended periods of time, etc.)
  - Long answer free response
- Question 13: What works well when using Jupyter at NERSC?
  - Long answer free response
- Question 14: What does NOT work well when using Jupyter at NERSC?
  - Long answer free response

Section 4 – Wrap-Up

- Question 15: Is there anything else you would like us to know? (feature requests, other Jupyter issues you face, etc.)
  - Long answer free response
- Question 16: (Optional) If you are willing to have us contact you to learn more about your work with Jupyter at NERSC please provide your name and e-mail address. This information will remain confidential to the DST and NERSC Jupyter team and will not appear in any reports.
  - Short answer free response

B.3 Cleaning & Analysis Process

Once the survey was closed to responses the first author cleaned and analyzed this data in a Google spreadsheet. This process involved identifying and removing one individual's duplicated response (two rows of exactly the same answers), normalizing free response answers for temporal questions (Q2 & Q9), plotting of responses to multiple choice questions, and qualitative open coding of free responses to identify core themes in responses [3,6]. Specific data cleaning decisions are listed below for each question.

Question	Data Cleaning Notes
1	Required to be yes to continue, no data generated here
2	• Users entered varying amounts for how long, from months to unclear numbers of years • Cleaned and normalized/categorized as follows: – All numbers converted to years for counts ∗ If months provided then divided out of 12 – "more than" X years ∗ Written as X+ ∗ e.g. "more than 5 years" became "5+" – "years" left as is – "months" left as is – "about X" rounded to X – "I started with IPython in Fall 2013" ∗ Rounded to 6.5 based on time of year response was provided – "About a year" ∗ Rounded to 1
3	• Free response question that was open coded/categorized • Ended up coming up with 10 codes for the free responses to how they use Jupyter – Exploring Data (e.g. on the filesystem, a database) – Data analysis including pre/post processing of data – Prototyping (including any basic code testing/exploration under this category) – Plotting / visualizing data – Modeling in a tradition sense of some phenomena – Machine Learning (ML) work of any type, including ML model training – Batch/HPC jobs - including setting up, kicking off things, and monitoring – Sharing results – Sharing code – Teaching (whether collaborators or in a traditional educational setting) • Very few of the responses included an answer to what influences them to use Jupyter so we ended up not coding this very much – Among the codes we did bother to generate before stopping were: ∗ Easy to use ∗ Inline figures ∗ Interactivity / Quick turnaround time ∗ Easy to share ∗ Ability to manipulate data easily ∗ Easy to debug individual cells – In the future if asking about influences or reasons for using a tool need to ensure that this is a distinct question of its own

Question	Data Cleaning Notes
4	• Provided a multiple choice question with the following options – One notebook per analysis – One notebook for multiple analyses – Multiple notebooks for one analysis – Multiple notebooks for multiple analyses – Other (free response) • Cleaning this up – One person used other to say "all of the above" so that was counted in every other category – A few people checked an option then explained further via the Other field ∗ "Typically one notebook for one simulation type (one simulation can have multiple parts)" · Went ahead and categorized as "One notebook per analysis" by interpreting simulation as such. This is a loose interpretation though. ∗ "One notebook per analysis, Multiple notebooks for one analysis, Varies. I try to use a consistent template for all notebooks with a header, imports, main code, and appendix section" ∗ "Multiple notebooks for multiple analyses, Usually I'd structure the analyses first then decide how many notebooks to use" – A few used Other and did not check our provided choices. We did not categorize these into our provided options ∗ "One notebook per project, which may or may not require multiple analyses" · Did not categorize as any of our default choices since this is unclear ∗ "I find my work in notebooks to be generally unorganized." – Since people could check multiple boxes, we went ahead and produced a count of how many options each respondent checked to give us a sense of how frequent multiple patterns of using notebooks are
5	• Multiple choice question with following options – Not sharing – Email – Version control repository (e.g. GitHub) – Shared file system (e.g. NERSC CFS, Global) – Other (free response) • While cleaning we ended up categorizing the Other entries. These came to include: – Other: Misc – Other: Dropbox – Other: Google CoLab – Other: Slack • Since respondents could select multiple options we did a count of how many each person checked.
6	• This was a gridded checkbox where frequency was the rows and computing contexts the columns • Respondents could check multiple for each row and column • Cleaning was not necessary
7	• A fair number of respondents listed other systems they use. • We open coded and counted up the number of mentions for every system

Question	Data Cleaning Notes
8	• This was a free response question so we categorized responses as: – No – Yes w/ explanation ∗ The explanations we broadly ended up also categorizing as a mix of PC & HPC/Cluster/Cloud ∗ No Response
9	• This was a free response question asking for an amount of time • Cleaning notes are similar to Q2 but slightly different due to shorter timespans in these responses – All numbers converted to years for counts – If months provided then divided out of 12 "more than" X years ∗ Written as X+ ∗ e.g. "more than 5 years" became "5+" – "years" left as is – "months" left as is – "about X" rounded to X – Less than 3 months ∗ Rounded to 2 months
10	• Multiple choice checkbox grid question where the rows were frequency and columns different activities • Respondents could select multiple options for each row and column • An Other option asked for followup information in Q11 • Cleaning was not necessary for these responses
11	• Out of our total of 103 responses only 3 provided something here • Responses were simply noted and not cleaned up or categorized

Question	Data Cleaning Notes
12	• Free response question that we open coded • We ended up with these codes: – Juicy – my way of noting responses we found interesting and want to revisit – Interactive Use (real time to short term, up to hours maybe day long length) – Extended Use (letting run for days, weeks, etc.) – Full analysis run (split this out since that can mean many things) – Other • Coding notes: – Some responses may be categorized multiple times thanks to answers provided – Single Other case was just nonsensical response to question – "Full analysis run" was distinguished since what this means is vague but multiple people explicitly stated it. – Some of cases categorized as extended usage is really people just leaving a notebook running then working in it day after day. However the exact split of interactive versus long-term was hard to define and we erred on the side of extended usage here. • The line between Interactive and Extended use could get fuzzy since some respondents say they leave their notebooks open and available for days but it was unclear whether things were being computed all of that time. Some were explicit that they just leave it up until it's shut down by NERSC to have easier access to where they were working. • The one response I coded as Other was incredibly odd and I wasn't sure what to make of it but I feel like they misinterpreted the question: – "A few times a month."
13	• Free response question that we open coded • We ended up with many codes as follows: – Juicy – Misc / Other – IT Help – Jupyter Itself – Works as Intended (meets expectations) / No Issues – Documentation – Nice UI w/ access to many things – Easy to access – Environment / Kernel Resources Available – Filesystem Access – Database Access – Fast Access 2 Data – Job Submission – Powerful Nodes – Plotting – Collaboration Software Setup Available – Exclusive CPU Use – GPU Access – High Uptime / Availability – Login/SSO

Question	Data Cleaning Notes
14	• Free response question that we open coded • We ended up with many codes as follows: – Juicy – No problems – Misc / Other (not NERSC, more Jupyter issue) – Documentation – Difficulty Sharing – Bugs or Unclear errors – Tab Completion Slow – Missing keyboard bindings – Extension support issues – Login Issues – Slow startup – Kernel Hangs / Restarts – Env Quirks – Maintenance interruptions / Outages / Stability – Long Job Execution Time – Scaling / Speed – GPU Node Issues – Long executions of notebooks – Custom packages are annoying to setup – Idle Disconnect – Lag / Disk Latencies – Julia support – Dask / Parallel Challenges
15	• Free response so we open coded and categorize some common things mentioned • Codes ended up being: – Positive Comment / Thanks – Misc – Jupyter Issues Generally – GPU Nodes – Extensions / Plugin Flexibility – Stability / Reliability / Availability – Parallel Help – Best Practices / Documentation – Sharing / Collaboration – Julia support
16	Left as is for respondents who provided contact information