

1. Ease of use :

- a. Were there any actions you found difficult to do in Jupyter Notebook?
  - i. I am pretty comfortable with Jupyter Hub and Jupyter Notebooks, so there weren't any difficulties this time. I've always enjoyed Jupyter Notebooks though, even from the beginning. There were not any actions that I struggled with. Making sure that the person runs the cell that includes libraries is important because if they start up their server and just try to run the code cell they are working on it most likely won't work.
- b. Any elements of the UI that seemed "unintuitive", in that you weren't sure what their purpose or function was?
  - i. Dragging the cells around was something that was a little "unintuitive" for me. For example, if you want to modify a notebook by making a cell at the very top, it can be difficult because the normal new cell option will put it below the one you selected. At first, I would have to make a new cell under the spot that I wanted it to go, then I'd copy info from the existing cell into the new one and then put the information I wanted to add in the old cell. Then I realized you can click and drag the cells around to move them.

2. Github

- a. What interface did you use to upload your Notebook to Github?
  - i. I just used the GitHub website because I'm just adding one notebook. I wasn't concerned about doing multiple pushes while creating the notebook, so I just stored it on JupyterHub until it was completed. Then I uploaded the completed notebook to the new GitHub repository. I tend to stick to the website because it is the simplest thing to use. I have the local install on my computer, but I do not use it very often.
- b. What, if any, issues did you have with setting up your repository and/or uploading your code?
  - i. It wasn't an issue this time because I already struggled through the learning process before. I think GitHub definitely requires a bit of training to use correctly, especially if you've never used it before.

3. Programming in Jupyter Notebook

- a. How did you feel about writing code in Jupyter Notebook?
  - i. Good, I'm very familiar with Jupyter Notebook. Writing code is very simple and intuitive. Issues usually only arise when there are problems with installed extensions. But, that is more of an issue with JupyterHub than Jupyter Notebooks. The ability to use markdown helps a lot in terms of the amount of information we can present explaining what the code does. I think markdown has a bit of a learning curve. However, once the basics are understood any syntax issue is simply a google search away. Partially completed Jupyter Notebooks are amazing and are my preferred way of doing homework. It allows space for the instructor to explain the problem and give you any formulas that you might need. It also gives you space to

explain your thoughts surrounding the code as well as show mathematical work with markdown.

b. How does this compare against your preferred coding environment?

i. Detail your preferred programming environment, compare & contrast it with the Jupyter Notebook.

1. I normally use VS (Visual Studio) Code for almost everything because it is a local install on my computer and I like the layout of it. However, I frequently use Jupyter Notebooks on VS Code for my other classes. My data science class gives us our homework in Jupyter Notebooks and I love the format of it. I think that Jupyter Lab is good for new programmers because they do not have to deal with file management, software installs, etc. I used the online coding environments for my first couple of years at CU because it was simple and easy. The ability to log on to Jupyter Hub from any computer is very handy for people who have a desktop and laptop and like to do work on both. Without storing your files on Jupyter Hub you would have to set up cloud storage or heavily use GitHub to get files from one computer to another.

ii. What are the strengths and weaknesses for you?

1. Strengths: The biggest strength of Jupyter Hub is the online accessibility. As I previously mentioned it makes life easy for new coders because they don't have to worry about file management as much. You can also use Jupyter Hub on Chromebooks that would otherwise struggle to run/download coding environments locally. Jupyter Notebooks are awesome because they combine markdown with coding blocks. This allows instructors to create detailed explanations of what the preceding code block is doing. Completed Jupyter Notebooks are great to go look at for examples and references, and partially completed Jupyter Notebooks are excellent for homework assignments because you can write detailed instructions with markdown on the same document that the code will be written on. We no longer need a separate PDF that explains the assignment and the questions because we can put it all in markdown. Students can also use Jupyter Notebooks for more than just code. Any hand-written work can easily be transferred to markdown. Handwritten work is more readable in markdown which makes the graders like way easier. Students can also answer logic-based questions that don't require any code in markdown. Again, this is beneficial because it's all combined on one document.

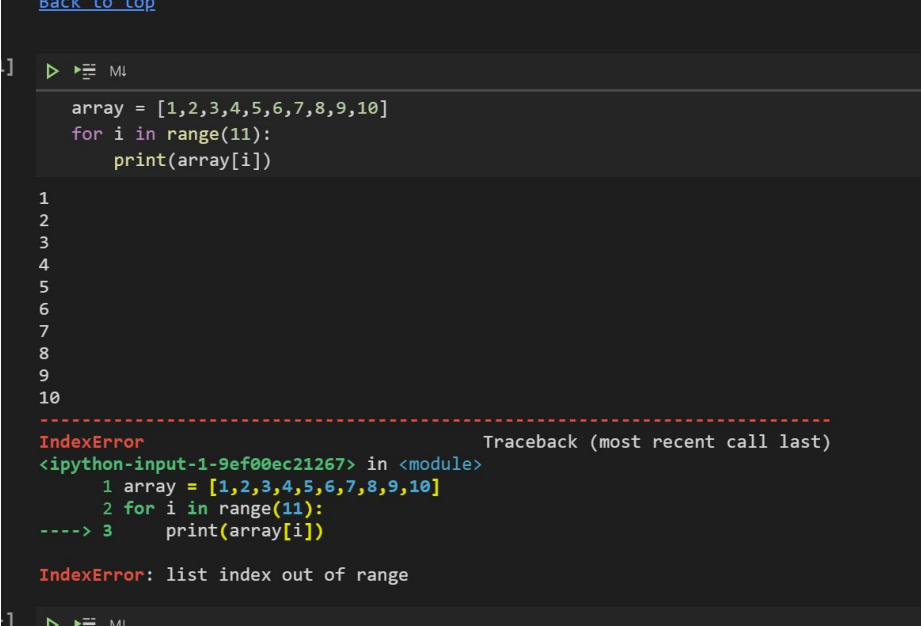
2. Weaknesses: Jupyter Lab has some speed issues, especially if a lot of people are using the server at once. There are a couple of homework assignments in other classes, the bomb lab for example, that can put a heavy strain on Jupyter Hub. Because of

this, I tend to just do stuff locally. VS Code also has more customization options than Jupyter Lab that I prefer using. Jupyter Lab can have some memory issues if you are working with big tables, especially if you have to use a “join” operation to combine them. The resources used in “join” operations forced me to work with R locally last semester, and I still had memory limitations with that. In terms of Jupyter Notebooks, I don’t think I would really opt for making one of these if I am only working with code. For bigger projects, I’d probably stick to working with individual files rather than a Jupyter Notebook. I think Jupyter Notebooks are better for study guides and homework assignments.

- iii. What might be the strengths and weaknesses from a freshmen’s perspective/inexperienced programmer?
  - 1. Strengths: You don’t need a powerful computer to use Jupyter Lab because we are on the school’s server. Some computers, Chromebooks for example, are not able to download a lot of IDE’s because of the OS on the Chromebook. For these students, Jupyter Hub might be the best coding environment available to them. Without these online coding environments, Chromebook users are almost forced to make a Linux partition on their laptops. Using Jupyter Hub also removes a lot of the headache of setting up IDEs and installing the necessary programming languages. You also do not need to worry about file pathing nearly as much because everything is stored in the folders on Jupyter Lab. Jupyter Notebooks are fantastic for inexperienced programmers because of the amount of information we can include in markdown. A good, informative Jupyter Notebook can provide students with almost all of the knowledge they need to complete an assignment. Markdown/LaTeX can easily be used in other classes. A base understanding of markdown would provide students with the ability to create beautiful study guides for other classes, especially Mathematics classes.
  - 2. Weaknesses: Online availability is one of the strengths of Jupyter Lab, but it is also one of its weaknesses. As I have previously mentioned, there can be a lot of strain on Jupyter Hub during certain times of the semester depending on what other CS classes are doing. The last thing we want is for a student to be unable to complete their assignments because they don’t have a coding environment available to them. Even if it is accessible, it can be very slow sometimes. This could be problematic when working with bigger data sets and could lead to timeout errors. In terms of Jupyter Notebooks, the biggest weakness is going to be learning the syntax. Not only do students have to learn basic R (or maybe Python?) programming, they also have to learn the markdown

syntax. I don't find markdown very difficult to use, but I have a lot of practice with creating math assignments through Latex and working with Jupyter Notebooks. This being said, markdown/latex are fantastic languages to know and can be widely used in other classes.

- c. What was your experience with debugging in Jupyter Notebook?
  - i. If you didn't experience any need for debugging, try to create a simple error (e.g. off-by-one error in a loop) and think about how you might debug it.
    - 1. Debugging in general is pretty good with Jupyter Notebooks because the cells are split up. You don't have to worry about breakpoints as much because most code blocks are rather small and isolated. On the other hand, the online Jupyter Labs doesn't have the most sophisticated debugging tools. I think that TA's/CA's/LA's will be needed to help the new programmers debug their code. It was very hard for me to debug anything as a beginner because it's hard to tell where your issue might be located if the error code isn't very specific.
    - 2. R in Jupyter Hub tells us less about an off-by-one error in a loop. To explain, look at these images.
      - a. You can see that this first notebook using a Jupyter Notebook in python on visual studio will give us a syntax error



The screenshot shows a Jupyter Notebook interface. At the top, there is a link labeled "Back to top". Below it, a code cell contains the following Python code:

```
array = [1,2,3,4,5,6,7,8,9,10]
for i in range(11):
    print(array[i])
```

The output of the cell shows the numbers 1 through 10, each on a new line. Below the output, a red dashed line separates it from a traceback. The traceback text is as follows:

```
-----
IndexError                                Traceback (most recent call last)
<ipython-input-1-9ef00ec21267> in <module>
      1 array = [1,2,3,4,5,6,7,8,9,10]
      2 for i in range(11):
----> 3     print(array[i])

IndexError: list index out of range
```

- b.
- c. Now take a look at the R Notebook on Jupyter Hub, it doesn't throw an error when the array index is smaller than the loop range

```

74]: array = c(1,2,3,4,5,6,7,8,9,10)
      for (i in 1:11) {
        print(array[i])
      }

[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
[1] NA

```

- d.
- e. In general, I think that Jupyter Hub has a lack of debugging tools, but the ability to split up the code into different blocks with Jupyter Notebooks definitely helps people isolate their errors faster.
- ii. The focus is on understanding how we might need to support students in debugging their R scripts.
  - 1. First-time coders will definitely struggle A LOT with syntax issues.
- 4. General Comments
  - a. What do you like/dislike about creating Jupyter Notebooks and working with GitHub? This can include past experiences (e.g. when you first learned to use R, git/Github, or Jupyter Notebook)
    - i. I think Jupyter Notebooks work best when they are given to students as study materials. The combination of or markdown + code makes it excellent for explaining what the code is doing. Without markdown, students are forced to read in-line comments or just decipher the code themselves. Jupyter notebooks are also awesome when they are given as a partial template for homework assignments. You can simply use the markdown to explain the questions and set up a general structure for the code that will be used. We can include the necessary libraries at the top as well. Writing a notebook from scratch would definitely more work for a student and I do not think it would be as beneficial as giving them completed notebooks as examples and partially completed notebooks as homework.