Hello mentors! This is Will Tirone, a past Python mentor. I’m writing this guide alongside the syllabus to help you guide conversation and demos in the class. Obviously, feel free to deviate as you see fit to discuss whatever students are curious about, but these are my thoughts as I was making the syllabus. When I mentored, it was exclusively over zoom so that affects my thoughts on how to approach class, so my ideas may not necessarily carry over easily to an in-person setting.

This is going to be pretty similar to Data 1 in terms of my notes, because it really is supposed to be an extension of those ideas. Especially in the later weeks, the big emphasis will be answering their questions and troubleshooting their projects.

I also developed an example project that you can follow along with [HERE](https://colab.research.google.com/drive/1JOOZCxAw07BD9kCTIQfBVuId_gGZstvh#scrollTo=452c6a08) and present each week of the class. I used topics from the corresponding week of the syllabus and worked through a data set with them. Spend about 30 minutes each week walking through each of the weeks in the project, and just use this as a guide for discussion. The students don’t have to do anything with this other than listen, this is simply meant as a guide so the mentors don’t have to spend an extra week writing code to show examples in class. I also (hopefully) have enough written instructions in there that if you don’t have time to make it through the whole thing, students should be able to read it on their own and understand the ideas. It’s also intended to show them topics from the syllabus while being an example of how they could meet requirements for their projects.

| **WEEK** | **NOTES** |
| --- | --- |
| 1 | Similar to Data 1, the big goal here is making sure they know how to run Python code and remember how to use git. Some of them may need a refresher so I included an optional resource this week. The goals for the course are to develop competency in SQL / Pandas / Tableau / plotting libraries primarily, with some other stuff thrown in there.  Go over the syllabus at a high level, answer any questions about the class. The first class is mostly very basic stuff about answering questions, telling them what to expect over the next few weeks, and introducing everyone.  On zoom, we would typically split into breakout rooms evenly with the mentors. We’ve done both random breakout rooms or assigning people to specific mentors and having those mentors checking in every week then talking to them in a breakout room. Honestly, either option can work depending on mentor preference but I lean towards just randomly assigning into breakout rooms and doing standup. Just do a quick 2-minutes per person “tell me about what you worked on, anything you had trouble with, and any project ideas”. That usually suffices, but if you have a more formal agile framework you want to use with them that’s fine too. |
| 2 | Start with standup / breakout rooms and ask them how they felt about the videos. At this point, people are probably pretty caught up but **in future classes people will start to fall behind on videos.** I have seen many very capable students just fall behind on the videos and drop out of classes (either due to other commitments or feeling like they can’t catch up) so try to gently emphasize to them the importance of staying on track. The most important goal of mentors is just to help keep them on track and keep at the material - you don’t have to be a code wizard, you just have to be encouraging to them. That’s so important, just keep encouraging them.  For SQL, in the past, some mentors have mentioned spinning up an AWS server that students could connect to to query. I’ve never done this, so I just included a basic online resource that they can use to practice queries in a browser window. If any of them want to install a db locally then by all means, but that’s always been a bit of a trickier thing to have them do with their project. However, if you want to set up a server for them and have more knowledge in that area than I do, then go ahead! |
| 3 | Some discussion about best practices this week, do your regular stand up and see if they have any questions about the best way to write a certain piece of code or something like that.  The optional guide to making your first contribution on GitHub was super helpful to me a few years ago, and code lou has had a goal of getting students more involved in contributing to larger projects. We don’t have firm requirements for this yet but this guide could be a good start.  Spend about 30 minutes on the knowledge check here:  [**Knowledge Check 1**](https://colab.research.google.com/drive/1UqcdOT_3B6vjZqafnU32l2d0O_Vc1G7w?usp=sharing) |
| 4 | At this point it will probably smooth out until week 9 or 10 when they’re really in the middle of their projects. Just proceed with standup like normal and try to help them brainstorm project ideas. For their project, I’ve noted that they can do a data analysis project or something more abstract that involves data (for example, building some kind of GUI that takes user input) so start helping them brainstorm general ideas for topics.  The seaborn videos I added we haven’t used before - this seemed like a pretty good introduction to a different plotting library, and seaborn (to me) is more straightforward to use than matplotlib. |
| 5 | I’m going to be honest, I’ve never used Tableau so I’m no help here. I’ve spent a lot of time in PowerBI though, and it seems fairly similar. Since it’s a pretty visual tool, most students have picked up using Tableau really quickly and the guides that Tableau themselves publish are pretty good.  If they decide to use Tableau to meet one of their final project requirements, they just need to have a link to their public dashboard in their README. |
| 6 | Students may start to drop the class here as they fall behind, so if any of them tell you they can’t finish the class, have them reach out to Code Louisville staff to see if they need extra assistance to finish the class. Historically, it has been due to work conflicts / child care / not enough time, so just see if there’s anything you can do to help them out.  Help them with project ideas - tell them if they don’t have a dataset or project idea they want to work on at this point that you’re going to pick one for them next week. If anyone is really struggling to think of something, jump in a break out room with them and ask about things that they find interesting, programming related or not. This usually helps to get them thinking about it enough to commit to something.  Spend 30 minutes going over the knowledge check here:  [Knowledge Check 2](https://colab.research.google.com/drive/1CV6llGPjn05pqwO-PSHJdO5O8E1OrXm6?usp=sharing) |
| 7 | Have them commit to a project idea. I also included a guide here about virtual environments - they will be an optional project requirement, but students are usually generally curious about how they work. Feel free to walk through setting one up if they’re curious and explain their purpose.  Go through standup and if any of them haven’t committed to a project idea, try to get them to commit to something before class ends so they can start working on it. |
| 8 | I included a few more useful videos about PySpark and Big-O, but emphasize to them that we don’t expect them to use either of those concepts on their project. Just in there for them to see slightly more complicated ideas.  At this point, it will mostly be helping them troubleshoot their projects and come up with ideas for what direction they want to go with their data analysis. Encourage them to come to class with questions.  It can also be *hugely* helpful to other students to have one or two students that have started working on their project to show their code to the class, even if it doesn’t work. It’s actually maybe even better if it doesn’t work so the mentor can help them troubleshoot and explain to them how you would go about fixing their issue. It’s always very helpful for students as well to see other people at a similar point in their projects so they don’t feel like they’ve fallen behind. |
| 9 | Emphasize how you will download and run their projects. Students have gotten confused about how someone else would download and run a jupyter notebook (fair) so just explain that you’ll very literally just clone their repo and run it from the start. Tell them to restart the kernel and run all cells before uploading their notebook to make sure everything runs in order.  If they have an API key, students have used config files in the past and just DM’d their mentor the API key on slack to replace in the file. Anything works, as long as they aren’t uploading the API key to GitHub  Spend 30 minutes on the knowledge check:  [Knowledge Check 3](https://colab.research.google.com/drive/1oIOejKr-qHC4L3EGb1ycQaAsjiChbft1#scrollTo=kYd3iy8gKYwV) |
| 10 | I cannot emphasize this enough: have the students send you their project or put it in the main slack channel so other people can try to run it.  One of the more easily fixed problems is a student not realizing their code doesn’t run on someone else’s machine until the night before the project is due (because of a path error in a file, for example, or because they forgot to tell you to install a certain package in the README.) This is essentially the last 3 weeks of class. Some people will have finished their project by week 6 and some people will be scrambling to finish at the end, which is fine! Just encourage them to have someone else as well as a mentor run their code on Windows, OS, and Linux (if possible.) Usually Windows and OS will be sufficient but it’s not something that might be obvious to them. |
| 11 | In the past, if students don’t have many questions about their projects at this point, sometimes we’ll go over resumes and job hunting tips. You, the mentors, have valuable perspective on that for them even if they’ve already talked to code lou staff, so they usually benefit a lot from interview tips / resume help / things of that nature.  Otherwise, it’s more of running their projects on your machine to see if you can get them to run. Encourage them to post their projects in the class slack channel so other students can see what they’re working on and also try to run their projects. Remind them to have their tech sessions / tech interview / project turned in at this point.  It’s usually useful as well to go through a project and grade it during class to show them how you will do that once you’re actually grading them. |
| 12 | Last class, woohoo! Remind them to have their projects turned in and help with any last minute issues.  90% of students will be done at this point, and 10% will tell you they don’t actually know how git or Python work and you’ll have to scramble to help them finish, hence my emphasis on running their projects in week 10 (or earlier). Regardless, just do your best to help them however you can and try to get them across the finish line.  Congratulations on finishing Data 2! |