Δήμος Σιθωνίας

G-Node German Neuroinformatics Node www.g-node.org

Advanced Scientific Programming in Python

a Summer School by the G-Node and the Municipality of Sithonia

August 28—September 2, 2017. Nikiti, Sithonia, Halkidiki, Greece

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 October 2, 2017.
No answer was mandatory.
The free-text answers have not been edited and are presented in their original form, including typos.

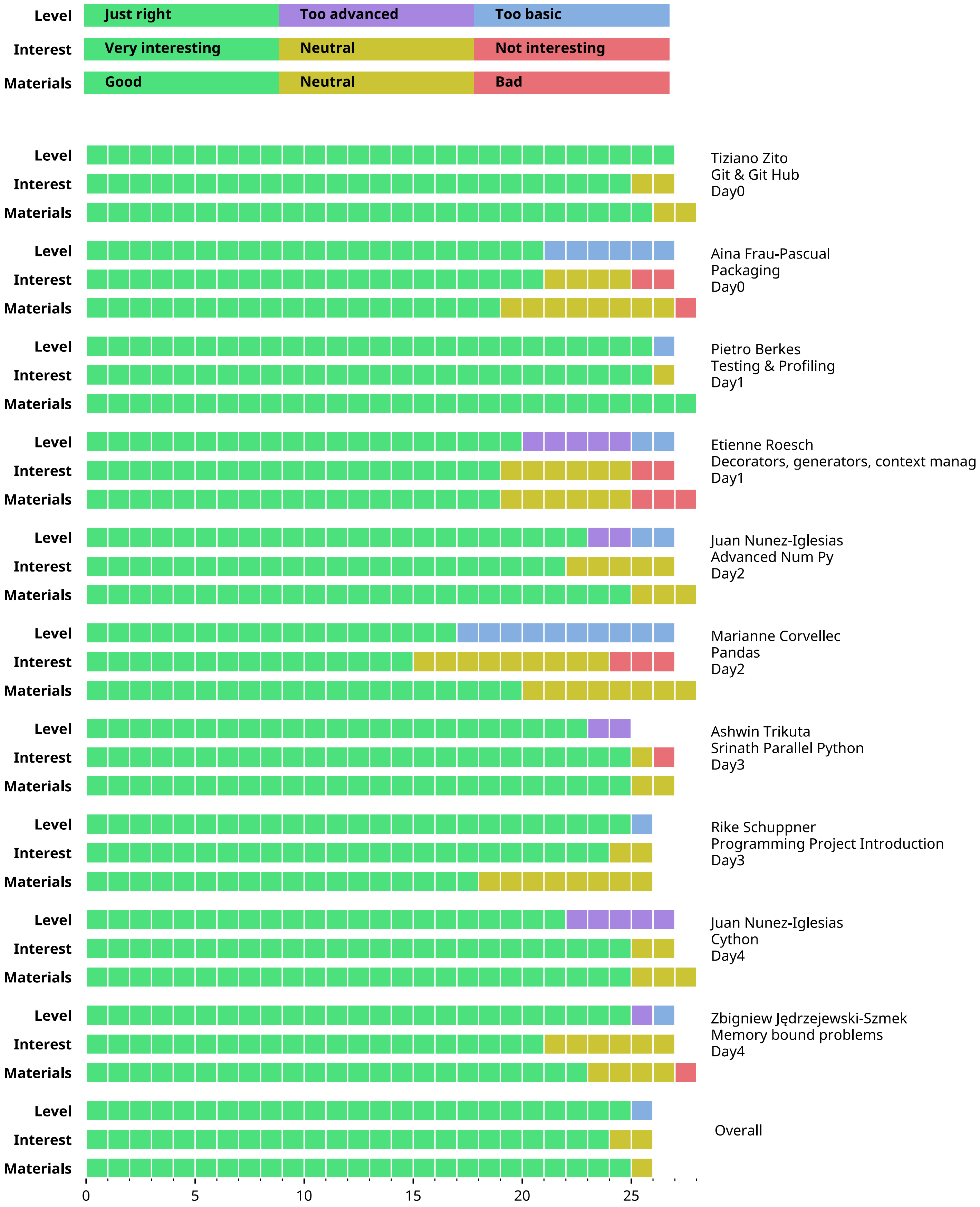
Attendants and Applicants Statistics

	Attendants		Applicants	
	30	21%	141	
Different nationalities	15		37	
Countries of affiliation	12		29	
Female	17	57%	61	43%
Male	13	43%	80	57%
Already applied	23	77%	78	55%
Bachelor Student	1	3%	5	4%
Master Student	2	7%	13	9%
PhD Students	19	63%	82	58%
Post-Docs	6	20%	23	16%
Professor	0	0%	2	1%
Technician	0	0%	1	1%
Employee	0	0%	6	4%
Others	2	7%	9	6%
Completed surveys	28	93%		

More stats about attendants are available at: https://python.g-node.org/python-summerschool-2017/students

Lectures & Exercises

Q: Grade the level of the lectures
Q: Grade how interesting were the lectures
Q: Grade the quality of the teaching material provided by the lecturer, e.g. the clarity of the slides, references given, exercises etc.



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

1. The Cython and memory-bound problems lectures weren't directly relevant to my current needs, but I'm glad to know about these things in case the need arises in the future.
2. the group challenge
3. I think all the topics are highly relevant.
4. You should cover provenance. Maybe have one block together with packaging and testing.

Q: Are there further topics relevant to the programming scientist that could have been presented, given that the total time is limited

1. Data visualisation
2. The packaging lecture was too basic and did not cover dependencies and hosting on private PyPI servers.
3. perhaps a develpment idl such as pycharm could be covered
4. Data visualization
5. I think the programme is complete as it. It offers a very well-rounded mix of relevant topics.
6. A bit less numpy/pandas and maybe some ML toolkits instead?
7. Debugging wasn't covered due to time limitations.
8. Visualization (which I know you meant to present), on using GitHub's issue tracker as a "to-do" list
9. Maybe tips on general structuring of programming projects (e.g. how to modularize in order to reuse code etc.).
10. Topics were well-chosen, given the diverse audience and time limits.
11. How to structure your project? Python classes, whether it's necessary, what can be added into classes and how to construct a class.
12. A quick survey of different scientific programming workflows could be interesting. For example, using a Jupyter notebook for interactive work and having a python package installed using pip install -e with IPython's ability to autoreload modules automatically. I think it would be interesting to see if there are any very practical tricks to be learned from other people in the class.
13. Visualisation would have been interesting (I realise that it had to be dropped at the last minute, just reinforcing it :))
14. Data visualisation (which was originally scheduled but cancelled due to unforeseen circumstances)

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

Yes, I have learned from my partner / I have helped my partner	86% (24)
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.	11% (3)
Other	4% (1)

Other: All of the above, depending on the exercise and the partner.

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	11% (3)
Lectures were too short, there should be more time for lectures	0%
The time dedicated to lectures and exercises was well balanced	89% (25)
Other	0% (0)

Other: .

Q: Any further comments about the lectures and exercises?

1. Overall very enjoyable and really useful having a mix of both lectures and exercises. It would have been good if the materials were less scattered (i.e. on the central ASPP Github repo) so we could find them easily to follow along during lectures and look-up afterwards.
2. This was the first course that I followed where I was never bored. Lectures and exercises alternated at an agreeable pace, the topics were very interesting and there was always someone to help. Even though the subject of some lectures was not on something I am planning on using, something did pop out that helps me in one way or another. The only thing I can think of improving is the proficiency in teaching of some tutors. While some tutors have a natural talent for teaching and structuring their classes, this proved to be more difficult for others. However, I do believe that this can be fixed by following a course on teaching :-) Apart from that the summer school was way better than I expected.
3. The level of the lectures and exercises was great, but often things moved a bit too fast and I felt a step (or several!) behind. It might have helped to have all the slides/course materials more easily available. Often during the exercises I felt like I needed to refer back to something the instructor had shown a few minutes earlier. But thank you to all of the instructors and tutors for helpful and patient responses to questions!
4. I do not think that direct use of MPI/multiprocessing is useful given the audience. Higher level tools like joblob and dask provide abstractions and are much easier to use. The lecture should cover these two instead of just mentioning them briefly. The lecture about decorators, generators and context managers was imprecise and unmotivated. Maybe one can showcase basic ideas of functional programming instead. Pair programming was useful but working with non native keyboard layouts very painful. This slowed down exercises.
5. Fantastic job. Will highly recommend to my fellow grad students.
6. The quality of lectures and exercises was brilliant, the lecturers have put a lot of effort into preparation of the materials and the tutors were helpful during the exercises. I found it most effective learning using jupyter notebook where a lot of the code has already been written but every now and then we stopped to solve a problem ourselves. I think that way we get exposure to a lot of concepts quickly and learn by applying them to simple problems. I can see the benefits of paired programming and it was interesting seeing how other people code. However, I think I would find it useful if for some of the lectures I had my own laptop and worked independently. The game project was a lot of fun and was really well organised!
7. It was a great course, and I am thinking of adopting this pair-programming routine to the research in my own project (for instance, by assigning tasks to pairs of students). The course is fantastic, I hope there will be next edition next year!
8. The lectures and exercises were great in general. I found the exercises very helpful in understanding the material. In particular, I really enjoyed the exercises for Advanced Numpy and Parallel Python. Since numpy is a commonly used scientific package, I enjoyed diving a bit deeper into its internals rather than rehashing basics. The parallel python exercise was a great real-world example of putting the lecture into action. The balance of lectures and exercises was also good for keeping attention high.
9. Some lecturers used ipython notebooks, this helped following the lectures and exercises. It could be useful in all lectures.
10. I think the IPython notebooks are good way of teaching especially if they are available to the students already during the lecture, because it is a lot easier to follow.
11. A couple of times the instructions for an exercise and the information relevant to complete the exercise were on different slides. Therefore, it was impossible to see the instructions and necessary information at the same time! This was inconvenient. Generally though. The topics were very interesting. They were pitched at the appropriate level. I think the focus on exercises was strong. I really enjoyed that. I really really enjoyed working in pairs. I think the course would not be nearly as engaging if we were working individually. I would have liked the lecture on decorators, generators etc to go a bit slower. I think the level was good but if more time were allocated to this topic we could have been introduced to the concepts more slowly. This would have been helpful for understanding the concepts.
12. It was really cool that having lectures and then have hands-on exercises right away together with the tutors. The demonstration with git-graph really helped me a lot to understand what is going on underneath. Parallel python with multiprocessing and mpi4py with a practical image process problem was a great idea. Learned concept and had hands-on exercise to have a deeper understanding. All in all, the lectures and exercises are very well balanced and I did learn a lot.
13. Preferred lectures that had fewer longer exercises - ones that took maybe 10-15 mins to complete, rather than many short exercises
14. The "negative" grades above (level too basic, subject not as interesting) have more to do with my personal view and previous experience with the topics. For someone with different experience, I am sure they would be much more interesting. A couple of the lectures came across as a bit less well-prepared (packaging, memory-bound problems), but even then the tutors were more than satisfactory.
15. A few comments:

    - I think in some cases we could have used just a little more time for exercises, but this is really minor. Overall I think the balance between exercises and lectures was great.

    - Each lecturer at the beginning of their talk should make it clear how they expect us to "take notes" for the lecture. Is it writing on paper? (like the git lecture) Will it be filling in empty exercises? (like the advanced numpy lecture, in which case the lecturer should provide the github repo url) Or are we supposed to panic and try to type everything the lecturer is typing? (not ideal for us) It would help if each lecturer takes a moment to orient us at the beginning so we can optimize our knowledge intake.

- On a related note, please have all lecturers who have code-heavy presentations (so pretty much everything except git) provide us the code in a repo at the beginning of the lecture. It was not always posted online, which was a little inconvenient.

- All lecturers should know how much content they want to present and make sure it will fit within the allotted time frame. Ashwin, in particular, did this extremely well by turning an exercise into a demo when he was short on time. I was really impressed by his efficiency and time management.

- You might want to consider some system for the lecturers to easily tell the students "hey shut up and listen right now". I personally have a lot of trouble listening to one person when there are other voices around, so sometimes when the lecturers were speaking and other pairs were talking to each other in a normal voice, I had a lot of trouble staying focused.

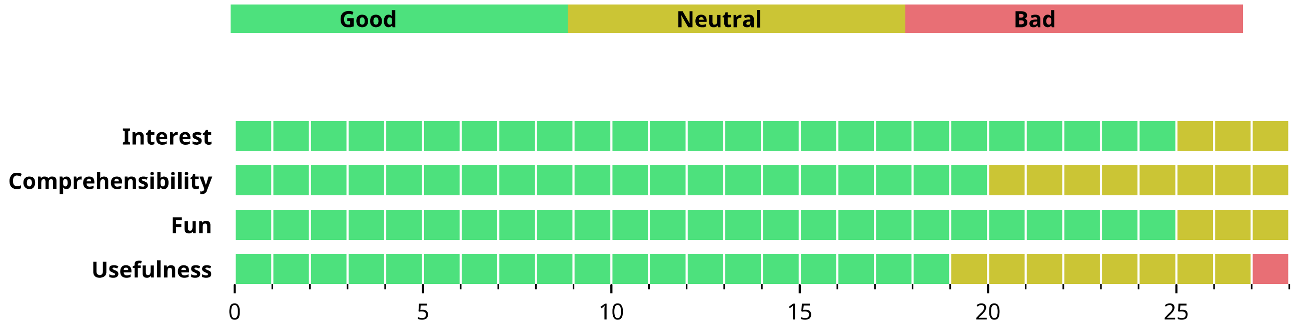
- It might be interesting to have a quick half-hour session on how we can continue to develop our skills. E.g., what books, online classes, conferences, etc. are good for keeping our Python abilities sharp?

I know this is a lot, but I'm really just nitpicking at this point. Overall I think the course was incredibly useful to me, and on my first day back I've already used a lot of the knowledge I acquired in Nikiti.

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: 89% (25)
No: 11% (3)

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

Yes: 15% (4)
No: 85% (22)

Q: Any further comments about the programming project?

1. Re: whether it should be a real-world problem: it seems like a good idea to have a scientific problem, but finding a topic that is interesting for enough people would be challenging, and the time frame feels too constrained to do a whole project (decide on the particular problem, understand, solve). I think the game is a great replacement, and everyone seemed to enjoy the fun and competitive side of it. At the end of the project, I felt much more comfortable with rebasing. I don't know if that's good or bad.
2. A W E S O M E
3. The programming project was very fun and I enjoyed working on the problems presented by it. The documentation and API could have been a little more developed so more time could be focused on the interesting problems rather than figuring out the basics of interfacing with the game. The most valuable aspect was figuring out how to work with your teammates. Though maybe not so common in science right now, it is becoming more likely that you'll have to work with others on a project that involves sharing code and working together on one codebase.
4. Introduction should be a bit longer and showcase more aspects of the api. Maybe live code a basic non-random player. Working with unmotivated people is frustrating.
5. I have the feeling that the competitive/fun aspect of the project consumes a lot of the available attention. A more "boring" project might be more useful for actually making use of and deepening the knowledge taught in the lectures (good programming style, version control, testing etc.).
6. Working on the project was fun and I think still useful. I noticed, however, a slight mismatch between the lectures and the project, in that the project had a lot to do with object-oriented programming, while the lectures had little to do with it.
7. I really liked the programming project! I would always prefer the game instead of a real world problem, because everyone can contribute, it's a lot more fun and you can focus on practicing what you learned in the lectures instead of solving a serious problem for which there wouldn't be enough time anyways. It was a good opportunity to learn how to work together with others on a programming project by finding agreements, distributing tasks and using git in a way it doesn't break. ;) The git lecture was a very good preparation for that I think. It was also a good way to practice the programming methods we learned in the lectures.
8. - Consider adding in actual points/rewards for PEP8 convention and test development. If the point is to have us use some of the things we learned during the week, this might encourage us to take a little more time to produce a high-quality product.

   - I think a few things could be added to the introduction that would make the beginning a little easier. First, as Tiziano already noted, an explanation of pair programming. Second, a clear list of deliverables and deadlines. Third, a recommendation for how to begin working as the group. For example, I would recommend that the group start by walking away from the computers for half an hour and discuss 1) their git workflow, 2) a rough timeline, and 3) a preliminary delegation of tasks. Obviously they don't have to follow this, but it might help the groups that don't have a "I'm going to take charge now" kind of person.

- I think the selection of the programming project is perfect, because you are working as a team for a competition, and the subject is not so complicated that we have a steep learning curve.

9. I am torn on the project. It was fun and allowed us to use some of our new skills, but ... perhaps not the most relevant to science. However we all study different science topics so finding a common task is tricky
10. Yes, for me it was demotivating that it was just a game, I would highly prefer a real world data-science problem that could potentially be published later on.
11. Well, it's hard to choose a scientific problem that is interesting to everyone, especially that we are from different scientific fields. On the other hand, it would be interesting to apply what we've learned about Numpy, packaging, scipy, machine learning methods, data analysis to a practical problem and get some result. By the way, it would be helpful that we have a data visualization session. It would be interesting to everyone.
12. Really enjoyed it - was definitely the highlight of the course!
13. I was a bit skeptic about developing a game instead of a real-world scientific problem, but it turned out that this is a way better way of learning because it is well-structured. If the programming project had started a bit sooner with half a day project and half a day classes I think more could have been achieved, and it would have been better for the group dynamics. Nevertheless, it was a good way of learning how to work together on GitHub and how to quickly ask questions. Even though everything we needed could be found on a website, this was sometimes hard to find. However, this teaches you to look up the documentation before you ask questions :-)
14. I am not sure that the team-programming project is relevant to my work as a programming scientist. I also think that most of the things we learned from the lectures could not be applied to the programming project. On the other hand the project was really appealing to my personal interests. I did something similar with a pacman agent in an online course with python before. Maybe the team-programming project could be of something that combines all the things we saw in the lectures.
15. The programming project was great. While I don't think the team based nature of it was relevant to my current work, I found it an eye-opening experience and will be more prepared for that type of work if I do ever encounter it in the future. It was also an excellent opportunity to implement some of the topics discussed over the previous few days. The main negative was the documentation for the classes and methods used as part of the project. While I think all the information needed was available somewhere, I found it very difficult to find precisely what I was looking for which made the project more frustrating.
16. In my case, most of my work is as an analyst, so algorithm development is not really my field. This made it so that the project was a little out of my reach, making it less interesting. Than being said, I think the project being a video game is the correct choice. If the project was a real-world scientific problem, depending on the field the problem comes from it may either be too easy for some or just as foreign as the video game. So I think the video game is the correct compromise and it lets the participants focus more on the programming more that the actual problem.
17. I don't find games to be very motivating, but it's hard to think of a real-world scientific problem that would be satisfying for so many different kinds of scientists. And there was a lot to learn from the game. The group sizes may have been a bit large: it seemed like most groups had at least one member who didn't feel like they were able to contribute much.

The School in General

Q: How do you overall evaluate the school?

Good: 96% (26)
Neutral: 4% (1)
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: 96% (26)
Too basic: 4% (1)

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: 0% (0)
Just Right: 93% (25)
Too basic: 7% (2)

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

Yes: 93% (26)
No: 4% (1)

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

Good: 96% (26)
Neutral: 4% (1)
Bad: 0% (0)

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

Yes: 100% (27)
No: 0% (0)

Q: How did you hear about the school?

Google Search: 5
Professor/Tutor/Supervisor: 6
Colleague/Friend: 12
Mailing list: 8
Other: [website] 1

Q: There might not be further editions of the school unless we find a way to make it a self-supporting event. Would you have attended the school even if a fee were introduced to cover the running costs?

Yes: 93% (25)
No: 7% (2)

Q: If yes, do you think a fee of about 200 € would be appropriate?

OK: 72% (18)
Too high: 12% (3)
May be higher!: 16% (4)

Q: Any further comments or suggestions?

1. It would be nice to organize a get-to-know-everyone session. We give a self-introduction in front of everyone about everything we'd like to talk about. So we know each other better than random chatting.
2. Overall this was a fantastic experience and I learned a lot. I would definitely recommend it to others.
3. perhaps some machine learning could be covered next year
4. The school was exactly what I was looking for. The topics were well chosen and fit the needs of data scientists. The tutors were friendly and always helpful. Lectures were well designed and there was a good balance between exercises and theory, which helped applying the concepts that were presented. This is the best training I ever followed, hence I sincerely hope that the school will continue to exist.
5. The school was great! All of my expectations were more than fulfilled. I learned a lot of new things that I was already able to put into practice back at home. There was a very good balance between lectures, exercises and project work as well as between courses and social activities. Despite the difficulties it was very well organized. The atmosphere among all participants and tutors was very friendly and cooperative, so there was no need to be afraid of asking questions. I really hope this school will continue for a very long time! :)
6. I really enjoyed the summer school and would like to support you guys in any way I can. A possible way to help support the school could be through private donations. I know that if I'm in a financial position in the future to do so, I most certainly would love to contribute. This could be done though some sort of non-profit or charity. But that may be too much hassle so it depends on how easy it would be so set up and operate.
7. I hope it becomes a self-supporting event with its own budget. Fingers crossed!
8. THANK YOU! IT WAS AMAZING. It was one of the best activities I've done in both my professional and personal life.
9. I entered the school with high expectations, and you exceeded them. Thanks very much.
10. I said "Too high" for EUR 200, assuming accommodation etc would be extra. Because the answers are not nuanced, I want to make it clear how great I think the school was. I learned a lot, much more than I have in previous attempts with online tutorials, and even in subjects where I did not expect it. I think you managed to provide something for a broad range of backgrounds, without overloading the less experienced students or boring the more experienced ones. It is obvious how much effort and love you put into organising this. I would not hesitate at all to recommend future editions of the school (fingers crossed). Seriously, congratulations and thank you.