Henry Transcript

Katie [00:00:01] OK. So um, can I just start by asking if you have any sort of personal bias towards either Python or R.

Henry [00:00:13] I would say I am a, I mainly use R, um sorry, I mainly use python, but very recently I've started using R for a project I'm working on.

Katie [00:00:26] OK.

Henry [00:00:27] So yeah. Yeah, I'd like to think- I'd consider- I'd like to think I'm unbiased, I'm not I am a python biased person.

Katie [00:00:34] I think everyone that programs probably has a bias towards some sort of language. Er, do you have any sort of initial comments on the code or like the readability, similarities, differences?

Henry [00:00:49] Uh, I mean, I mean yeah. I mean how about like what kind of comments are you looking for? I guess um, with-

Katie [00:01:01] Um, just anything that's popped into your head while you've been looking through it.

Henry [00:01:05] Um, well with the with the matplotlib stuff and using just the straight plt dot whatever, rather than creating a figure and an axis and adding stuff to the axis directly. That's one thing that I notice, I understand like there's reason for doing that because it's meant to be more readable.

Katie [00:01:23] Okay.

Henry [00:01:24] But like, erm, I wonder whether I, I personally I find that harder to read because it's harder to keep track of where things are kind of like how R code is, like R code, like there is just this sort of thing that's there and this is the figure that we're talking about. And then when you move onto the next thing, um, a new figure is created and. But yeah, otherwise, yeah, the code looks acceptable. I can imagine what the code does, which I think is the most important thing for both the R code and the and the Python code.

Katie [00:02:03] All right. And do you feel there's any way either of the codes could be changed in your opinion?

Henry [00:02:15] Yes, so, yeah, the the fig.ax matplotlib thing is something I would I would I would do because like where you're setting, like the y-scale, x-label and y-label and x-ticks, you can do all of that in one call if you do it the other way.

Katie [00:02:33] All right.

Henry [00:02:36] And, yeah the same with the R code there, like, where you have like xlab, ylab and then labs, you can just do one labs call, and just have all of them together, those kind of things. The code, I think functionally is all correct. But yeah, in terms of cleaning up, I guess that those will be my only comments.

Katie [00:02:55] Okay, that's helpful.

Henry [00:02:58] I don't know, is that what you wanted?

Katie [00:03:04] Yeah, just sort of like, just general comments of like, oh, in your opinion, how would you change this? Which you've like, answered well because I hadn't thought of doing them.

[00:03:13] Yeah, yeah no, functionally-

Katie [00:03:16] I'm not the most efficient programmer.

Henry [00:03:20] Functionally it all- it all looks looks good, but yeah.

Katie [00:03:26] All right, and sort of- so based on sort of your knowledge of programming and sort of like the two sort of Visualisation packages and these codes, how well suited do you think each seems to visualisation? For each language?

Henry [00:03:44] Well, that's a really good question, and I think I think matplotlib is great. I think it's great because you have a huge amount of control. It's kind of difficult to do complicated things very quickly in matplotlib, whereas ggplot2. You can have very, very complicated data in whatever form, and you can plot it in a relatively complex way very, very quickly. Like oh, I want to make a stacked bar chart and you just say, give me a bar chart and change the position to be stacked and it will do it, it'll do that for you, whereas in matplotlib you have to say, oh, no, this is where the bottom is now and you're setting them over the same x value, that kind of thing. So I think if what you want is to have, like, nice looking images straight away, R makes sense for that. I don't know how good ggplot2 is for doing the more complicated things. Whereas in matplotlib you just have control over every single element and it's very easy to access it, if you are familiar with how to access things, access objects in Python, matplotlib is a very, very sensible tool to have.

Katie [00:05:02] Okay, yeah, that's good. And sort of on, I guess, sort of a follow on note, which language- I'm going to show you a couple of the plots, and I just want to ask you which language you feel has a more publication ready output if you if you think either of them do. I'll just show you this, so these are just sort of the control plots that I haven't really done anything to I haven't done anything to the axes.

Henry [00:05:35] Yeah, um. They both, they both look good, like the things that the things that I look for, I guess, you know, good- good um relative size of like fonts and labels and stuff, I would argue that that's a little bit clearer on the right, which is the one made in, um, matplotlib.

Katie [00:06:03] Yes, yeah.

Henry [00:06:05] Whereas the one on the left, it looks cleaner and there's less- you know because there's fewer lines and stuff and, the uh, the size of the the labels and stuff is a little bit small compared to the size of the figure, I would say.

Katie [00:06:21] OK.

Henry [00:06:21] Um. But that's, again, as you said, you haven't changed anything, so.

Katie [00:06:30] And then just a couple more where, these ones are-. Oh I- apparently I opened the wrong image. Sorry, um, but yeah, the ones I'm going to show you now will be, I changed it to using a log scale um, and- yeah so this is just the output you get when you take those two plots I just showed you and just specify that you want a logarithmic scale.

Henry [00:07:08] Yeah.

Katie [00:07:14] Apparently I've closed both, just subconsciously, in the last one, sorry.

Henry [00:07:19] That's alright.

Henry [00:07:22] How are you finding working with Vince?

Katie [00:07:24] Er good, yeah, I mean, he's been my personal tutor since first year.

Henry [00:07:31] Oh nice.

Katie [00:07:31] Which is nice, yeah, so kind of. It's been good, we end up just talking about skating and things.

Henry [00:07:44] Cool, er, cool. And these are the these are the same plot.

Katie [00:07:50] So these are one- so it's the plots before that I've just applied a log scale to, so a specified in the code, um, I want a logarithmic scale. And again, it's just um, which do you feel is more sort of publication ready at the output?

Henry [00:08:05] So with this one, because it's a log scale, I would say the matplotlib one. One hundred percent, because the scale starts at zero.

Katie [00:08:16] Yeah.

Henry [00:08:17] And even though you can't have, like, log of whatever is never zero, having that I think is, it's kind of important because otherwise, otherwise it could be misleading and the same way the bar plots where you have a mean with a standard deviation error bar at the top can be a very misleading way of showing data, and in the same way I think not having the full scale is misleading as well. Also, I think it's clearer that this is isn't and is in the log scale because the the the Y ticks are shown as what they should be, whereas in the ggplot2 one, um, they're not. But you have like ten to the zero, ten to the one, and yeah obviously you could have the, the- what did you call them? Like the individual, the sub gradient things, whatever in the matplotlib one, you do get them sometimes, but um, yeah. If you were just looking at the plot, it's very clear that this is in the- the one of the right is in the log scale whereas the one on the left, it's not necessarily clear that it's on a log scale.

Katie [00:09:30] OK, yeah.

Henry [00:09:38] Cool.

Katie [00:09:38] And sort of finally. So which of the two sort of plotting libraries do you think would be easier for sort of beginner to visualisation to pick up if they had an equal amount of sort of R and Python experience?

Henry [00:10:01] Um, so as somebody who's picked up ggplot2, in the past, like two weeks, I would say that's easier because it is very clear like, you have your geoms, you have your stats and you just mush them together and it all- you just have lots of pieces that can all work together.

Katie [00:10:18] Yeah, okay.

Henry [00:10:18] Whereas with um, with matplotlib you have to have an understanding of what you're actually trying to do, like from a programmatic, programmatical point of view, like you have to think about this is the dimensions of my array and stuff like that whereas in ggplot you don't. So I'd say ggplot is easier to pick up.

Katie [00:10:36] Yeah, I think um speaking to someone who's also sort of, on placement ended up picking up ggplot in like a couple of weeks, yeah I kind of agree with that. And do you have sort of like, any other comments on the code that have been brought up or that you've just thought of?

Henry [00:11:01] Erm, not- not really. Yeah not really. It looks all- looks good.

Katie [00:11:10] Okay, thank you. Thank you so much for your time as well.

Henry [00:11:15] That's quite right. Don't worry about it

Katie [00:11:17] Hope you enjoyed the short interview?

Henry [00:11:22] No, it's been great. It's been really great. I'm glad I could provide some qualitative data for your work. But yeah.

Katie [00:11:30] Okay, I'll stop the recording now.