

ASPP Asia-Pacific 2018

Melbourne, Australia, 14 - 21 January



Advanced Scientific Programming in Python Summer School

By the INCF, Melbourne Bioinformatics, and the Florey Institute of Neuroscience and Mental Health

January 14 - 21, 2018. University of Melbourne

Method

This survey was administered via a web interface created with LimeSurvey.

All answers were submitted by February 21, 2018.

No answer was mandatory.

The free text answers are unedited and are presented in their original form, including any typos that might have been present. Some answers have been redacted, however, when we felt that they would be easily de-anonymised.

Comments from the organisers

In January of 2018, we had the first ASPP summer school outside of Europe. (This was a parallel workshop to the European one, which will be held in Italy in September 2018.) In general, it was a great success, with some caveats that we will elaborate on below.

First we want to note that this school was a bit different than the European ones, in that we only had attendees from Australian institutions, where the European school has broad international representation, including some from out of Europe. This was in some ways inevitable, as it is more expensive to travel to Australia from almost anywhere than to travel within Europe. On the other hand, we advertised relatively late, and we were unable to secure travel grants during the advertising period, so there is hope that a future edition would be able to attract a more international crowd from the Asia Pacific region.

Given all this, there was a question as to whether we would be able to capture the atmosphere of the school, which normally sees the students living together and socialising for basically the whole week. In this case, most students just went home after classes were finished. But although some of that atmosphere was missing, by the end of the week we did manage to get some close links between all the students and the faculty. The evaluations below show that most of the value of the school was preserved.

We note that 100% of the respondents (29/30 of the students) would recommend the course to their peers. So, although some lectures were better received than others, and although the programming project was not universally loved, we managed to provide value for everyone. All of this is in line with the evaluations at previous schools (available at <https://python.g-node.org/wiki/archives.html>).

The project, which consists of programming a videogame bot, is controversial every year, but, consistently, more people like it than don't, and people get to practice git, pair programming, and programming as a team, which is the single most difficult skill to practice when programming for science. Indeed when we walk around during the project programming sessions, we see people

extremely engaged in what they are coding. It's difficult to imagine a scientific problem engaging such diverse people as the school's attendees (which come from very disparate scientific fields).

Of all the feedback, two particular statements, we hope from people in the same project group, broke our hearts. We decided not to include them in this report, because they might be easy to de-identify by group members, but they boil down to the following: a group member, by being combative and rude to others in their team, and deciding to essentially complete the project by themselves, ruined the programming project for all of their team members, with some even feeling that they were not good enough to contribute. This is tragic, because we want everyone in the school to feel *empowered* to do anything at all in Python.

Absolutely every student has something to offer in this project. Here, as in life, teams are comprised of members of varying skills. But we know from our selection that everyone has the skills to contribute (and this is confirmed by the fact that most attendees, for most lectures, felt that the difficulty level was "just right"). So if a student felt inadequate, it can only be because of the toxic team member.

Ned Batchelder recently wrote an excellent blog post about what he calls "Toxic experts" and what Tiziano Zito calls, somewhat more bluntly, "Arrogant assholes". (In discussions about this post, Tiziano and others noted that one does not have to be an expert to be toxic, or arrogant, or an asshole. No matter: the points below apply equally to anyone meeting any of the above characteristics *regardless* of expertise.)

The feedback we received should serve as a warning to selection committees and hiring managers everywhere about how damaging it is to allow such a person into your ranks. Due to the anonymous nature of the survey, we can't tell whether there was one or two toxic experts in our midst, but if it's one, they soured the school for five other people. If it's two, then that's ten people, a third of the school, that might have had a terrible experience. The problem with toxic experts is that they can so quickly cause damage to so many others. Thus, even if they are a mythical "10x engineer", **they are not worth it.**

Literally nothing that the above-described team member could have done, coding-wise, could make up for the damage they caused. Despite their strong opinions, they missed the entire point of the programming project, which is not to win a medal, but to *learn about working in a team*.

We try to avoid toxic experts in our selection process for the school, but they slip through every so often. In response to this feedback, we will aim to be even more vigilant in our selection, and also make the aims of the project *as a learning exercise* more explicit during its introduction. We will also make sure to be more aware of group interactions during the actual school; we apologise to the students involved that we did not catch this behaviour this time. We are truly sorry.

If you are in the position of being an expert during a school or workshop, don't go it alone. That is a waste of your time, because you can do a programming project on your own whenever you damn well please. Slow down, and think instead about practicing your teaching and mentoring skills. They are also important in life, and, in many contexts, they are your responsibility.

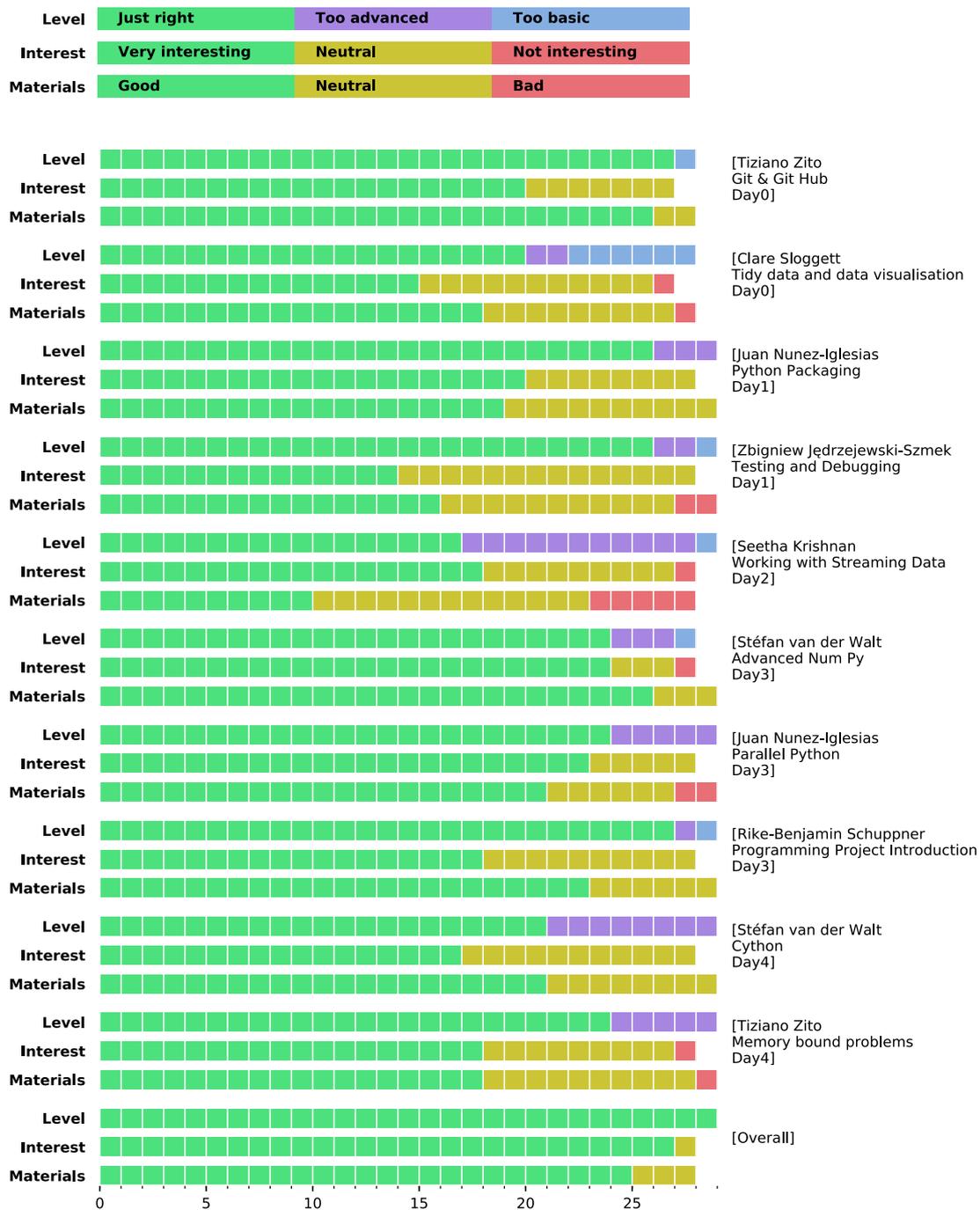
– Juan, and the Organisers.

Lectures and exercises (overall rating)

We asked the following questions for each lecture, and for the whole set of lectures overall:

- Grade the LEVEL of the lectures: just right, too advanced, or too basic?
- Grade the INTEREST (or relevance) of the lectures: very interesting, neutral, or boring?
- Grade the QUALITY of the lectures (including materials, exercises, clarity): good, neutral, or bad?

The results are summarised in the following figure:



Lectures and exercises (freeform questions)

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

- Yes, I have learned from my partner/I have helped my partner : 24 (83%)
- Other : 1 (3%)
- Neutral. It was OK, but I could have worked by myself as well. : 4 (14%)

Other answers:

- Somewhere between no and neutral. It wasn't a waste of time but I think splitting up the tasks made it a bit inefficient at times

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

- The time dedicated to lectures and exercises was well balanced : 22 (76%)
- Lectures were too long, there should be more time for exercises : 3 (10%)
- Other : 3 (10%)
- Lectures were too short, there should be more time for lectures : 1 (3%)

Other answers:

- I think the balance was good, as people who finished early could go to tea but others could stay behind and continue to work on it.
- Mostly well balance. Too much time spent on broadcasting exercises in the numpy lecture.
- Simply more time for playing around an exercises would have been VERY valuable. Having a more remote location with floks staying would be perfect.

Are some of the topics presented in the lectures not relevant for a programming scientist? [Leave blank if all topics are relevant].

- [3] The programming project did not really exercise many of the things that we had been taught, only git/github usage. It would be great if the project could be reformulated to be more about scientific problem solving.
- [5] Twitter streaming. Streaming data in general is relevant, but streaming using the twitter API is not very relevant.
- [9] I found the testing and debugging class very interesting, but I don't know if I would use it any time soon. Most of the code I write serves a specific purpose and then maybe gets pulled out of the code grave yard if I need to do something similar again.
- [17] Git and Github was taught first, but due to the layout of the course is only relevant for the programming project, so should have been taught later.
Tidy data was given too much time. This could be re-titled "how to stop csv importer failing" and can easily be learned on your own. It should have been touch on only briefly.
Way too much time was spent on how currying works, without a good example of why it is relevant.
Too much time was spent on broadcasting, which is useful to know how it works if you're using it, but only really relevant in very niche contexts.
- [19] TL;DR: Teaching a single tool to do task X is enough, teaching multiple tools to tackle the same task X can be confusing and isn't necessary to get people going with task X.
A general comment I have is that when you want to teach a concept (e.g. parallel programming, or any other concept), it is probably better to showcase ONE tool that does it. Yes, we know there are many tools, with each of them having pros and cons. But by showing me 4 tools that aim to solve the same problem, I get lost and confused. Whereas if you show me one good all around tool to do X, I might actually use it. You can mention at the end that there are other similar tools, which I can look up later when I better understand the limitations of the approach I've been applying.
- [21] I felt that advanced numpy and memory bound problems could maybe have been condensed to some relevant tips and tricks. While this was all very interesting, I didn't take much practical knowledge from this for how long the lecture was.

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

- [2] Finding time to go over more complex python elements (classes, decorators...) would be beneficial
- [3] Generators. Decorators. When and how to use an object-oriented programming style.
- [5] Topics on how to profile code would have been hugely beneficial. We had several sessions on different ways to speed up code, but nothing on how to judge when it's worth investing the time to do that, or how to be strategic about which parts to improve first.
- [6] Ecosystem simulation modeling - while the basic tools to achieve this were taught it would be great if a day could be dedicated to ecosystem simulation modeling in future courses
- [7] I think integration with supercomputers could have been interesting, e.g. how to use the slurmified package.
- [9] I think there was this general vibe that we should be writing code that can be reused, but never really covered how to write reusable code..
This is related to my response above, I generally only do one thing, once. So the code I write is pretty specific. Other than not hard coding in any variables, I wouldn't know how to go about making something more general/versatile
- [11] I wish we had more time for the Advanced Numpy and Cython.
- [12] I think the chosen topics were relevant
- [16] Data storage, management and retrieval strategies?
Interfacing with hardware?
- [17] Too many things were skipped over that would have been useful.
Using matplotlib to generate publication quality plots needed to have more time spent on it. For example, how to add secondary axis, insert arrows and overlay text, use loops to automate adding multiple plots with relevant labels, use symbols for black and white publications etc etc.
Profiling was completely skipped over, which would have been very useful. And I would have liked to have seen debugging include using a gui based debugger such as those integrated in spyder or pycharm.
Not enough time was spent on the various packages available in scipy. I would have liked to see someone touch on things like fourier analysis, image processing etc etc. Personally I would have like to see a lecture on linear algebra solvers, which could easily have been linked in with the numpy lecture and the parallel programming lecture. From the conversations I had I suspect none of the tutors have the relevant knowledge to teach those topics though.
I would have liked to see some time spent on the python implementations of advanced programming features (Classes, decorators, exceptions etc). One good example was in the streaming data lecture, when too much time was spent in the weeds of explaining how curried functions work. That time would have been better used explaining how decorators work and then using the curried function decorator as an example of a useful application for a decorator in a streaming data pipeline.
- [19] I think this was an excellent balance of things to cover and I thoroughly enjoyed this workshop. Kudos!
- [21] Maybe a half-hour on bash scripting would be useful (and how to get parameters from bash into python scripts). I'm aware that you are able to find excellent online resources for this, but in my experience working with other researchers, it's something that's not on people's radars but makes everyone's lives much much better.
As for the data visualisation lecture, I think having more interactivity for data analysis would have been great. Options for this would be plotly+dash (to make effectively web apps - but with really not much work) or Holoviews. Both tools allow for dropdown menus, sliders etc. that would be useful for larger datasets. Holoviews also integrates well with pandas.
- [22] Decorators!
- [25] Numba
- [27] I thought it would have been helpful to include a short introduction on object oriented programming as I don't usually use Python this way, and predominantly use Matlab in my work.

- [28] None that I know of at the moment

Any further comments about lectures and exercises?

- [3] The lecturers were awesome - it was clear that they were very well prepared. Tiziano was particularly engaging. speaking mostly without slides and with everyone's full attention.
- [5] I liked it when there was a mix of exercises in with the lecture, rather than having all the exercises at the end.
- [6] Thank you! Overall it was one of the best programming courses have attended (and I have attended a few)
- [7] I really enjoyed all the lectures! Advanced NumPy and Parallel Python were probably my favourites and will be incredibly useful to my work.
- [9] The live demos are great to see things actually happening, but things went wrong quite a bit. Anything to reduce this down time would be great :)
- [16] Wonderful lecturers and well-designed, thoughtful content. I found the course interesting and valuable, and it has already had an impact on the way I am approaching the design of an embedded system in my research. It has also prompted me to revisit some old problems with a fresh perspective.
- [17] The course seems to struggle with balancing between between "advanced" and "introduction to" style teaching.
 "Introduction to" style teaching is useful, particularly when covering a lot of topics in a scientific programming course, however for a course with "advanced" as the first word in the title I found the balance disappointing and I would have like to see some more advanced programming topics covered in more detail. Ultimately this occurs because you are trying to cover too many topics in very limited time which limits what you could do. I believe you could fix that balance by splitting the course into two courses, and having beginner and advanced courses that don't try so hard to balance content quantity with quality.
- [19] - Kind, helpful and knowledgeable organisers (5 stars!)
 - Reusable mugs that we can take home were a great idea
 - Having well planned framework (Pelita) for people to tackle a project in groups is excellent
- [20] I preferred the exercises embedded within the lectures rather than lecture followed by a block of exercises.
- [21] This was by far the best workshop experience I've had and I'm very grateful that I was allowed to be part of this!
 ..you've set a high bar on relevance and enjoyment of a workshop, so please understand my comments in that context, too!
 So here comes some minor nitpicky feedback (because you're asking)
 - Testing lesson would benefit from an exercise that takes you through the entire pipeline on a relevant feature. I think we started with a good example (writing a function that does a certain thing). The next thing could have been to come up with edge cases, implement a test, and then run it. Integration on our personal Github accounts with Travis would be A+
 - The debugging lesson made me not want to use a debugger. I think the only problem was that we used a version that doesn't highlight syntax, so that's easy to fix :)
 - I might have not followed this correctly, but it seemed like we learned 3 ways of using parallel programming, where the last one was the one we'll end up using. Skipping the first two might free up some more time.
 But as I said, I absolutely loved it.
 I'm still impressed by how well prepared the course was. Especially the group project.
 <3
 Thank you!

- [23] There should be way more time for exercises and just geeking out together. Putting stuff into practice immediately really helps the uptake. I know the location didn't allow for this this year but having more of a bootcamp atmosphere would have been awesome. I would have retained more with more exercises.
- [26] I was very interested in parallel programming before the lecture started and then got totally lost just few mins after the lecture started. Commands Juan typed and what he did were too speedy to catch on. I personally feel very sad on missing it because it would have been really useful for my research.

Working with Streaming Data was interesting but generators 'yield' was completely not useful for my work and I believe it will just clutter an already messy program. But I am not dealing with large quantities of data and only works with bunches of images. So it's relative to the field I guess.

Tiziano's lectures were just amazing and perfect. I wish he could do more sessions because he can explain even the hardest things quite simply. Good work.

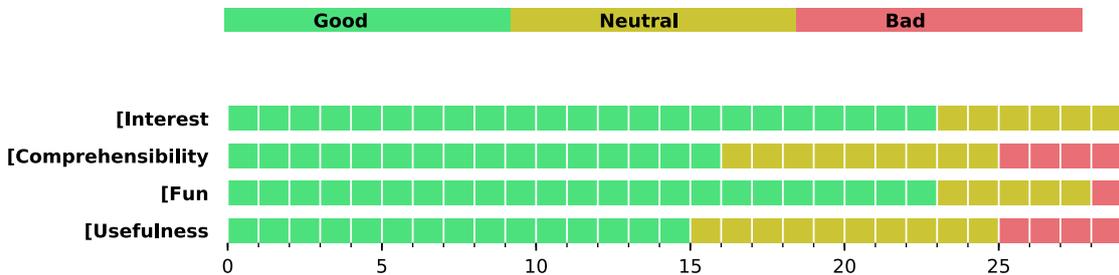
- [27] When introducing new topics such as streaming data, it would be helpful to limit the material and go through them more slowly. Many students, including myself, felt overwhelmed by the fast pace and the amount of materials.

Programming project

For the programming project, we asked the following questions:

- Was the programming project INTERESTING?
- Was it EASY TO UNDERSTAND?
- Was it FUN?
- Did you learn USEFUL things doing it?

The results are as follows:



We asked more detailed questions, including freeform text, to allow respondents to elaborate on their views about the project.

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

- Yes : 22 (76%)
- No : 7 (24%)

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

- No : 19 (66%)
- Yes : 10 (34%)

Any further comments about the programming project?

- [3] .[REDACTED]
The participants were selected based on their research and experience with scientific computing. Many people new to scientific computing work in a fairly procedural fashion. The project required a reasonable level of familiarity with object-oriented programming methods. One team member [REDACTED] didn't know where to start and didn't really feel like they could contribute.
Even though it was fun to watch the tournament at the end, I think it would be better to align the project with the scientific programming used elsewhere in the course.
- [4] .[REDACTED]
- [6] It would be less fun, but more relevant if the programming project would be dedicated to simulating a system - this days many scientists work with simulations under different scenarios (i.e. what would happen to an ecosystem under different climate scenarios and how would it affect it if different management actions were applied to it) - I think this would be a lot mire useful than a game.
- [7] I thought it was a lot of fun, and a good way to finish the course. It was also a great hands-on way to learn how to use git and GitHub.
- [9] I like it as a game, it's super fun! A scientific project will be a bit boring and won't bring out the team spirit. Especially is someone is an expert in that field. A game put as all on an equal ground :)
- [10] Essentially more depends on the team than the actual project. Maybe something less competitive and open-ended could help "constrain" some people's "imagination" , but then would also be less fun!
- [11] I wish we had more time for the project because it requires a lot of time to think and put our thoughts into practice. The documentation was very good and detailed but it was hard to find out whether some functions or classes have been available or not. We spent an afternoon trying to write a function for a move then turned out it was available. Again, because of the limited time we didn't read the documentation very carefully as well. Maybe if you can somehow make things more visible and structural on the front page of the documentation it would be easier for us. We really enjoyed the project though.
- [13] It was great to have a chance to work in a team-programming exercise. I haven't done this properly before, but definitely saw the advantages of working in a team, and would definitely try to do this more in the future.
- [14] It was awesome ! I loved that the groups were well gender, background and expertise balanced
- [16] I really enjoyed this course, and I feel the project may have been a valuable experience for me had I not become swamped down in other work and then too ill to usefully participate on Saturday.
- [17] I'm torn about the project. It is a perfect exercise for learning to program with git/github, and the team programming aspect is useful, particularly for scientists who tend to program solo. However, as fun as it is, programming a simple AI is completely irrelevant to scientific programming. Yet I don't want you to remove it from the course because its great..... Torn.
- [20] I think there needs to be a short intro to classes prior to the project activity.
- [21] I went back to work trying to convince my colleague to code together.
It's something software developers have been doing and it's working. So why shouldn't scientists?
We'll try it soon. :-D

- [23] Great!
- [25] It would be better if the programming project introduction is given on Day 1 as some students like me do not tackle a program as fast as other group members. If the program introduction is given earlier, then the students who work slowly could have enough time to think of the problem and be involved more actively in the group exercise.
- [26] It was hard to learn all the wonderful things in just one week. After getting the head so full in daytime it was impossible to do a re-check at night. It was harder to keep up as the days went by so that I personally (and most people I talked to) had given up trying to get hold of everything that was taught. Rather the goal was to get the basic and do a recap later after the school. This meant that when team project was given, i personally hadn't got hold of all the new concepts. So when coding I just coded as how I have always done (no iterators or any cool newly learnt techniques, just ones I have worked with before summer school). So the project was actually not useful to me since it was just another project that I worked on using the programming knowledge I had before the school. I personally would have liked to skip the project, stay at home and re-capped all i learnt.

The school in general

How did you hear about the school?

- Professor/Tutor/Supervisor: 7 (24%)
- Colleague/Friend: 9 (31%)
- Google Search: 2 (7%)
- Mailing list: 13 (45%)
- Other:
 - VLSCI
 - From the school of Physics, Melbourne University, mailing list
 - melbourne bioinformatics website
 - twitter

How do you overall evaluate the school?

- Good : 28 (97%)
- Bad : 1 (3%)

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

- Just right : 24 (83%)
- Too advanced : 4 (14%)
- Too basic : 1 (3%)

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?

- Just right : 25 (86%)
- Too advanced : 3 (10%)
- Too basic : 1 (3%)

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

- Yes : 29 (100%)

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

- Good : 23 (79%)
- Neutral : 6 (21%)

Would you recommend this course to other students and colleagues?

- Yes : 29 (100%)

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?

- No : 5 (17%)
- Yes : 23 (79%)

Do you think a fee of about 500AUD would be appropriate?

- Too high : 16 (55%)
- Ok : 6 (21%)
- May be higher! : 1 (3%)

Any further comments or suggestions?

- [3] I don't see why it was necessary to fly in so many of the lecturers. Australia has plenty of good lecturers in computer science, python programming and all genres of scientific programming. I would love to see the course continue here, and I don't think that we need to belittle ourselves by thinking that international is better - there is plenty of local talent.
- [4] Overall, I found this to be an extremely valuable course and I've taken away useful practices and started applying them to my work. Thanks to Juan and the organisers for all of their hard work setting up the program and for providing this fantastic opportunity.
- [5] The answer to the question above is "it depends" on the fee, and whether or not my university could cover it.
- [6] The fee would depend on what it included - for Australia it is difficult for postgraduate students and early career researchers (most of your students were in this category) to get funding to cover extra curricular workshops and conferences. It is a bit easier for students than postdocs.
The average national travel grant is about \$500 AUD but that should cover flights, accommodation and course fees, anything extra comes out of the student's pocket.
If the \$500 is just the course fee, then it is too high, if it covers accommodation as well, then it is ok.
- [7] I really appreciated being a part of the course and would like to thank the faculty and organisers :)

- [8] My main suggestion is that as we learned so many topics in the school and there were all great, why don't you put a nice comprehensive course on Coursera or EDX? That would be very valuable. I know there are many python courses there, far from the level of this course. I beg you to do it.
- [9] I think keeping the school as a free course seems really integral to it's purpose. But if you were to charge, \$500 seems to be inline with other workshops I have attended.
- [10] Keep it up!
- [13] Only in hindsight do I think the school was worth paying for. I don't think I would ever have signed up if there was a cost as there's so many free online courses floating around, and I didn't actually realise how much can be learned in person instead.
- [16] A wonderful program. Thank you!
- [17] Something about this survey is broken. I spent about an hour writing details responses and when I hit next the same page reloaded and it cleared what I had written. So now you have my much briefer opinions.
- [20] You changed the way i worked and think about scientific programming - for the better. You made me feel more confident by showing me that i'm not as inept as i thought - the instructors were all so positive and encouraging that it made me feel like i can do it and i can be part of the python sci prog community. THANK YOU!!!
- [21] Oh no, I think I used an earlier field for this. ...sorry... flip back a few pages, there's all the praise.
But maybe one note to the above: 500\$ seem fine for the content, but I'd be worried that PhD students won't be able to pay this and it's going to become a course for the (even more) privileged.
- [23] Happy to help make this happen more often. This is an awesome effort.
- [24] Regarding the fee, it's a complicated subject. If a supervisor is paying, then 500AUD might not be too high. Without a supervisor paying, it may be for postgraduate students.
- [25] I would prefer the venue is in a remote resort rather than the city if the event is charged.
- [28] Excellent workshop! Thank you very much!
Just a comment on the survey: Would recommend doing it right after the course finishes, might have been able to give better feedback.