13th Advanced Scientific Programming in Python

a Summer School by the ASPP faculty and
the Institute of Neurodegenerative Diseases, University of Bordeaux

23–28 August, 2021. Bordeaux, France

Evaluation Survey Results

Method

The survey has been administered with a web interface created with the LimeSurvey software available at:
http://www.limesurvey.org
All answers have been submitted by 11 October, 2021.
No answer was mandatory.
The free-text answers have not been edited and are presented in their original form, including typos.

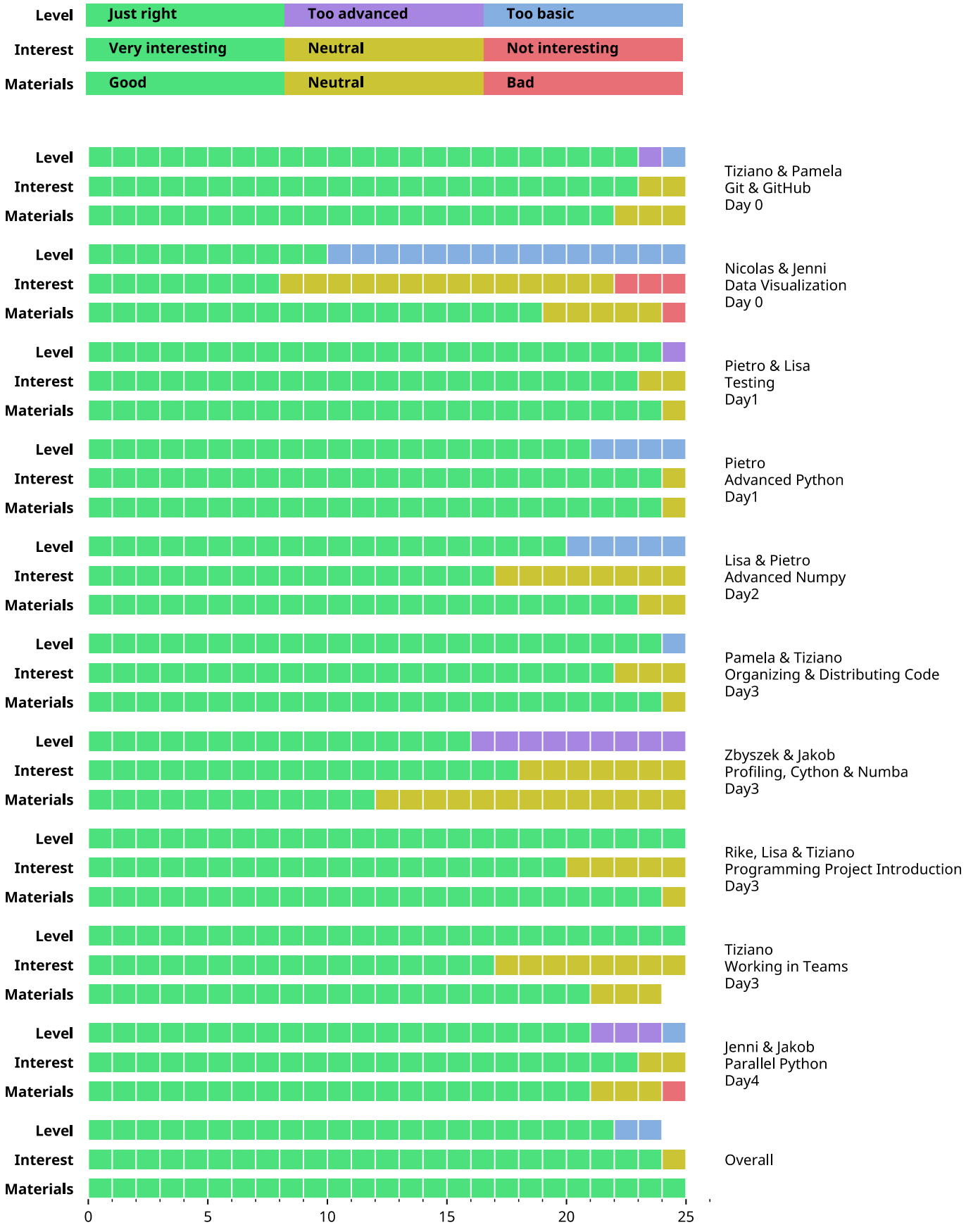
Attendants and Applicants Statistics

	Attendants		Applicants	
	28	17%	165	
Different nationalities	16		39	
Countries of affiliation	7		27	
Gender: other	0	0%	1	1%
Gender: female	13	46%	73	44%
Gender: male	15	54%	91	55%
Already applied	9	32%	17	10%
Bachelor Student	0	0%	2	1%
Master Student	4	14%	31	19%
PhD Students	18	64%	105	64%
Post-Docs	4	14%	12	7%
Professor	1	4%	2	1%
Technician	0	0%	1	1%
Employee	1	4%	9	6%
Others	0	0%	3	2%
Completed surveys	26	93%		

More stats about attendants are available at: https://aspp.school/2021-bordeaux/students.html

Lectures & Exercises

Q: *Grade the* ***level*** *of the lectures*
Q: *Grade how* ***interesting*** *were the lectures*
Q: *Grade the quality of the presentation style and/or of the teaching* ***material,*** *e.g. the clarity of the slides/code, the exercises and the solutions, etc.*



Q: Are some of the topics presented in the lectures not relevant for a programming scientist?

1. The cython and parallel programming weren't really relevant because most of the modern python frameworks take care of that under the hood and also because it is such a complicated topic that you can't really practically learn in such short time.
2. I genuinely believed that you did amazing work and that the teaching is very suitable for scientists ;)
3. All topics are relevant to the programming scientist but I feel that from all of them the data visualization lecture is the least relevant for the course.

   As a programming scientist I already had too make tons of figures/graphs over the course of my studies and in my work. More importantly I already received a lot of instructions and feedback on these from my University professors, PhD promotor, paper reviewers, ...

   This is not too say I did not learn anything new from the ASPP course or that there is no more room for improvement but I am already knowledgeable on the existence on this topic and where to self learn/find help. For the other lectures such as the cython one ASPP was much more helpful as I knew virtually nothing about this topic at the start of ASPP.
4. I would consider stripping down the theoretical part of the data viz session and maybe replace it by something more hands on; most people know by now that jet color map sucks but facetting subplots still makes me head aches when thinking of it (because I never got properly introduced to its logic)
5. I would have loved to learn more on proper documentation of the code and linting.

Q: Are there further topics relevant to the programming scientist that could have been presented, given that the total time is limited. Please also mention which topics should be replaced by the new ones.

1. making GUIs eg using tkinter could be a nice addition - this can be useful for visualising graphs from different experiments but with the same protocols (eg. having a 'next' button to scroll through plots from experiments of the same kind, having the option to select particular experiments etc). Therefore, it could be combined into the data visualisation topic (I would replace the numpy topic since I think most people are quite familiar already with numpy)
2. I'm thinking about pandas as a relevant topic for programming scientists. But actually, I wouldn't cut any of this years' topics for pandas. Although it is relevant to programming scientists, it isn't as complex and there is a large amount of introductory material on this particular package. Pandas does not require as elaborate a course as you have offered.
3. I would have appreciated if there were some sources about using online servers, cuda and docker containers
4. I would replace data visualization with a topic related to working on a cloud or something like this. I think maplotlib and seaborn already have many beautiful tutorials and it seemed that all of us already had some data vis workshops before so I think this is the course that was least new to us ;) I also think a course about working with bash or sth like this could be cool.
5. ----
6. I personally would restructure the lectures about Cython and parallel processing. Both of them provided some solutions for running our code faster, but were focused only on the solutions which in practice should be our last resort. I think it would be beneficial to talk instead a little bit about more basic stuff, like vectorization, other good practices and profiling. The tool for profiling that Jakob showed us seems very useful, but we didn't have time to really use it (for example, to optimize some basic function with it). Instead, we could spend less time on talking about Cython and running parallel computations on a cluster. Theses topics were so complicated that there was no time to explain them really well anyway, so I guess a Cython demo (instead of multiple Cython exercices) should be enough.
7. pandas & seaborn (maybe instead of matplotlib)?
8. - PROs and CONs of various IDEs, text editors, console interfaces in different contexts
9. I think it would be beneficial to have a module on creating interactive dashboards, hosting them and sharing them with others (e.g. collaborators)

Q: Do you think that pair-programming during the exercises was useful?

Yes, I have learned from my partner / I have helped my partner	92% (23)
No, it was a waste of time for both me and my partner	0% (0)
Neutral. It was OK, but I could have worked by myself as well.	8% (2)
Other	0% (0)

Q: What do you think of the balance between lectures and exercises? When answering, please keep in mind that the overall time is limited ;-)

Lectures were too long, there should be more time for exercises	8% (2)
Lectures were too short, there should be more time for lectures	0% (0)
The time dedicated to lectures and exercises was well balanced	84% (21)
Other	8% (2)

Other:

1. Some of the lectures were good. Some had a really short time for excersices
2. Sometimes the time given for exercises were too long - maybe exercises can be more complex and cover more material

Q: Any further comments about the lectures and exercises?

1. Having two lecturers per lecture/topic was a great way to keep attention focused and not make the lectures seem too long, it also made the lectures have a more informal feel without them becoming too chaotic.
2. Maybe I'm just slow but I wish I had had more time for the exercises. This wasn't always the case though! I sometimes simply needed more time to reflect on stuff. I am aware that I should use the next time (after the summer school) for this.
3. About the Git lecture, I would add a summary slide or two with the different scenarios so that we make sure that everybody understood properly how it works :)
4. I didn't know that it's possible to teach Python so clearly and efficiently. Good job!
5. All lectures were immensely useful. The enthusiasm displayed by ALL instructors was super contagious. The atmosphere in the lectures felt very safe and open-minded to me, so that I was never afraid to ask a question or ask for help. It is an amazing and special school that you guys have created there. Thank you so much!
6. The lectures and exercises were great!! Concerning the balance between the length of lectures and exercises, there were (only) a couple of instances where we had (in my opinion) a bit too much time for an exercise, e.g. we were given 10 minutes and all we needed to do was to execute 3 Jupyter notebook fields. But other than that it was very well balanced, and I really enjoyed the switching between lectures and exercises, it kept my attention high during the whole lectures (which is an achievement in itself!)
7. I sometimes felt that there was not enough time spent on doing exercises.

   In particular for the testing and cython lectures I would have liked more time to exercise and for more exercises to be discussed.

   However the accompanying lectures were not too long and I would not shorten them.

   My choice would be to cut the data visualization lecture in favor of more time spent on cython, testing and git.
8. The concept of pair programming is brilliant! I think it would be great to integrate it more into the standard practices of scientific programming.
9. Best course ever!
10. Great job to all lecturers. I was able to take something from every section! Most of it is super relevant to my work and I have already started implementing some practices.

    A few sections suffered from being (too) short, such as Data Visualisation which could have been explored more deeply. I do appreciate the teaching materials being hosted on GitHub though as this might allow me to study those sections in more detail now after the summer school.
11. There was a very good mix of theory and practice. Due to the exercises in between the lectures were not tiring and easier to follow.
12. It was a great experience!
13. It is difficult to make a course for a large audience of the programming scientist. You manage to make interesting lectures for scientists with different backgrounds and different programming skills!
14. Amazing!!!
15. It was a blast. There's not much to add to that.

    One consideration: how about offering "parallel tracks" for some of the more complex topics? So the participants can/have to decide for some topics but then have the chance to dig deeper into those rather than scratching the surface of all of them. E.g., for documentation it'd have been nice to work through one small (undocumented) dummy project from beginning to end, same for optimizing a script via Cython. That needs time which could be won by deciding for one of them (materials of the other track can then be worked through by participants themselves afterwards). Just an idea - you probably thought about it and have a reason wh things are as they are.

    Anyway, keep up the good spirit and thanks a lot for everything!

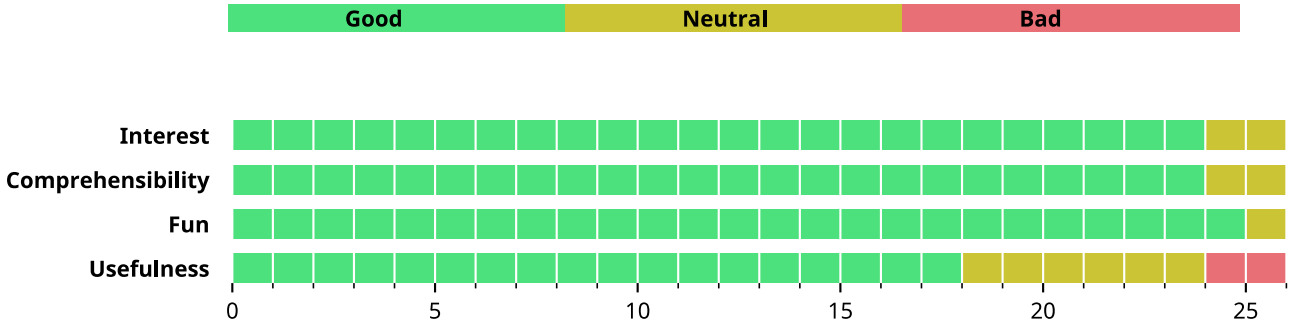
16. It was just exceptionally well organized. It was definitely one of the best courses I had during my years of study, if not even the best, and never learned so much in a week. And this taking into account that its for quite a spectrum of different levels!

    Only the last lectures in the evening were too hard for me too concentrate, and I might be a possibility to rearrange the schedule a bit (and e.g. not have a need for 2h lunch break)

Programming Project

Q: Evaluate the programming project.

Interest: How interesting was the programming project?
Comprehensibility: How clear and comprehensible was the code and the available documentation? Was it easy to work on the programming project
Fun: Was it fun to work on the programming project?
Usefulness: Was it useful to work on the programming project? Do you think you may re-use what you learned?



Q: Do you think the team-programming experience is relevant to your work as a programming scientist?

Yes: 88% (23)
No: 12% (3)

Q: Do you think that the project should be about a real-world scientific problem instead of a video game?

Yes: 8% (2)
No: 92% (24)

Q: Any further comments about the programming project?

1. I found the programming project extremely useful for learning how to (and how not to!) organise a code project from scratch, especially how to work best with other people on the same project and integrate each others code.
2. I actually had problems adjusting to the situation that someone sat beside me while I code. I tend to fully engage in the code. This was the case in this particular programming project. When me and my partner switched positions I often had the impression that I disrupted my partners train of thoughts. We each identified the typos made by the respective other though. But I believe that these wouldn't occur as often when coding alone. It was an interesting experience though! And I'm certain that having someone sitting beside me challenging my ideas will help at a later point of my career when I'm more confident in the code I write and I don't have to reflect on it as deeply. The project being about a video game was the right choice. It increased the fun immensly! Wouldn't change that.
3. There wasn’t a lot of room to use the advanced topics we learned during school. We could only use basic for loop and ifs. I would have preferred a project that involved decorators and context managers and also maybe defining classes. And of course some visualization.
4. The staff was so nice !! Thanks <3!!
5. I was a little afraid of the "competitive" factor of the project, so I really loved the fact that Tiziano stressed the importance of learning, good practices and group cooperation before we dove in. I also appreciate the fact that it was a non-scientific project (more fun and no coincidental experts). Since I never develop code in collaboration with others, it was exciting for me to see what this could be like as well. Loved it!
6. The programing project was super good and fun, but because of the limited time we actually didn't implement most of the stuff we learned. We these I mean, documenting the code, testing, making it readable and reusable. That also show how us under pressure bypass all this topics and go full into the pure coding.

   For the other side, we learned indeed a lot of git and teamwork.
7. The tournament was so incredibly exciting!!1
8. I found the programming project super helpful in learning how to make use of git.

   However I felt that the other lectures were maybe less well represented.

   In particular I would have liked the project to include speeding up the code with for instance cython and the testing aspect.

   Try to include an incentive in game to write well optimized code.

Try to force the groups to write tests by forcing the bots through a series of particular situations/challenges prone to introduce bugs (does a bot crash when teleported from his homezone to enemy zone, does a bot crash when completely surrounded by enemies, ...). This could be a fun addition to the tournament.

I would definitly keep the project in game format, the competition and fun aspect of the current pelita project is an enormously effective motivator.

9. PELITA!!!
10. I think the programming project can contain a challenge that is easier to solve and put much less emphasis on actually figuring out the best way to solve the task and focus more on implementation (using the learned material). The challenge of programming a robot requires one to focus a lot on the strategies that the robot could take and since these strategies are almost unlimited, one can easily get "distracted" while figuring out or discussing good strategies. If the task allows fewer (or more straightforward) possibilities, then one can focus more on what one has learned and apply the concepts. It is indeed really fun to think of what your robot and the enemy robot can do and figure out/try solutions but it sometimes becomes "too exciting" that one can lose the real purpose of the project work.
11. Though I mostly work alone as a programming scientist, the team-programming experience was extremely useful and rewarding!
12. It was very interesting to test what we learned in practice
13. Be bop ba bodda bope

    Bop ba bodda bope

    Ski-bi dibby dib yo da dub dub

    Yo da dub dub

    yeah, I'm the Scatman

    Where's the Scatman?
14. The project was very exciting and one can learn a lot through play. I'm even glad we didn't have a real-world scientific problem, we have enough of those in real life.
15. it was super good, and helpful to learn!

    One minor suggestion would be to make sure there is at least one "git-expert" in each group on the day of starting the programming project. In our group this would have increased the learning experience even more I believe, if there would have been one already having experience using git in a project with multiple people.

The School in General

Q: How do you overall evaluate the school?

Good: 100% (26)
Neutral: 0% (0)
Bad: 0% (0)

Q: How do you evaluate the general level of the school? Was it too advanced/too basic with respect to your expectations*?*

Too advanced: 0% (0)
Just Right: 88% (23)
Too basic: 12% (3)

Q: How do you evaluate the general level of the school? Was it too advanced/too basic with respect to what was advertised in the announcement*?*

Too advanced: 4% (1)
Just Right: 85% (22)
Too basic: 12% (3)

Q: Did you learn more from attending the school than you would have learned from reading books and online tutorials alone?

Yes: 100% (26)
No: 0% (0)

Q: How do you evaluate social interactions and social activities at the school?

Good: 100% (26)
Neutral: 0% (0)
Bad: 0% (0)

Q: Would you recommend this course to other students and colleagues?

Yes: 100% (26)
No: 0% (0)

Q: How did you hear about the school?

Google Search: 4
Professor/Tutor/Supervisor: 3
Colleague/Friend: 14
Mailing list: 5
Other: 0

Q: Any further comments or suggestions?

1. I found the course an overwhelmingly positive experience and took a lot away from it.
2. Besides making wonderful new friendships, I think (hope) my Python has really leveled up in a sustainable manner. The content in my opinion was perfect for scientists who are looking to apply these methods to their work - all while maintaining an open, inclusive and friendly atmosphere at each moment. A warm thank you to all organizers and instructors!
3. If I may evaluate your evaluation form (so meta), I would prefer it if it had 5 options instead of 3 for these good-neutral-bad questions. Sometimes, things are in between “too basic” and “just right”! Also, the question about social interactions is missing "fucking awesome" as one of the options :)

   And sometimes the order of these options changes, which makes it difficult for me to give you all the best scores without reading the questions :P
4. Thanks a lot!
5. It was just amazing! Great lecturer, great lectures, great group.

   And definitely continue doing the pair lecturing. And advise it to everyone you know doing lectures ;)!