

Inf2 – Foundations of Data Science 2024

Task: Semester 1, Week 1 Lab



17th September 2024

Over the course of FDS there will be 13 weekly lab sessions, in the form of Jupyter notebooks that introduce data science programming concepts and contain exercises to test your understanding and develop your problem-solving skills.

You should expect to spend up to 2 hours on each notebook. You may find a notebook takes longer, especially towards the start of semester 1, when you're getting used to Pandas, so we encourage you to ask and answer each other's questions about labs on Piazza at any time. In order to help guide you throughout the labs, you will be able to ask questions to Lab Demonstrators, as well as be able to reveal Hints & Solutions provided within the Lab itself. In this handout we give reasons to work on the labs in pairs, and explain how to run the labs.

Rationale for pair programming

There's considerable evidence that learning is most effective when it's **active** and **collaborative**¹ In the notebooks we provide you with exercises to complete, which should help the learning to be active.

To make the learning collaborative, we suggest you use **pair programming**, meaning:

- You take turns at being **driver** and **navigator**.
- The **driver** is at the keyboard; the **navigator** watches the screen and makes suggestions to the driver.
- Every 10 minutes you swap places.

Pair programming is used in industry. Evidence suggests that it improves code quality over solo programming and reduces time spent on programming compared to solo programming (Hannay et al., 2009). It is also more enjoyable (Williams et al., 2000). There's also evidence to suggest that pair programming is beneficial to student learning, because:

- it helps you to solve problems when you get stuck
- it leads to greater confidence
- talking about what you're doing can lead to understanding (Williams and Upchurch, 2001).

¹This makes sense neuroscientifically: Learning happens due to changes in the structure of our brains, particularly at synapses, the connections between nerve cells. The changes depend on activity, so the more our brains are active, the more learning there's likely to be.

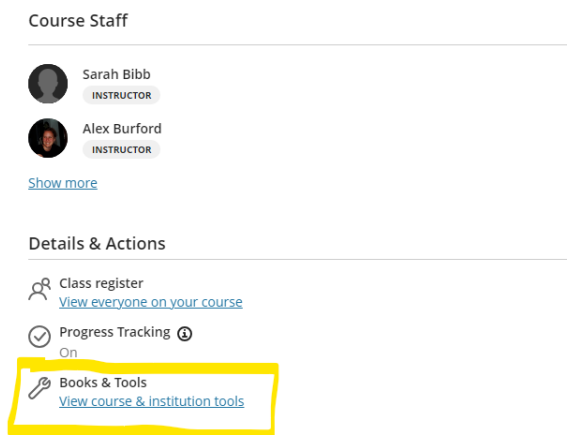
Of course, we shouldn't accept all of this evidence uncritically: there are variations between the findings of various studies and potential publication bias (Hannay et al., 2009). The benefit of pair programming may depend on your personality and whom you pair up with.

If pair programming really doesn't work for you, for whatever reason, no worries. However, if you're comfortable with the concept, but it doesn't work out with one coding partner, we do encourage you to try to find another coding partner the next week.

Running the labs

To access the labs we recommend you use **Noteable**, the University of Edinburgh's Jupyter notebook service. If pair-programming isn't possible, then you can go solo by following steps 1–11, and then ignoring all references to pair programming in the notebook.

1. Decide who's going to start off as the **driver**, and who's going to start as the **navigator**.
2. The **driver** opens up a Firefox or Google Chrome browser window (not Microsoft Edge).
3. Make sure that you can both see the same screen – we suggest you use the lab machines and increase the magnification of the browser using `<Ctrl> + +`.
4. The **driver** goes to the FDS Learn page and navigates to **Books & Tools**, which is located in the sidebar and looks like this:



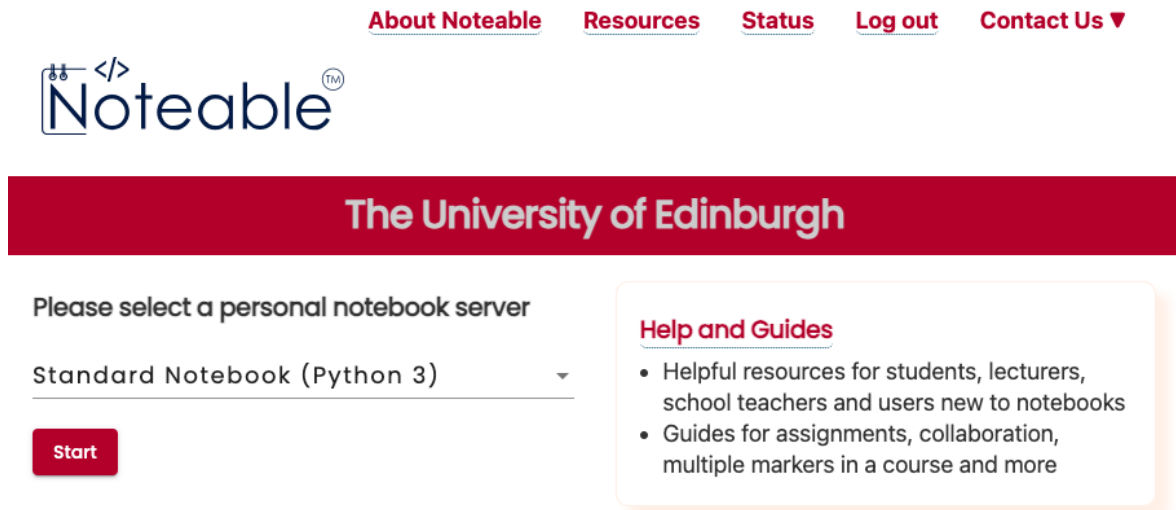
And then click on the **Noteable** link, which looks like this:



Noteable will automatically open in a new tab. If you try to manually open Noteable in a new tab, Noteable will not load.

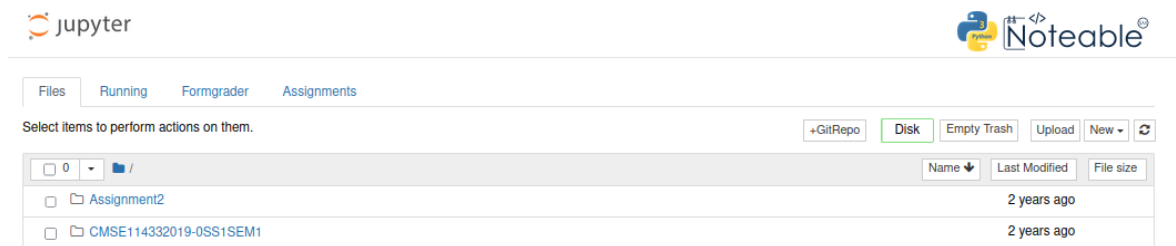
5. The next screen may say "Loading...". There may also be a message saying "Your browser may have blocked the opening of the new window; try clicking the button below to open 'Noteable.'" If so, click on the button.

- Once Noteable has loaded, you should see text saying “Please select a personal notebook server”. Ensure **Standard Notebook (Python 3)** is selected and click **Start**.

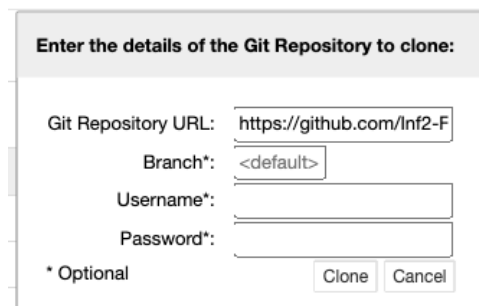


You may get a message asking whether you want to reconnect or shut down an existing notebook server. You normally do not have to click anything, but if the dashboard doesn't load in a couple of minutes, click “shut down” and start again. You will get this message if you already have Noteable open in a separate tab, in which case you can click “reconnect” to work in multiple tabs.

- The driver should now see the “notebook dashboard”, which looks like this:



- Click on the button labelled **+GitRepo**
- Type the following Github link into the box titled 'Git Repository URL':
<https://github.com/Inf2-FDS/FDS-S1-01-introduction>
 Leave all the other boxes blank and click **Clone**.



Future weeks

For weeks after week 1, to find the link, you would go to <https://opencourse.inf.ed.ac.uk/inf2-fds/course-materials>, then scroll down to and click on the appropriate week e.g **S1 Week 1 - Introduction and Data** and then find the lab e.g **INF2-FDS: S1 Week 1 Lab**, which will contain the link to the lab.

10. After a few seconds the **driver** should be redirected to a folder called `FDS-S1-01-introduction` in the **Files** tab. The folder should contain two items: `datasets` and `FDS-S1-01-introduction.ipynb`.
11. The file `FDS-S1-01-introduction.ipynb` is a Jupyter notebook containing Python code. (The `ipynb` extension stands for “Interactive Python Notebook”.) Click on `FDS-S1-01-introduction.ipynb`.
12. The driver and navigator should now work through the notebook up to the point where the notebook suggests that you swap. At this point the **navigator** takes over the keyboard from the **driver**. After that, swap over every 10 minutes or so – the first notebook has suggestions for when to swap.

If you would like to pair-program on two separate devices using Noteable, see <https://noteable.edina.ac.uk/documentation/collab-editing/> for how to do this. However, even if you do this, only one of you will be able to run the code.

File sharing and logging out after pair programming with Noteable

When you have worked through the notebook, you will probably want to share the actual file (not just access to each other’s Noteable, which will eventually time out). A simple (but not elegant) method is to download the file and email it to each other.

There are two ways of downloading a file:

- From an open notebook: **File**→ **Download as**→ **Notebook**
- From the notebook dashboard: check the box next to the file and click the **Download** button above the list of files. The **Download** button is only visible when notebooks are not running (i.e. the icon next to them is grey) so you might have to select them and click the **Shutdown** button before you see the **Download** button.

To logout of Noteable, you should:

1. Go back to the notebook dashboard
2. Click on the **Noteable** icon at the top
3. On the next screen, click on **Shutdown**.

Other ways of running Jupyter notebooks and doing pair programming

We are not offering active support for running Jupyter notebooks on DICE machines or your own machine, but if you would like to do so, take a look at the **Labs** page on the OpenCourse pages for some hints about how to do this, or the README file on GitHub (<https://github.com/Inf2-FDS/FDS-S1-01-introduction#readme>).

To pair-program, you could fork the Jupyter notebook's GitHub repository and push and pull each other's work. Unfortunately, due to the underlying JSON data format of notebooks, the diffs between versions of Jupyter notebooks are difficult-to-read and larger than the changes appear in the notebook viewer. Therefore, the benefit of being able to view diffs in GitHub is less than it would be for plain code.

References

- Hannay, J. E., Dybå, T. et al. (2009). 'The effectiveness of pair programming: A meta-analysis'. *Information and Software Technology* 51:1110–1122. URL <https://dx.doi.org/10.1016/j.infsof.2009.02.001>
- Williams, L., Kessler, R. R. et al. (2000). 'Strengthening the case for pair programming'. *IEEE Software* 17:19–25. URL <https://dx.doi.org/10.1109/52.854064>
- Williams, L. and Upchurch, R. L. (2001). 'In support of student pair programming'. *ACM SIGCSE Bulletin* 33:327–331. URL <https://dx.doi.org/10.1145/366413.364614>