Brief Survey on Analysis Formats and Tools

Annoucement To prepare for a discussion with the ROOT team, we have prepared a brief survey on I/O formats and analysis tools used within ePIC:

https://docs.google.com/forms/d/e/1FAIpQLSffmr7S2qGDECnp6_SjDEF2cd30Vgey4IxY0kv90CIlQ5rN0w/viewform?usp=sharing&ouid=117756119943977751147

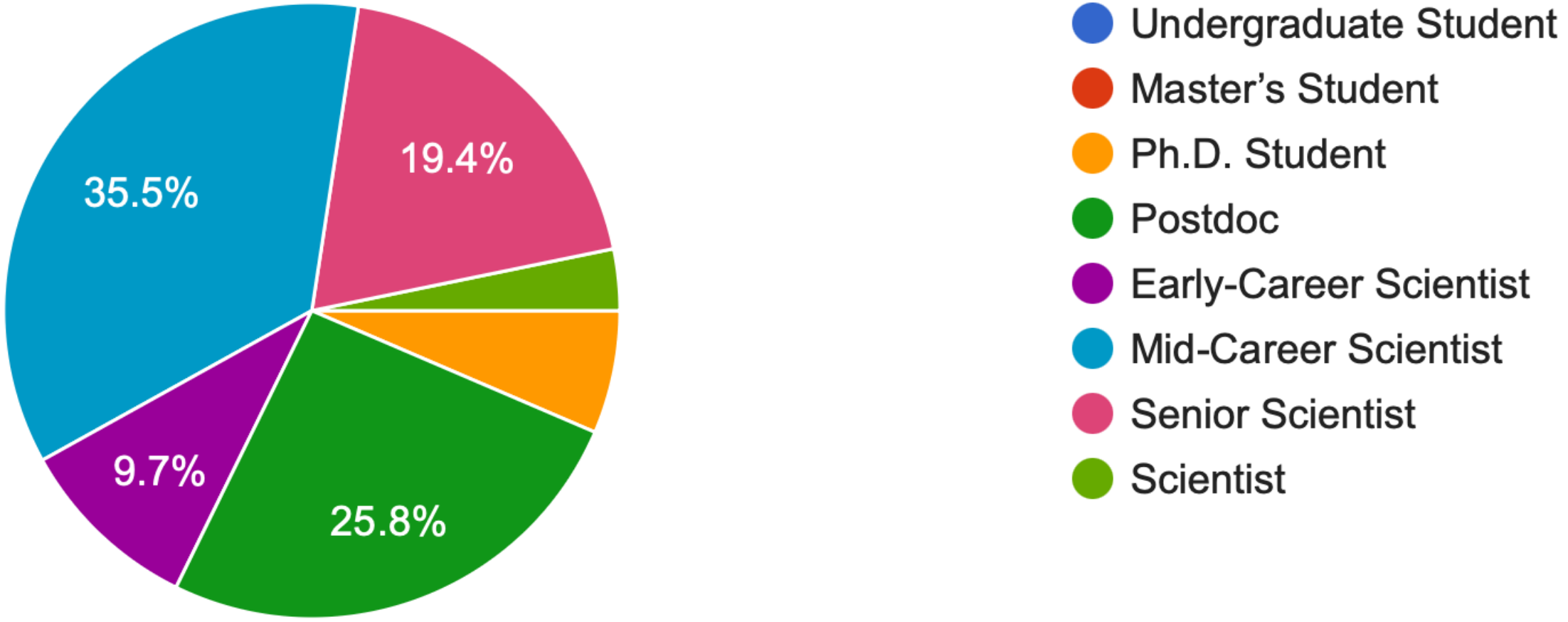
The survey has seven questions and should take only a few minutes to complete. Your responses will help us understand current usage and provide useful feedback on how to take advantage of developments in ROOT I/O and analysis tools to better support ePIC workflows.

This survey is limited in scope and is intended to support the upcoming discussion with the ROOT team. It will also help prepare for a broader discussion of analysis models, tools, and workflows within ePIC.

June 10: 31 Reponses

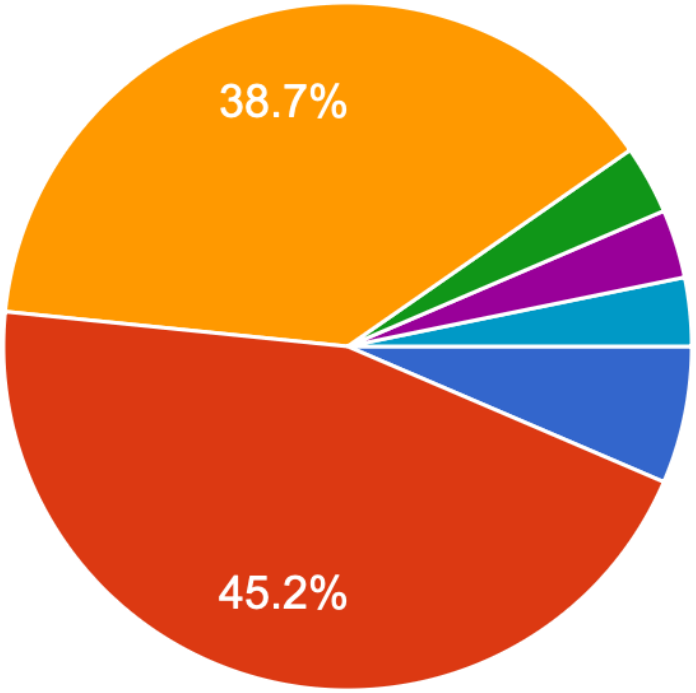
Question 1

What is your current role in ePIC?



Question 2

For your detector or physics studies, what do you mainly use for analysis?



- Python
- ROOT
- Python + ROOT, including PyROOT
- C++ and ROOT
- I mostly use C++, ROOT and also python.
- C++ and root

Optional Question 3

Are there limitations in the ROOT TTree output from ePIC simulation productions, or in the ROOT interfaces commonly used to analyze it, that affect your studies or workflows? Please list any usability, performance, or workflow issues you have encountered, e.g., with TTreeReaderArray, TTree::Draw, or related tools.

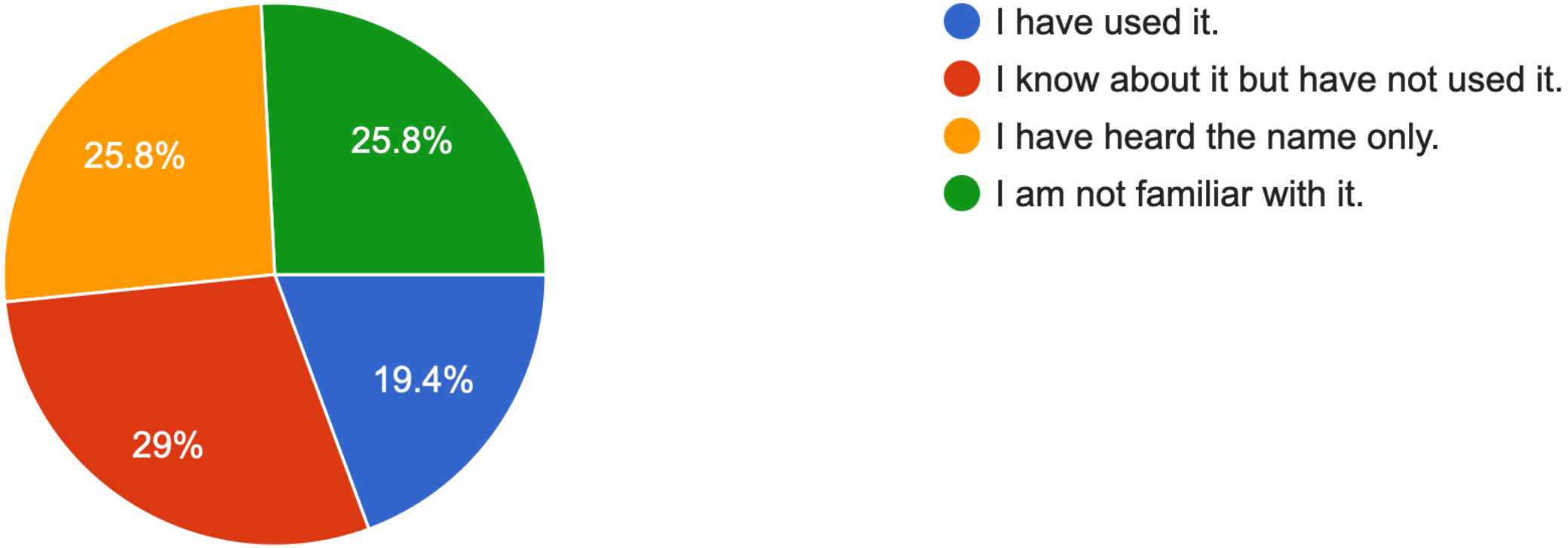
1) Yes
2) – 6) no. Has been fine so far. None that I've encountered so far. So far no! none.

7) One particular issue I encountered is that when trying to access the calorimeter information associated with a ReconstructedParticle, it is not easy to figure out which calorimeter the information comes from.
8) Absence of descriptions for the podio collection branches that can be easily consulted inside the TTree, with e.g. originating algorithm and ancestor podio collection branches.
9) less of of a TTree issue, but lack of Branch/Leaf variable documentation. With >3k variables in the TTree it is difficult to find the relevant ones and their relations. E.G RecoParticle->RecoTracks--> how many hits were used in tracking.
10) Edm4Eic is strictly version dependent. It just crashes if any software version mismatch. No easy way to check what stack versions to read older data

11) i wanted to use plotting with ROOT while working with uproot - as of now, i use matplotlib RDataFrames Multithreading is limiting plotting Is there a way to convert between those methods? For RDataFrame speed and easy event loop code understanding. Make TTreeReaderArray to read types from header before compilation? PyRoot also require declaration for C-type variables beforehand c_int and this is awkward in python .
12) Not usually. I find that ROOT TTree output is fairly reliable. For analyzing ROOT outputs, I also tend to use uproot+awkward.
13) No limitations, but I find TTreeReaderArray not very elegant.
14) Using TTreeReaders makes using the provided associations somewhat cumbersome. I've since moved to using edm4eic types to make this easier.

Question 4

RNTuple is ROOT's modern I/O format for columnar event data. It is designed as a successor to TTree and aims to improve performance, scalability, and usability for future analysis workflows. How familiar are you with RNTuple?



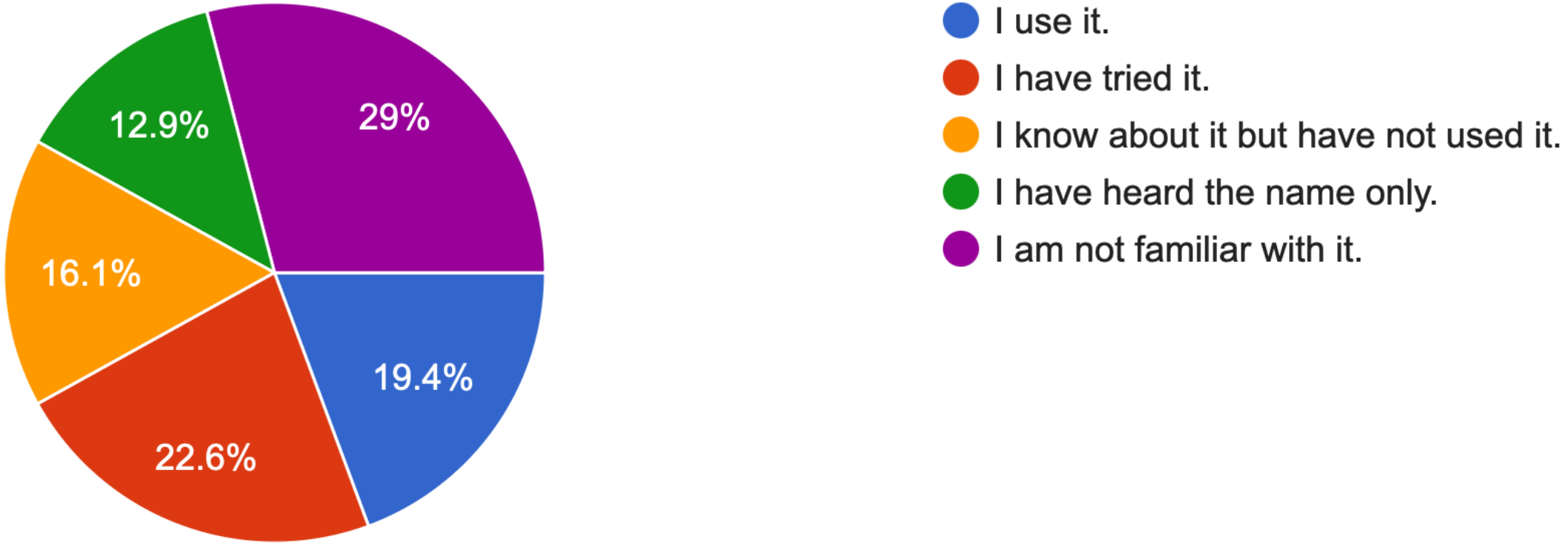
Optional Question 5

Are there modern analysis techniques and workflows you would be interested in learning more about?

1) This is a vague answer, but pretty much everything new is interesting to me.
2) Yes: open to anything eficient and clean.
3) I use a lot of Python for analysis. I'd be interested in learning more about dask and if it's useful for ePIC workflows.
4) Distributed analysis with dask in ROOT (familiar already with the python-only approach with uproot).
5) This might be out-of-scope for the meeting, but I know that there were recent developments in using PODIO with RDataFrame. I haven't been able to work through it myself yet, so I'd be interested in there's a chance to have a tutorial/talk on that!
6) ROOT7 adjustments
7) I want to learn more about using AI for coding/data analysis.
8) neural network techniques
9) Duckdb and other modern data processing tools. E.g. DuckDB can be used for our files to process data with sql queries. OLAP is extremely performant.

Question 6

RDataFrame is ROOT's modern interface for describing analysis workflows. Users define selections, derived quantities, histograms, and outputs in a declarative way, using either Python or C++. How familiar are you with RDataFrame?



Optional Question 7

If you mentor students, what feedback have you received from them about analysis workflows, techniques, or output formats?

1) – 4) n/a. None so far. nothing in particular that I haven't already known. Nothing specific. They seem happy using what there is.

5) Many things are not documented so they are facing challenges
6) Python is bread and butter
7) Students use python in web interfaces almost exclusively.
8) Over time, there is less interest in using ROOT, and more in using PYTHON.

9) I have received very mixed framework. I am mostly working with Bachelor and Masters' students in Italy. They have a wide variety of background in C++/python. Their personal backgrounds often decide their opinion on the data format. However, over time they get used to it and work comfortably. The initial training can be time consuming and tedious.

10) Many of them are not familiar with object orientated programming so ROOT is often a stumbling block for them. However, they then use python and progress to a point where they hit a corner case where python *does* care about a type which stumps them completely. Usually once they actually try to engage with ROOT and use it, they have a better understanding of their analysis. My concern with more columnar analysis approaches is that they often obfuscate what is actually happening in an analysis. E.g. the process of selection cuts. With a TTreeReader approach, this is much more obvious generally - keep events which match a condition, drop those that don't as you look event by event for example. This is typically more intuitive than - "Apply this bunch of conditions, which create huge boolean arrays as to whether values match a condition, and smash 10 of them together at once.".
11) Functional programming is a whole new paradigm to study

12) We have developed an RDataframe based analysis framework for root, which ECR are using and benefiting from developing