

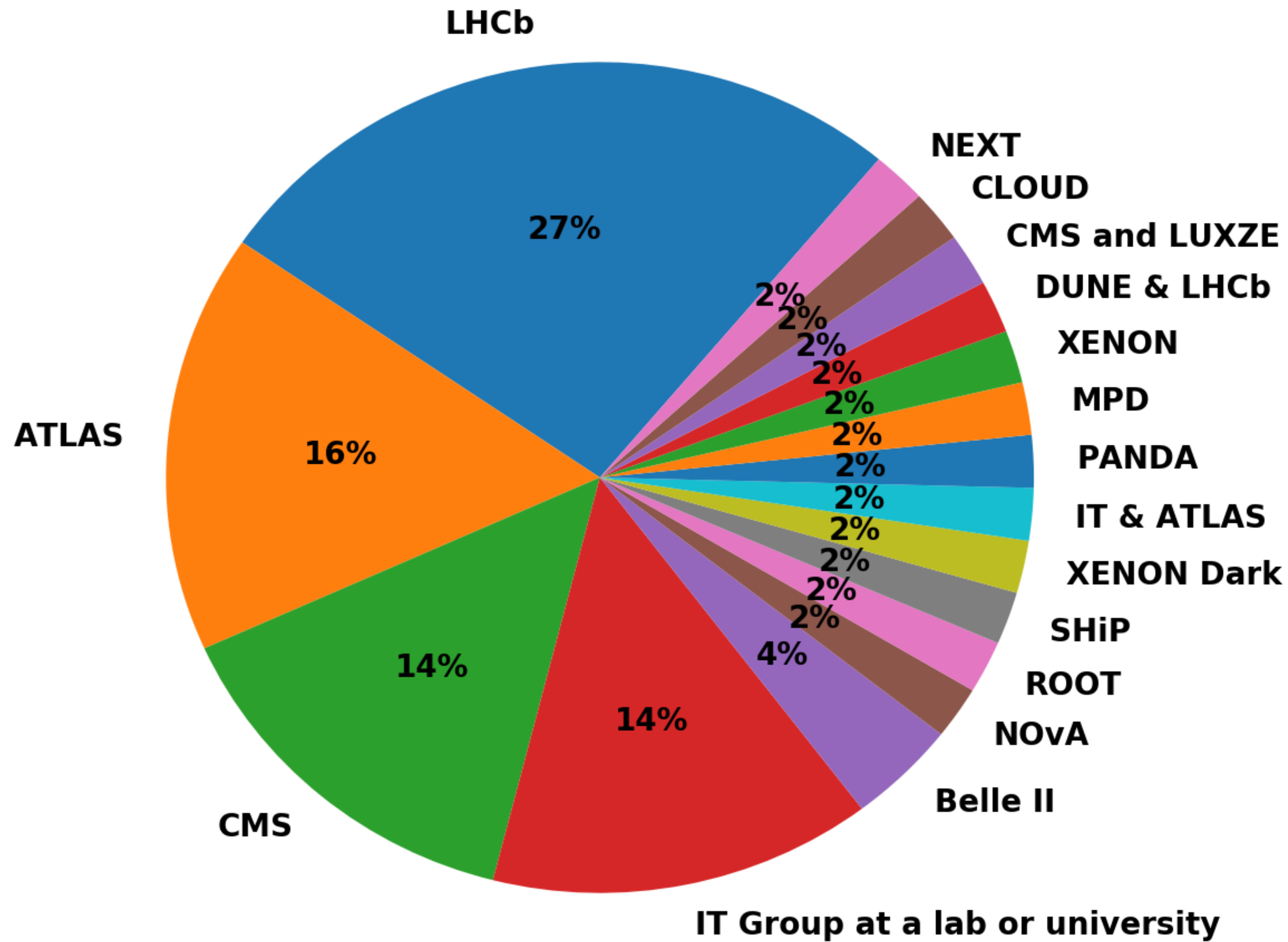
# PyHEP - Questionnaire and Discussion

Graeme Stewart, EP-SFT

# Questionnaire

- Many thanks to all who gave us input on the pre-workshop questionnaire
  - 50 people responded (of 70 registered, i.e. 71%)
- What was the idea?
  - To understand the background, interests and concerns of those coming to the workshop
  - We hoped it would guide the topics we address
    - We think it has validated what we put on the agenda
  - And could stimulate some discussion...

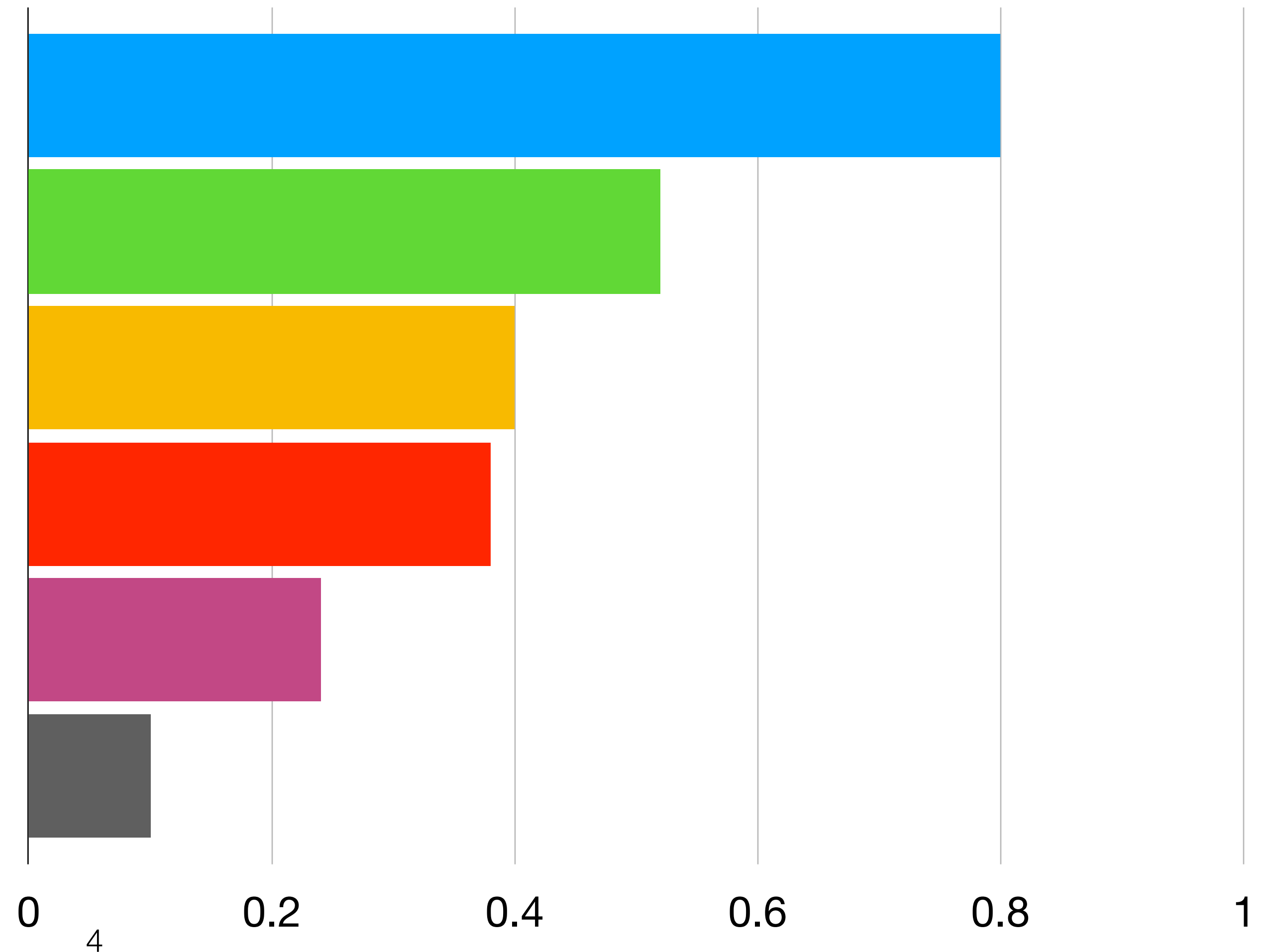
# Who are we?



- Healthy distribution across other LHC experiments, neutrinos, nuclear physics
- More IT people than we expected - welcome!
- No ALICE
- (I guess they have still to see the light :-)

# Motivation

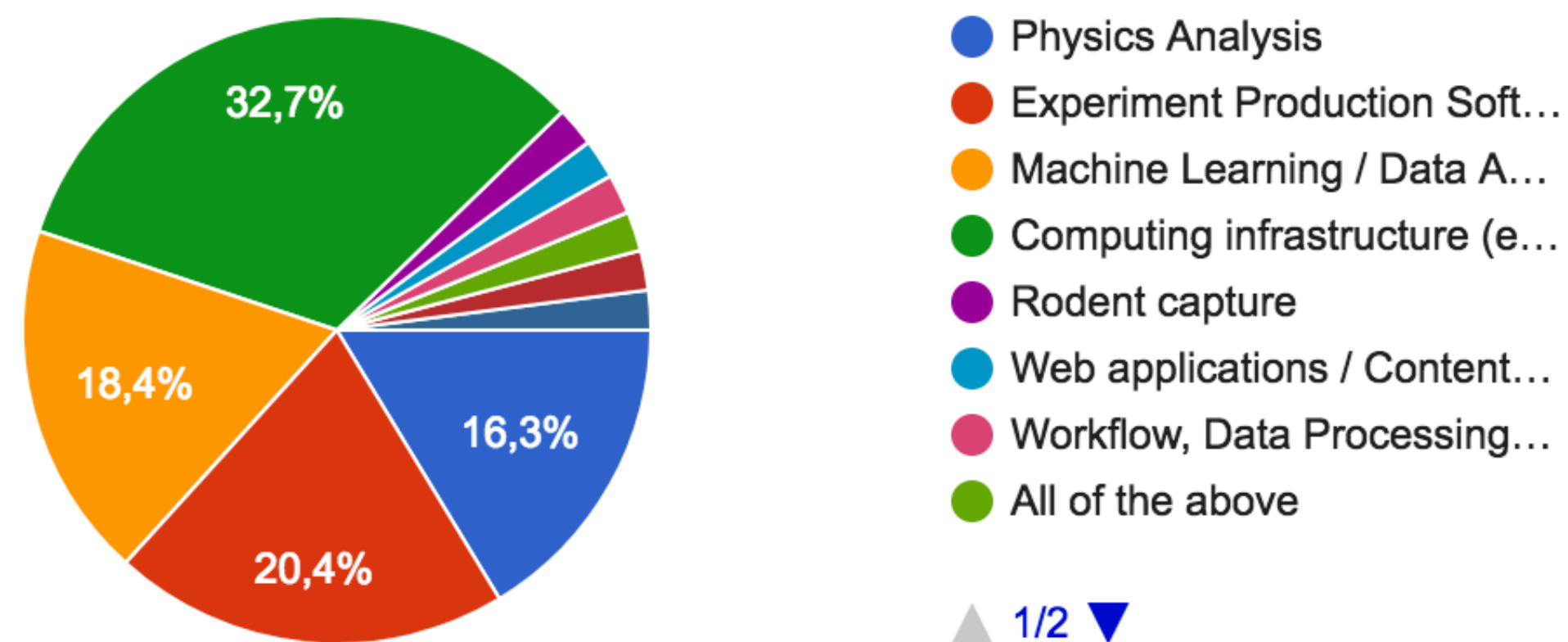
Motivation	Important to	%age
Development speed and efficiency	40	80%
Availability of other software packages	26	52%
Interface language	20	40%
Machine learning packages	19	38%
I just like it	12	24%
They make me do it...	5	10%



# What are we using Python for?

How would you characterise your principle use of Python?

49 réponses



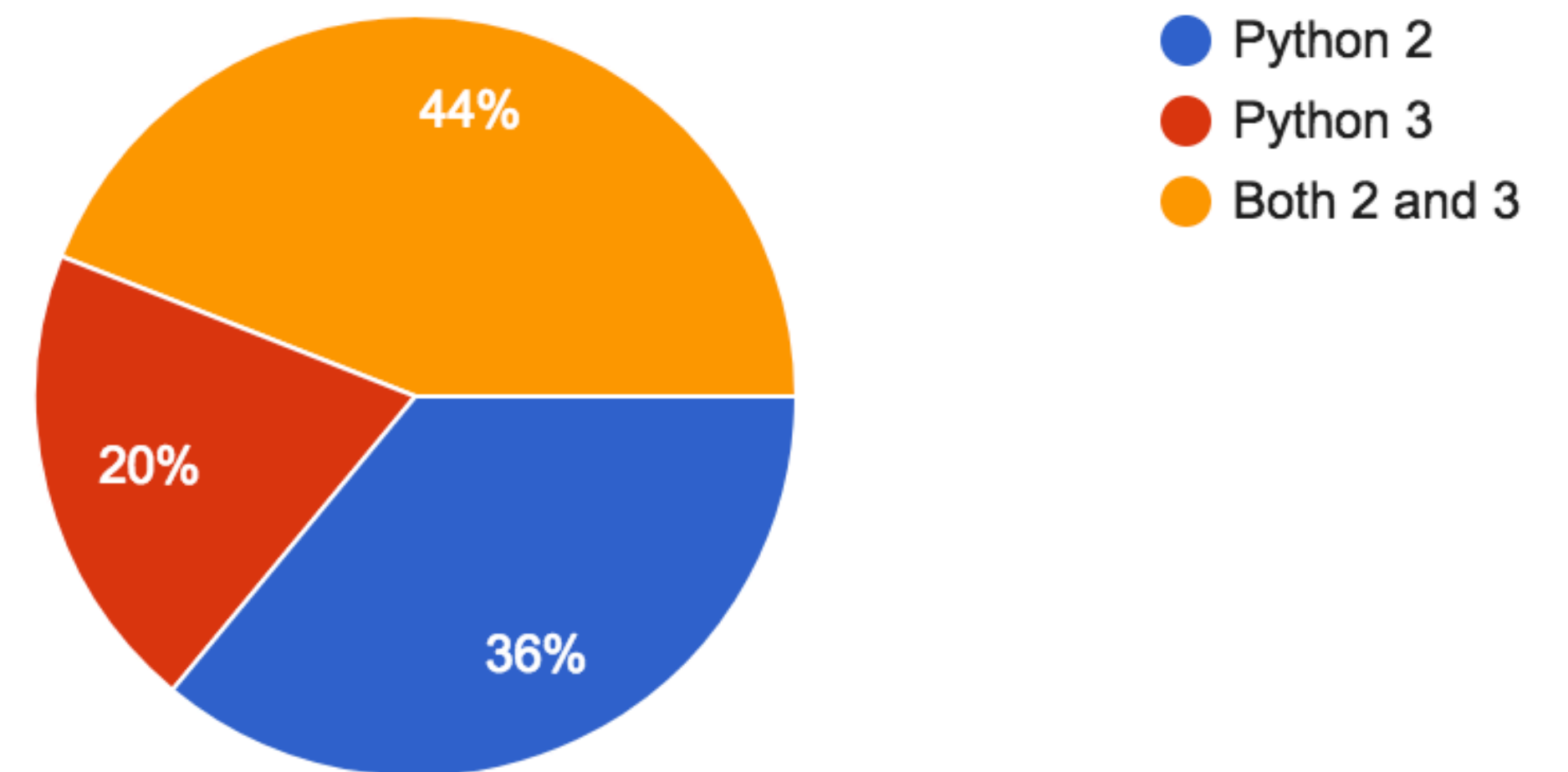
- Lots of physics (no surprise)
- ML, Experiment Production, Physics Analysis
- More infrastructure use than we expected
- But plays to Python's strengths, of course

# 2, 3, 2.5?

- Amongst us there is a very healthy use of Python 3
  - Both 2 and 3 we interpret as “3 when I can, 2 if I have to”
  - Migration to Python 3 is a big concern for the community as we’ll see later

Which major version(s) of Python do you use?

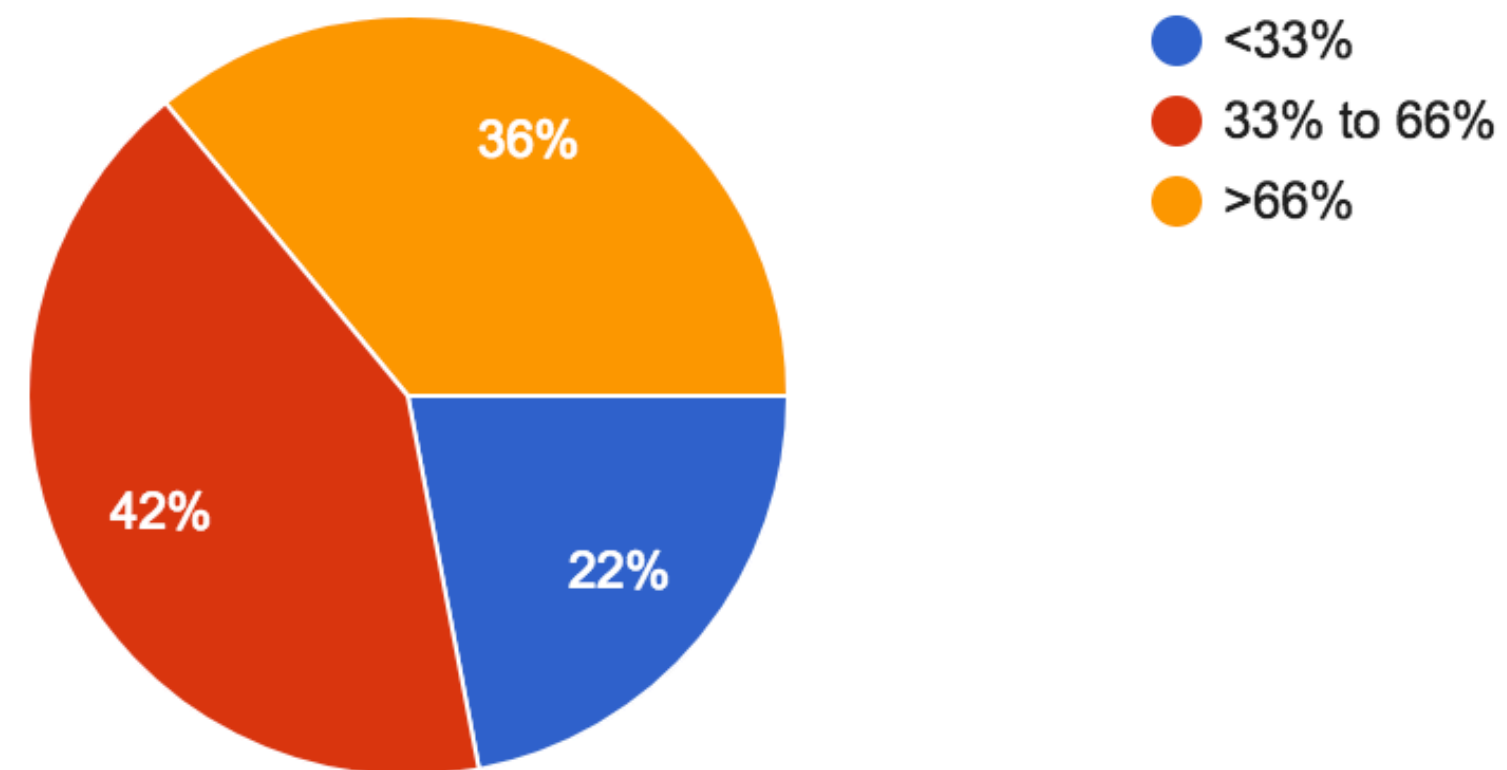
50 réponses



# Use and Evolution

What fraction of your programming is done in Python?

50 réponses

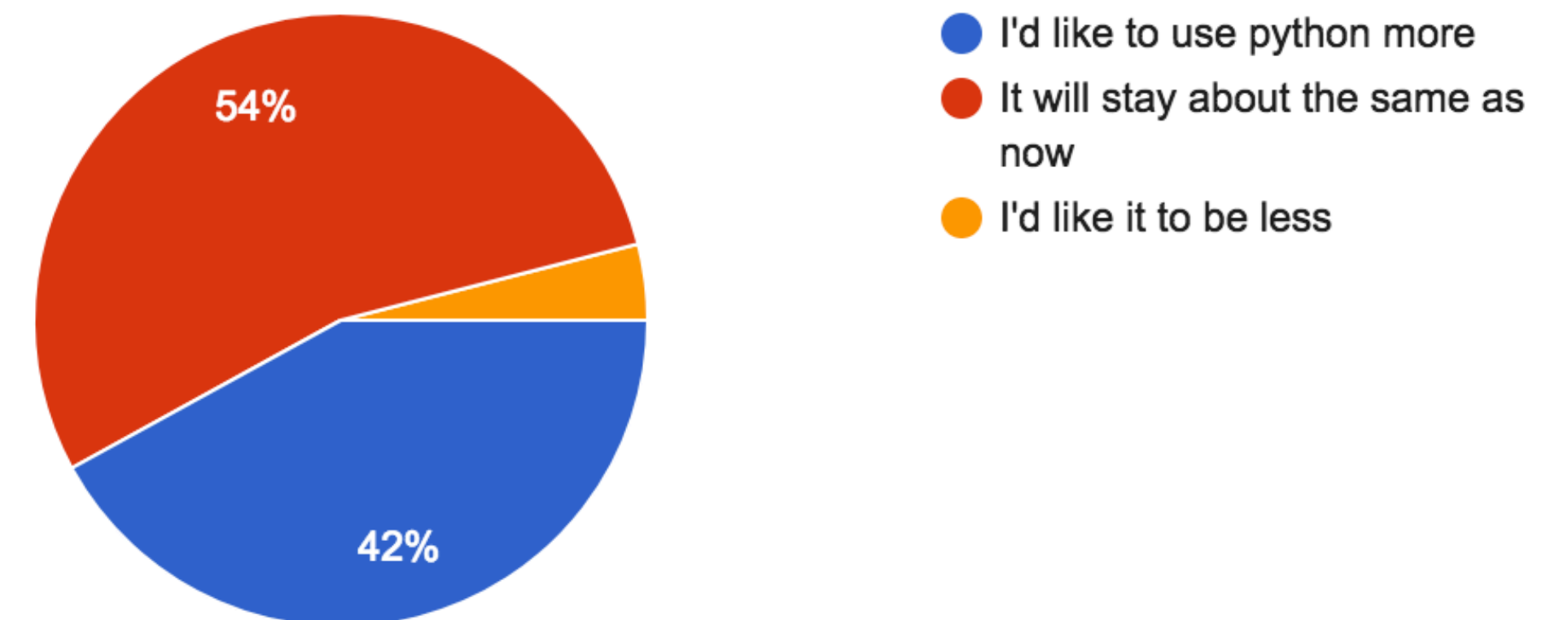


- We do use Python a lot (selection bias!)

In the future, how would you like your use of Python to evolve (as a fraction of your programming time)?

50 réponses

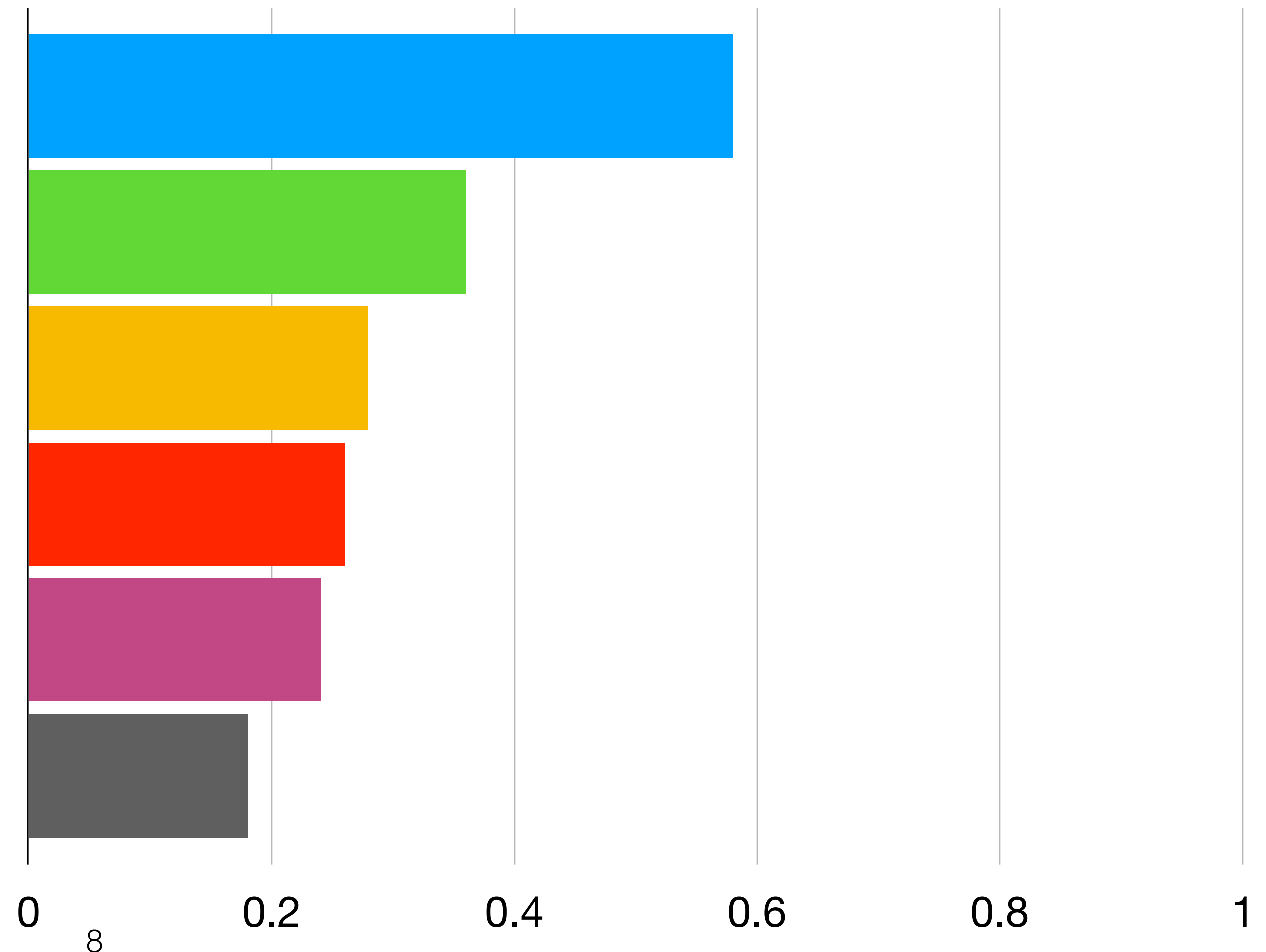
- Anticipating that will stay the same or increase





# HEP Investments for the future?

Important Development / Investment for HEP	Important to	%age
Migration from Python 2 to 3	29	58%
Better ROOT integration	18	36%
Better experiment software integration	14	28%
Better development tooling	13	26%
Improved training	12	24%
Runtime speed improvements	9	18%

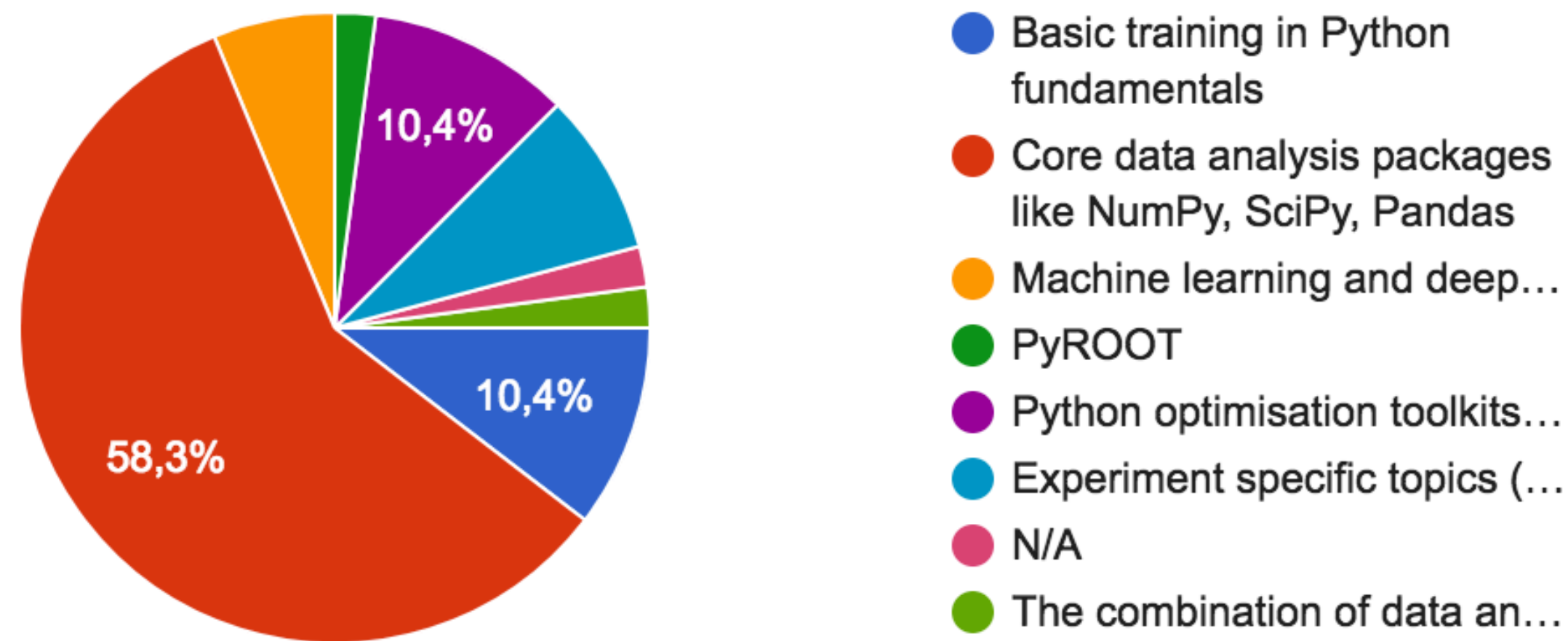




# Training Needs

For training, what do you think is the most urgent training to develop and give in HEP?

48 réponses

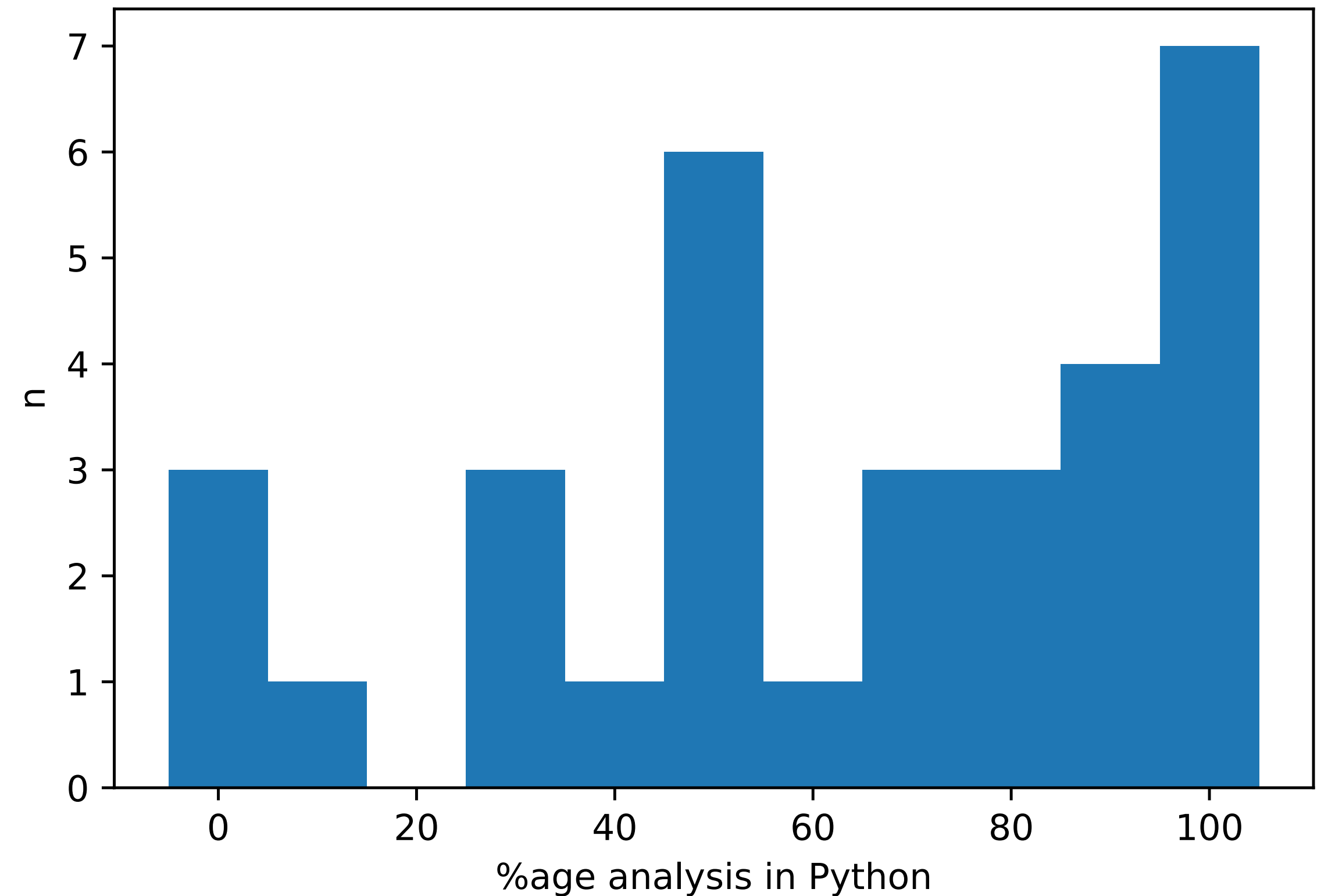


- Very strong interest in Python data analysis ecosystem
- Open question is what is the role of HEP in this sort of training?
- In particular what's the boundary between HEP specific topics and generic ones?

# PyAnalysis

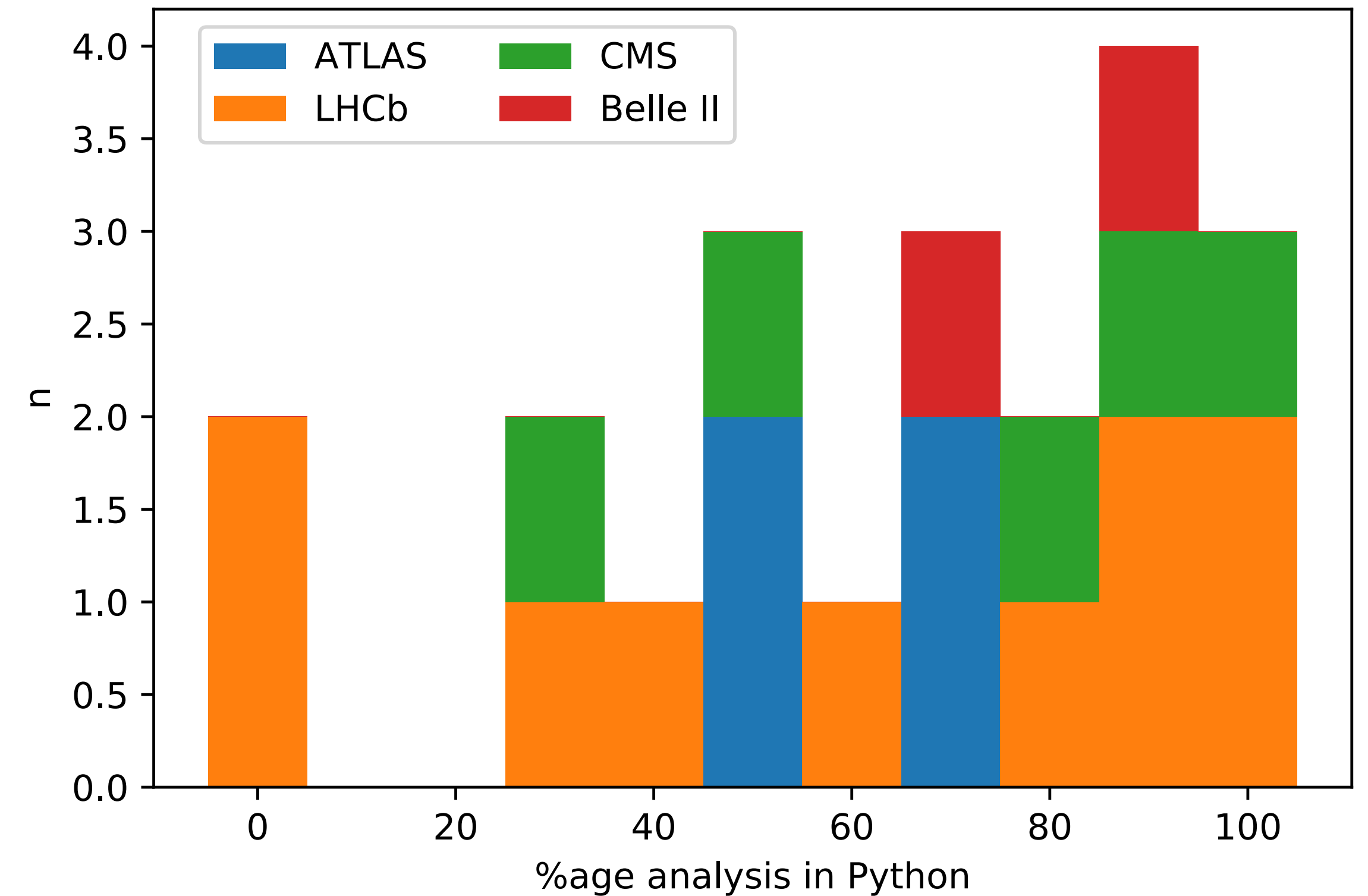
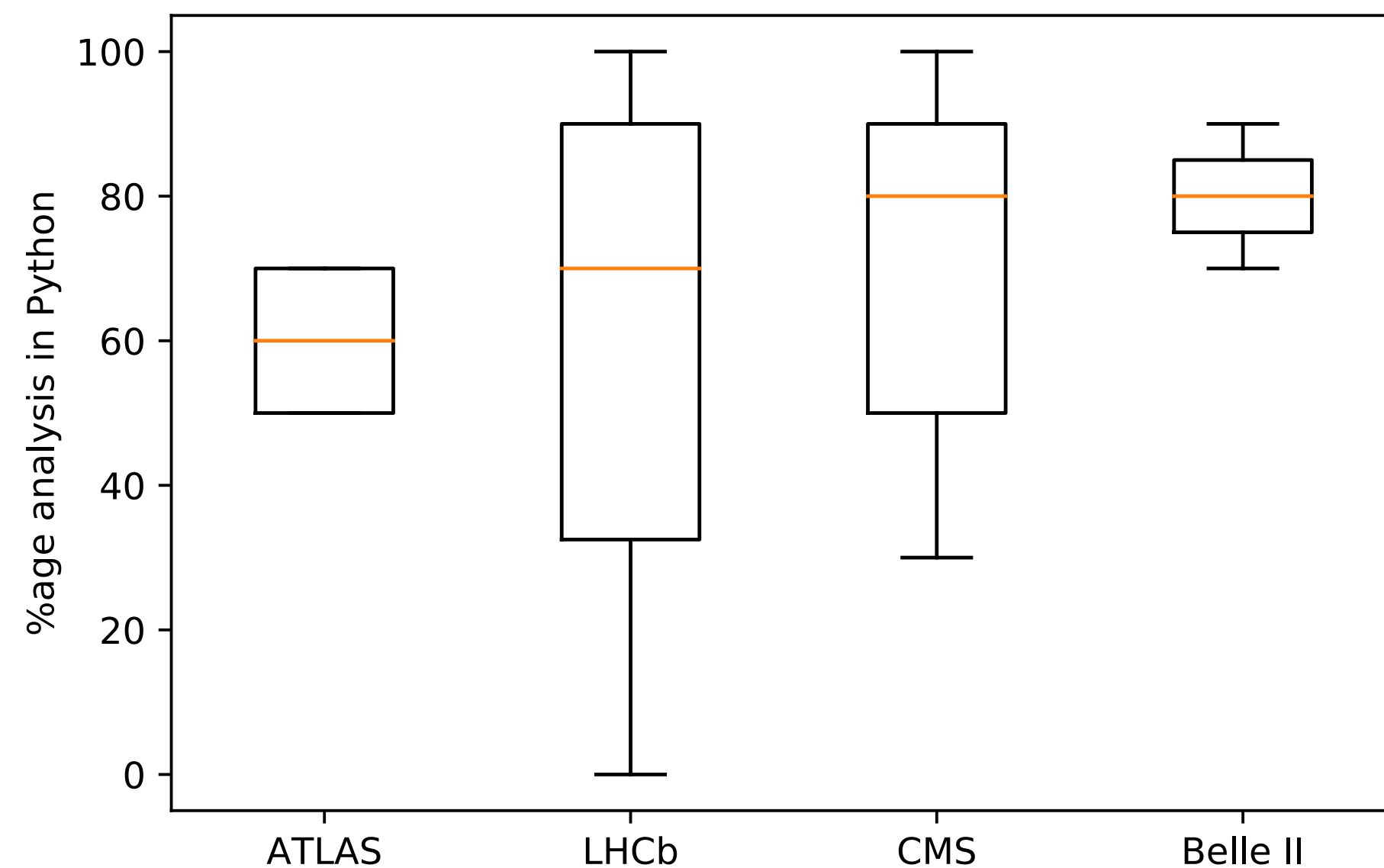
If you do physics analysis, to what extent (%) do you use Python, instead of compiled/interpreted C++, in your analysis work? 32 Responses

- Significant use of python for a significant fraction of analysis



# PyAnalysis by Experiment

- Surprisingly little correlation with experiment
  - Statistics are not large, *caveat emptor*



# What change or new development in the Python/ROOT ecosystem would have the greatest impact on your use of Python?

- A more appropriate **training**.
- **Easier installation** of ROOT (pip, conda, etc.)
- **Seamless integration between ROOT and the "Anaconda family" of tools** on all common platforms (Linux, Mac). In fact, could ROOT become part of the Anaconda releases?
- Better production frameworks
- Moving more code to **Python3**
- improved **ROOT** data (container) **interoperability with NumPy ecosystem**
- **Reduction of weird memory problems** and side effects (ROOT vs Garbage collector, gDirectory ...)
- Better ways to run with multiple pythons (better modular PyROOT or switching to uproot)
- [Data] **interchangeability** of pandas, roodataset, ttree, numpy array
- Using **Jupyter to disseminate analysis code** instead of just results, yielding in more transparency.
- To have a **pythonic** way of using **ROOT/RooFit** etc... and a user friendly one.
- **Tool agnostic file format**
- Ability to make plots with weighted events / bins and ROOT-like COLZ plots in matplotlib / seaborn, etc
- **Breaking of dependencies** on ROOT libraries and core software from experiments.

# Other Points of Interest

- **"Analysis on demand" via IPython notebooks**, e.g. SWAN, seems to be an exciting development in our field, and one which Python is very well suited to (although SWAN is not a Python-only platform). This may be something that we'd like to discuss at the meeting.
- Most major packages have announced end of support for Python 2 (Jupyter, Numpy, Matplotlib, and Pandas, to name a few), so I think preparing for the **transition** is important for most experiments (a few are already on **Python 3**). Focus on how to avoid installing ROOT for every python interpreter or virtual env would be interesting (currently uproot, histbook, etc).
- I'm very interested in **Python2 to Python3 migration**, as manpower for the migration in my team is very limited (boils down to only me, part time). I would like to learn from experience of others, and perhaps collaborate. Thank you!
- How could Jupyter and Python be used in HEP to **improve the coding education** and provide a smoother learning curve for students?