
Q1 : Do you have any particular bias towards either language;
R or Python?

R biased: Andreas, Owen
Py biased: Geraint, Henry, Nikoleta, Vince

Andreas (R)

- "I- I prefer a much, much more familiar with R than Python."

Geraint (Py)

- "Yeah, I use Python mostly, but I'm fairly new at R"
- "I probably understand the Python a lot more."

Henry (Py)

- "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."

- "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."

Nikoleta (Py)

- "Yes, I am familiar with both languages, but I am very biased towards Python because it's the language I use mostly and I'm way more comfortable in Python than in R"

Owen (R)

- "Yes! I'm an R person. I wrote a book on R, so..."

Vince (Py)

- "So I'm a lot more familiar with Python than I am with R or a lot more fluent in Python than I am with R so that's already one bit of bias in whatever I might suggest."

Q2: Do you have any initial comments on the codes? For example in terms of readability, or similarities and differences between the two.

Andreas (R)

- " I found it very, very busy. You know, I know- I know you're using the code to change some different parameters. But for me, when I do plots, visualisation and all these things, I, I do the you know, I rarely deal with them different parameters, I'm just doing a simple plot as possible that gives out the message as clear as possible."

Geraint (Py)

- "It's obviously a big difference in how R and python like implemented,

right? I've only just got to starting to know about ggplot and I quite like the way ggplot does things. It's quite nice to do things on."

- "I'm obviously seeing the python a lot easier to read because I'm more used to it."

Henry (Py)

- "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."

Nikoleta (Py)

- "pretty much if you are more fluent in one language, you can always translate it very easily to the other one. Um, so there were commands I didn't necessarily know what they meant in R, but then I could look to the right to the Python code."

- "I guess that means they're quite similar, they use similar syntax for several things."

- "So, for example, some of the pre-recorded functions such as, you know, changing the scale and things like that. This is not exactly the same, but it's very similar."

Owen (R)

- "One impression, strong impression I had with the code was to do with the size of the functions, and it's something that many people have noted, not- not just me, but small functions are much easier to digest. So so some of your code was written in, know, little sort of 10 line bites or chunks which are sort of then stuck together. And that, of course, is is much easier to to read than a great big, long things."

- "It seems to me that your R code, those seemed to come in longer bits than the Python code."

- "if I had to be honest, I say I probably found the Python code easier to read than the R code, in terms of what it was trying to do, but neither of them were particularly difficult."

Vince (Py)

- "looking at the Python one is I like that you've put your plotting code within- um in a functional way so within functions."

- "that that all looks like standard matplotlib to me."

- "Of course now taking a look at the R. And I suppose in a way, I have the the same comment that I like, that it's it's functional"

- "maybe, they don't call the strings in R, but a leading overall bit of documentation at the start would be nice. But I understand as well that that could take up quite a fair bit of the screen"

Q3: Do you feel that either code could be changed in any way?

Andreas (R)

- "Yes, it can be changed. And in the sense that that is no reason to. I feel like sometimes there is no reason to complicate the complicated code just for visualisation purposes. Sometimes you can achieve whatever you want with a much simpler code. Yes, but at the end of the day, to what you like to do, right? I mean, that is no wrong or right, I'm happy either way."

Geraint (Py)

- "OK, so speaking about the Python code first. There's a few things I've noticed that maybe I wouldn't have done, but like I'm really not that fussy about any of this. For example I know you've got, I believe, as your data is all in camel case. I don't think that's normal in Python. I think we usually do the snake case, is what they call it?"

- "And there was a couple of areas where I think you you defined local variables as the same thing as a global variable." (ie. overwriting viridis)

Henry (Py)

- "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."

- "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."

Nikoleta (Py)

- "Yes, for sure. Again, I'm not too familiar with, with R. So I don't know if I have any effici-. I guess if you argue to me that this is the most efficient way to do this, maybe I would have- I would agree. But if you go to bar plot one, for example, you are overwriting the plot each time, so that's an error, in my opinion. And at some point in both codes, you're referring to these things called names and ntimes, which are not defined anywhere past, even in the functions."

- "So my opinion is that the both codes right now, there are things that wouldn't work necessarily, right? Like if I run this it wouldn't work and there is some it seems to be some bugs as in the plots are writing themselves. So you're not returning each plot, so that seems to be- so from functionality, I think the functionality is a bit broken."

- "I think both languages, it's very good as in it's very obvious what is the colour is very obvious that you're changing the y labels and the x labels and things like that."

Owen (R)

- "Well, I suppose unless I knew what was being used for, I wouldn't I wouldn't be inclined to to try, I suppose. I suppose the other general impression I got is this is a sort of code I would expect to be sitting behind the scenes with some other sort of interface for me to use. In which case, as long as it works, I don't really care too much, I suppose."

Vince (Py)

- "I think for me, the main one would just be for the Python at least, and for the R, you know, adding some leading documentation in the functionality. But my GGG, my GGG, my ggplot is certainly rusty, so I can't suggest much. But apart from that, the matplotlib looks like essentially the way to do it. I don't normally use those colour codes myself, whatever they are called, the hexes, are they hexes, are they hex code? I never use those I normally use like the RGB thing. So like. In a very subjective way, that kind of stands out to me a bit. But I am beginning to wonder if actually it's nicer, more precise to use those standard codes as opposed to like the RGB, you know, vectors. But yeah, no, I think I think that's that's all that's those are my main thoughts on the readability."

Q4: Based on these codes and your own knowledge of programming, how well suited you think each language is to visualisation?

Andreas (R)

- "I know that in R, you use that special package, the ggplot package, which theoretically is suited for visualisation."

- "In Python, I am not sure, it doesn't seem that you used any special package. So I'm not I'm not really sure how to compare lets's say, um, the two."

- "I will say that er, if you- if you- if you have used the, let's say the standard packages in R, I would have told you that maybe maybe not the wisest choice to do if you want to change the parameters, but using the ggplot package at least, you know, if you use something that is specifically designed for visualisation, so it should be suited for visualisation, okay?"

Geraint (Py)

- "I don't really like saying that one is better than the other or whatever."

-"I think they are, they are both really well suited, especially with the ggplot library. I think the ggplot library does things really nice. It's when you're not used to it, it feels a little unintuitive, but then you realise that you can do like some really cool things with like

one variable or something and it's really neat. I really like it. Like the I don't know what stands for the aes thing where you can just tell it what your axes are, and then soon as you've told it what the axes are just throw our data at it, and it's like oh, that's nice"

-"I think matplotlib is- with, with Python you tend to like have one library, that does something really good, and matplotlib does it really, really good. And there are like two ways of doing things in matplotlib, like the object oriented way and then the just plot dot way. And I think the plot dot way, I find it a bit like un-pythonic, it's very different to how you write the rest of Python. Then the object orientated way, is making plots exactly like you write the rest of Python, which makes it really natural"

- "And with R- it's funny, like with R as well it's the opposite, like ggplot is very object oriented. whereas the rest of R is not."

Henry (Py)

- "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."

Nikoleta (Py)

- "Um, so the problem with R is that the type of plotting that I've personally used in R was more pre-recorded things, pre-recorded things, sorry. So, for example, I used the stat package and it would put the arrows and things like that just with a function. But that made me feel back then that in order to manipulate the plot further, it was a hassle for me because it was a recorded thing and then I had to go on top and force things."

- "Where with matplotlib when I plot, I do everything from scratch"

- "So how I gather when it comes to plotting both of the languages are quite powerful. And I know, for example, for plotting graphs R is fantastic where I haven't found something equivalent as good in Python. Plotting wise, both of the fantastic people will argue that one is better, but it has to do with bias, in my opinion. I would say matplotlib is fantastic, but I don't know R ggplot, so. Yes."

Owen (R)

- "Oh, I think it's a strength of R absolutely and the ggplot in particular, so I would say that's extremely well suited to visualisation. Well, ggplot is and again, talking about Python, I think, again, you have to specify what pack- or your library you're using for graphics."

- "Yeah, so so that's a sort of a python import of a matlab. Graphics. Language, isn't it, Google graphics and so. Yeah, I mean, Matlab certainly has good graphics as well. I don't have enough experience, I suppose, with Python graphics, to have a really informed opinion."

- "I think like most people, once you once you've learnt how to do graphics you are happy with, you tend to stick to the system you've learnt because learning a new system takes so long."

- "I mentioned Julia as a language, so I've learnt that fairly recently. And doing graphics in Julia, well, they've, they've imported matplotlib, but it's also fairly easy to use R to do plots for Julia output. And so that's what I do. And so instead of learning a new system."

Vince (Py)

- "Yeah, that's an interesting question. I mean. So. Just from the fact- essentially the reputation that ggplot holds, that's the immediate thing that stands out to me"

- "I'm like, you do visualisation with ggplot, you must really, you know, you must really know what you're doing. Because ggplot is kind of like the gold standard of of of visualisation. So it certainly seems appropriate, you know, that that that weird issue with the truncation. I would or I would wonder if that does mean maybe there's something else."

- "Because truncating some data is not, you know, that stands out to me. I'm like, oh, I don't know if there's a better way of doing it, but that would be a thing that I'd be like. I wonder if there's a better way of doing that."

- "And then kind of in a boring way, in a boring way on the Python side, yeah, that looks perfectly appropriate. That looks like good old boring matplotlib, you know, but boring in a good way, you know what I mean?"

- "One thing, you know, and this might almost be kind of like a slight bit of imposter syndrome because my matplotlib is so much better than my my ggplot plot, you know, matplotlib's the first time I go, I pick up to draw anything anything. Like there's a bit of me that's like, because it's ggplot it's probably better, you know, but but I don't know. I don't know. But that's all that's really just the reputation of ggplot. So you know, in Python.

  A lot of libraries, a lot of other visualisations, libraries come up and they kind of promised to be the ggplot of Python. Whereas in reality, matplotlib is perfectly fine. So those are my thoughts between the two of them. Sorry if that was a bit vague again."

Q5: for a beginner with equal experience in both languages, which library do you feel would be easier to learn, ggplot2 or matplotlib?

Andreas (R)

- "I'm biased towards R, um I think, I think it's, I don't know it's very simple that ggplot functions. It's you know, it's very simple to change the parameters and do whatever you want to. So at least, you know, I'm using it, I'm using it much more frequently, I use Python, you know, I'm not that familiar with Python because I use it a couple of times and just very simple programming exercise and have to get a feeling of it. So. Yeah, I'm definitely going to go with R."

Geraint (Py)

- "Probably matplotlib. I'm only saying that because I think the way matplotlib some of the syntax of it is very similar to the rest of Python. Like, dot bar is a function which takes two things during they're similar to how you use the rest of Python, whereas with ggplot, even though that aes thing is really nice, neat adding things to it, it's not like anything else you've seen in R, so it might be really hard to get."

Henry (Py)

- "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."

- "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."

Nikoleta (Py)

- "Yes, I am biased as in I find- I find Python to be more readable, also, as a person whose English is not the first language, I find Python to be more readable. So for any beginner starting something, I would say that Python, in my opinion, is slightly easier. Now, I'll say yeah plotting with the same."

- "Yes, because I think with matplotlib it's very clear, right? You have the plt dot or figure axis as each command and you know what you're

  doing where in R you have these weird plusses and, weird. So, yes, I think, like, oh, are we adding things to the things. Yes. Is not as clear to me."

- "And I guess you can't necessarily see- so the way that I'm set up, so right now I have bar plot 1 in both languages in front of me. And Fantastic, like, I understand that geom_bar is under ggplot, so it has to be part of ggplot, but apart from the fact that it's underneath it,

there's no other indication that it's part of ggplot plot where with the Python code, I can see that is part of the plt library. You know, no library, but you know what I mean. Everything's after that plotting instance, and I'm like ah, OK, well."

Owen (R)

- "Ooh, I think actually probably ggplot. Was the. And again, as long as you're not trying to do anything too different from this sort of standard set of plots that it caters to. Which is fairly rich, I think, and then it's got, I think, a nice, logical system for for doing things."

- "As soon as you want to do something a bit out of the ordinary, then life becomes much more difficult and my suspicion, not one that I've really tested, is that making ggplot do something a bit different is going to be harder than making matplotlib do something a bit different."

- "So for a beginner, that shouldn't be an issue. And I suppose I mean, another thing for a beginner, something that's very important, is getting a feel for what good plots look like and what's possible as well. And I think ggplot makes that quite easy. Well, relatively easy."

Vince (Py)

- "So it's hard for me to answer that question from the code you're showing me.

  OK, well, let me try to answer that question from the code you showed me. But I think what I'm trying to say is I'm not sure I'm answering the code from the code, from the question and the code you're showing me.

  I think I'm answering it from my own knowledge. OK, so looking at the code you're showing me, you're seeing, we're seeing these additions. We're seeing these these specific grammatical calls, the ggplot library of scale, like continuous scale, y discrete, scale_x_discrete, et cetera, et cetera.

  Whereas the code on the left, the Python code, seems to be: create a bar plot, create an x label, create a y label and possibly modify it, modify the text. So I think I think the matplotlib plot code you're using is probably the more straightforward one for a beginner to pick up if they just need to draw something like that."

- "But you are using there, you are using there the pyplot interface to matplotlib, which is kind of-there's actually, there's two ways of plotting in matplotlib, using matplotlib directly or using pyplot and pyplot is actually just meant to be matplotlib's interface to make it like Matlab's plotter. So it is meant to be relatively straightforward.

  So I think in a way whereas to plot with ggplot you essentially have to learn the grammar of graphics, right? That's kind of what ggplot is all based on, is the grammar of graphics.

  So I think my answer is twofold. If you just need to get a bar plot or

a histogram or something like that, I think that matplotlib's pyplot is probably what I'd recommend to someone with equal knowledge of both.

If you want to become very good at visualisation, then I think my advice is to learn, but not because it's easiest, but because by learning it, you'll not only learn syntax for visualisation, but you'll learn the graphics of visualisation as well. So I think, I think that's my answer."

- "Python being more- the problem with matplotlib, right, is because it's got those two interfaces to matplotlib, which by the way, Python prides itself on not having. Python prides itself as being a language where there's only one way to do things.

  And it's unapologetically like, no, no, no, there's no confusion. There's one way to do something, OK? And that is indeed wonderful about the language. But matplotlib is a wonderful example where that doesn't apply because it's this pyplot and there's matplotlib.

  And and in a way, having those two options is both nice because you've got that like gateway drug, which is pipeline at the same time as someone who really, really only ever uses pyplot. The few times now where I have to use matplotlib, it's almost harder.

  Whereas if you just kind of say learn ggplot. You know, then you- Whilst it's more user friendly, I'm not sure it's more user helpful, if that makes sense, I think in the grand scheme of things. Yes, learning equips you equips you better than anything else."

Q6 : Which package or library do you feel gives the most publication ready output?

Andreas (R)

Control Response - R

- "I like I like the the the fact that you don't box it, essentially."

Log Response - R

- "I don't I don't think the logarithm it helps in any way. So in that case, I'm I'm still going to start with this one here, because it doesn't it doesn't look like the the logarithm helps."

Geraint (Py)

Control response - R

- " But I think but I do think that's just opinion. I don't think it."

- "I don't know what publications would like more but that's what I like more."

Log response - Py

- "I rather the one where it's got 10 to the 0, 10 to the 1. I think that's a lot clearer about what's going on there. I suppose with the other one, well no it is, it is the actual values, it just feels really random until you- unless you knew it was a log scale, you just you've got to really look to see."

Henry (Py)

Control response - Py
- "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."

- "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."

Log response - Py
- "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."

- "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."

Nikoleta (Py)

Control response - Py
- "OK, I think a lot of people would argue that the plot on the left looks more beautiful, OK? I personally argue the I would- for my publication I would probably use the plot on the right just because the bar plots are not elevated. They're floating like they're floating on the left.

  I think if the left plot they were not floating and zero started from zero, I would probably go for the left because I think it looks more appealing."

Log response - R
- "OK, then the plot to the left. I think the plot to the right at least needed some sort of explanation, that is the logarithmic of, so yeah.

So if these the standard that it returns I think you know for purpose of corrections, I think the left one on these in this case."

- "But the plots are still floating. And now they start from one right before they start from zero, and if you put those two plots next to each other, it would have been so confusing."

Owen (R)

Control response - no preference
- "I have no preference. I feel like I'm at the optometrist, so I think these are both equally good."

Log response - Py
- "Oh, interesting, okay, well, the matplotlib on the right. Yes, so that certainly makes it clearer that it's using a log scale, which I think is a good thing. That would be my preference out of those two."

Vince (Py)

Control response - R (marginally)
- "Yeah, yeah, I, I, I mean, I recognise which ones which, but I can't say. I suppose the one on the right is slightly nicer that it doesn't have the box around it. OK, so out of the box ggplot is kind of slightly more publication ready, but I don't feel there's a big difference between the two there."

Q7: How much freedom do you feel each language or package allows for customisation of plots?

Andreas (R)

- "I think I think both of them allow the freedom to do things, is that I'm just feeling more comfortable in R essentially to this than in Python because I did it much more times. It's not that I feel R, it gives you more freedom, I do believe, as I said, if you if you go to the standard package, you're not maybe you don't have as much freedom as you have in the ggplot package or it's not as easy to change things as it is in the ggplot package. But given that you are in the ggplot package, I think you have as much freedom as you want to do, to do things. And and I do believe in Python you do have the same freedom, so it's not- In terms of freedom, you can do it in both languages. The- the most, let's say, challenging thing is probably, knowing how to do it."

Geraint (Py)

- "I, yeah, I, I think they both probably do just as much customisation, but I would find customising things in matplotlib a lot easier. I feel like, even if you don't know what's going on, you know how to change things because you know how to access the ticks or whatever whereas with ggplot you've got to have this new object on and then you've got to go look something up."

Nikoleta (Py)

- "A lot, they both do open source languages. Yes, R is open source language, so I think a lot of course, I guess you would need a more- if you want to use the features that are there, fine, you can do it as a user. But if you were meant to implement something, maybe you would need a bit to be more comfortable with coding. So maybe a beginner wouldn't be there. But given that you know, you have some knowledge, I think it's very easy for both languages."

Owen (R)

- "Well, I think I've probably already answered that I think matplotlib allows more freedom. Yeah, well, for the the less experienced user, I suppose that's the caveat."

- "Oh. Yes, I suppose one- Something that- an advantage that Matlab has with its graphics that Python and R don't have is interactive graphics. Yes, so matlab, you can you can you can actively tweak it and see the changes in front of you as opposed to you know going back and rewriting your code. And that that certainly has some advantages that can speed up production of graphs. But it's- It's not a deal breaker, I suppose, but. Yeah, I suppose that's the advantage of a commercial package over a free package."

Q8: Other useful comments

Geraint (Py)

- "Did you write those with Vince or did you write them yourself?"

- "Vince loves a function, whereas I like functions for most things, plotting is one of the ones that I just don't see why we do it functions whereas Vince really likes putting them in functions, so it's just interesting that you used functions."

Vince (Py)

- "I like that you've put your plotting code within- um in a functional way so within functions."