HSF – Python and PyHEP

Eduardo Rodrigues, for the HSF PyHEP WG
University of Liverpool

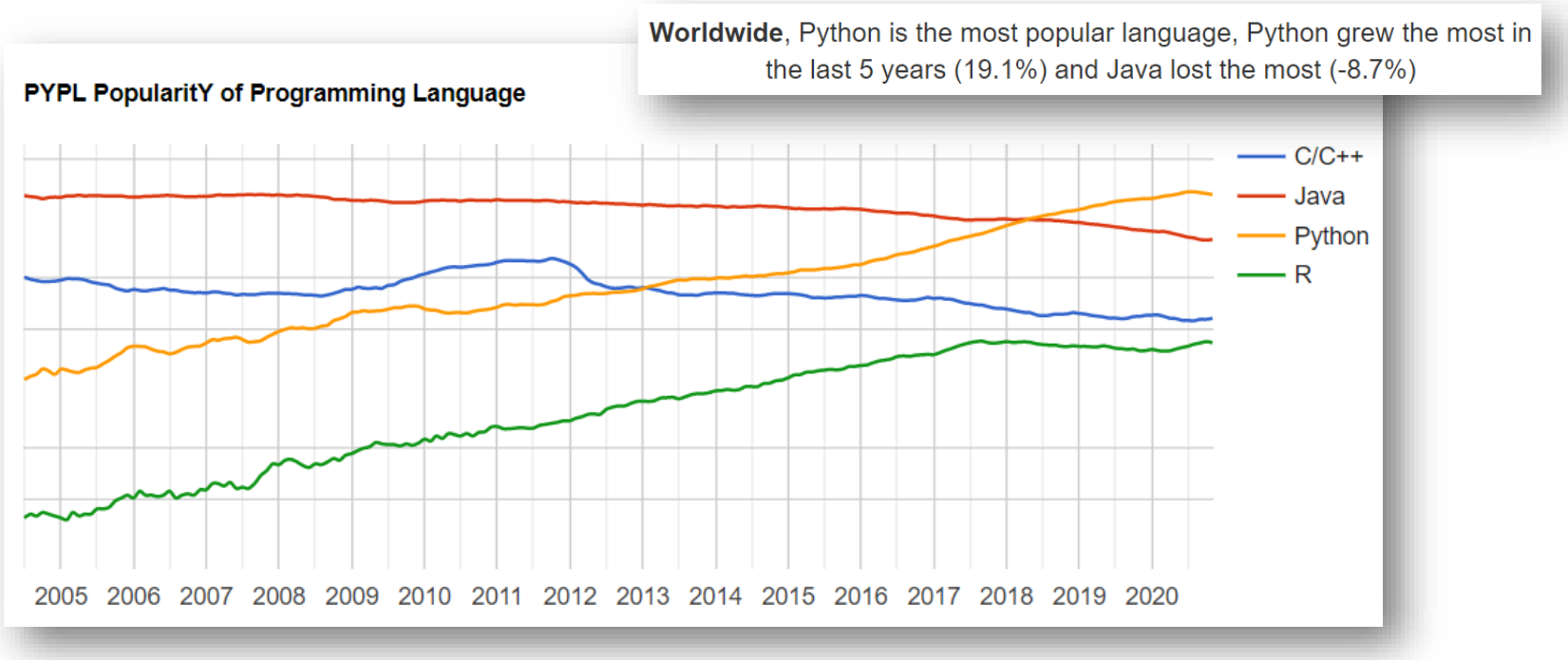
Software & Computing Round Table, 12th January 2021

Python in HEP

- I.e., Python in Particle Physics
  - "HEP" is a recognisable acronym

Python (in HEP), you say?

- PopularitY of Programming Languages (PYPL) – Python is the big winner!
- Popularity based on how often language tutorials are searched for
  - Data from Google Trends
  - Log scale!
- Same conclusion for popularity of languages for ML



[Taken from http://pypl.github.io/PYPL.html]

Why?

- Very large software ecosystem built atop NumPy and SciPy
- With very large and active community
- In general, excellent documentation and community support
- ...

Let's roll back a few years … at the time of the Community White Paper

The HEP Software Foundation

A Roadmap for HEP Software and Computing R&D for the 2020s

Computing and Software for Big Science (2019) 3:7
https://doi.org/10.1007/s41781-018-0018-8

- **Python had already been identified as a first-class language for Particle Physics back in 2016-17:**

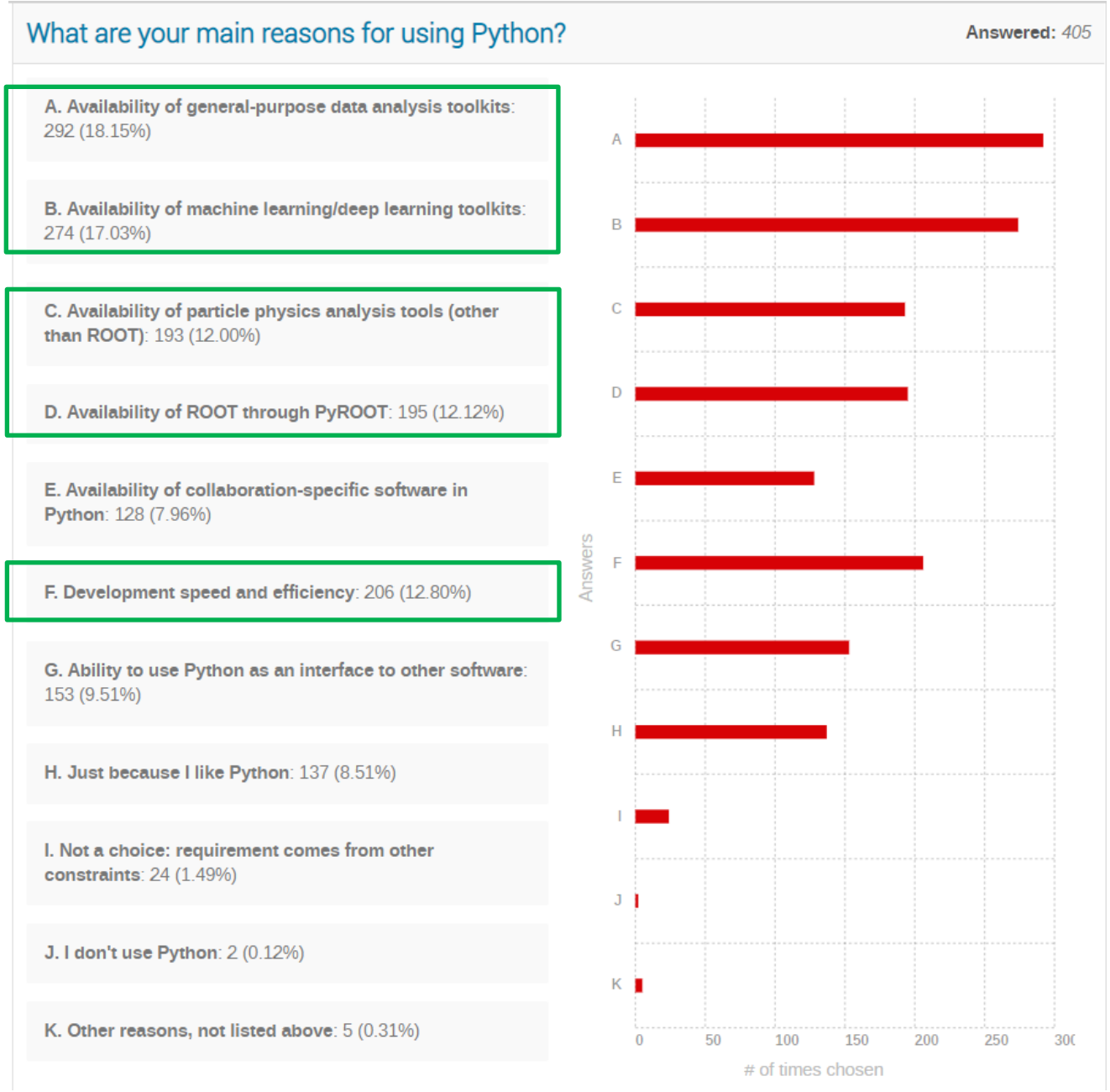
> "Python has emerged as the language of choice in the data science community, and its use continues to grow within HEP. … Python could reduce the complexity of analysis code, and therefore contribute to decreasing the "time to insight" for HEP analyses, as well as increasing their sustainability. Increased HEP investment is needed to allow Python to become a first-class supported language."

- **A lot happened in the meantime**
  - **Evolution of (Py)ROOT**
  - **Community-wise**

> "– Expand support of Python in our ecosystem with a strategy for ensuring long-term maintenance and sustainability. In particular in ROOT, the current Python bindings should evolve to reach the ease of use of native Python modules.."

- **What follows exemplifies the evolution and what helped shape that evolution …**

Why do particle physicists use Python ?



Taken from the PyHEP 2020 pre-workshop survey (408 respondents)

Python adoption in HEP – example of a CMS study

Direct method: look at their code!

GitHub API lets us query users and repositories (URL → JSON).

Can we identify "physicist" users?

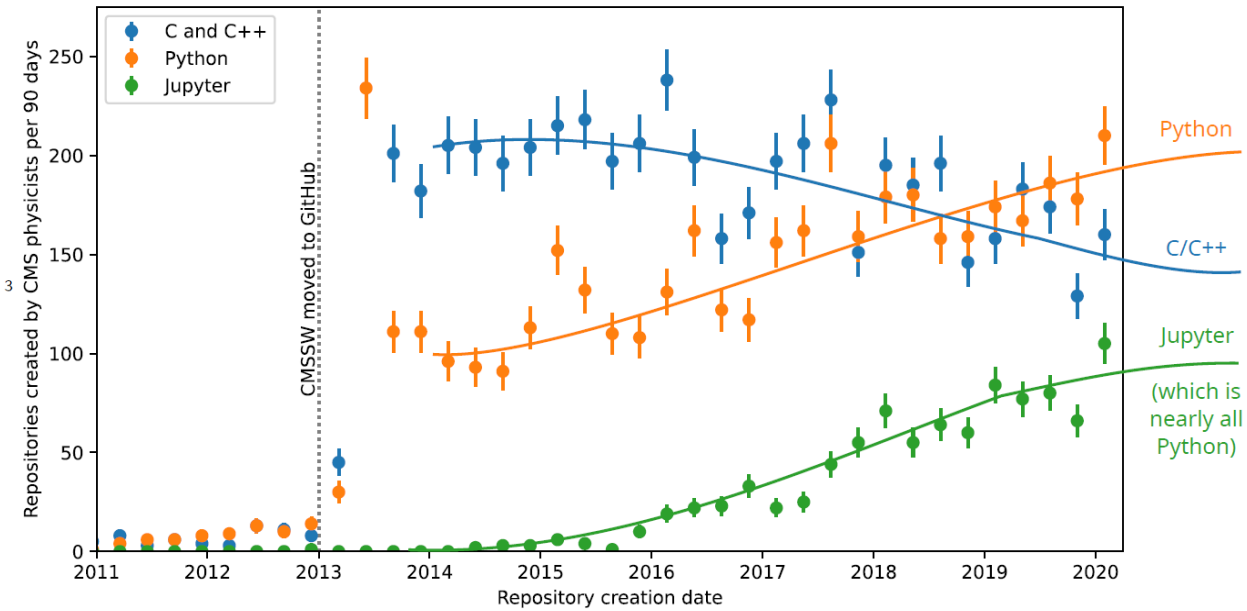
- CMSSW has been on GitHub since 2013.
- Assumption: most users who fork CMSSW are CMS physicists.
- Then examine their **non-fork** repositories.

Why GitHub/CMS? Until recently, all (free) GitHub repos were public, making them searchable by the API.

Large dataset: **3100 users** with **19 400 non-fork repos** spanning **7 years**.

- **Study by Jim Pivarski**
  [presentation @ Snowmass 2021, Aug. 11th]
- **Not from survey but rather directly using GitHub API to measure software adoption**

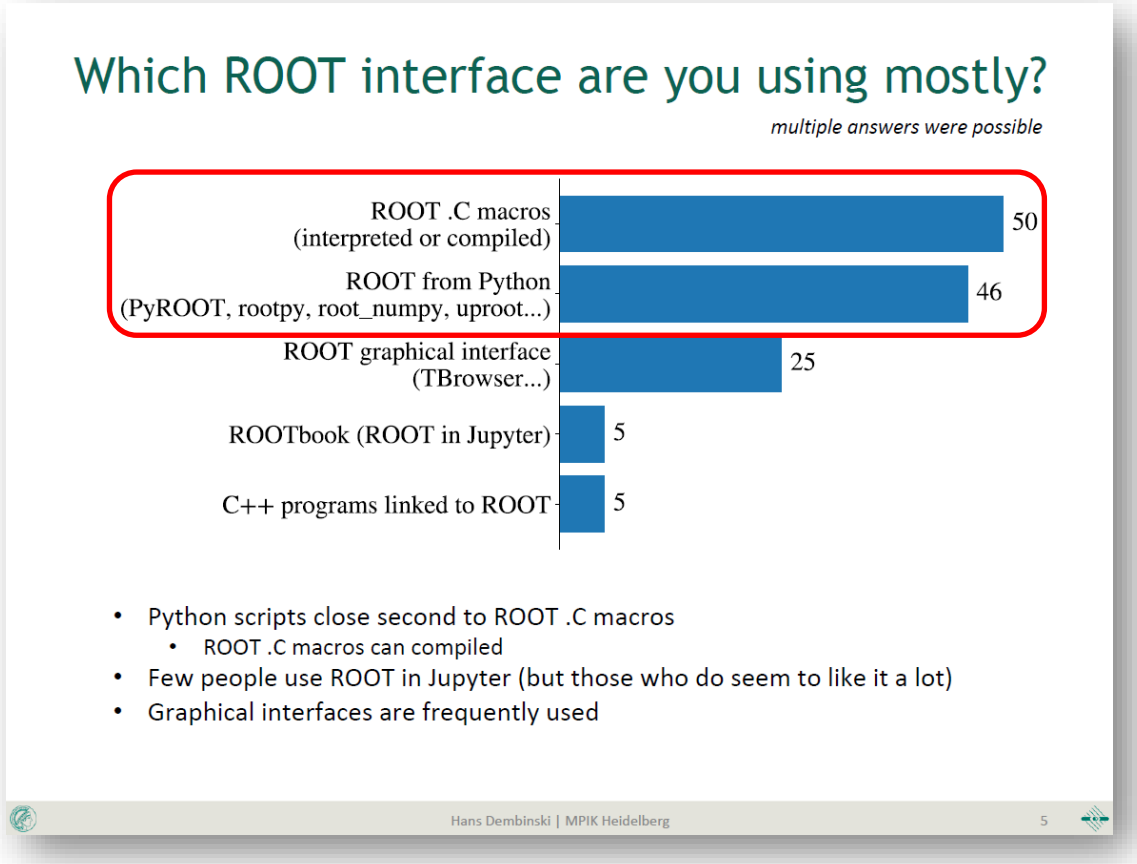
Language of repos created by CMS physicists



Python adoption in HEP – ROOT from Python in LHCb

Surveys from the LHCb experiment

- **Python and C++ equally used among analysts**
  - **Trend seen in our LHCb survey for the ROOT User's Workshop in 2018**
  - **And in the LHCb 2018 Analysis Survey Report (by Eduardo Rodrigues)**
- **ROOT from Python is just as used as is from plain C++ !**
- **Conclusion even stronger if discussing analysis tools independent of ROOT**



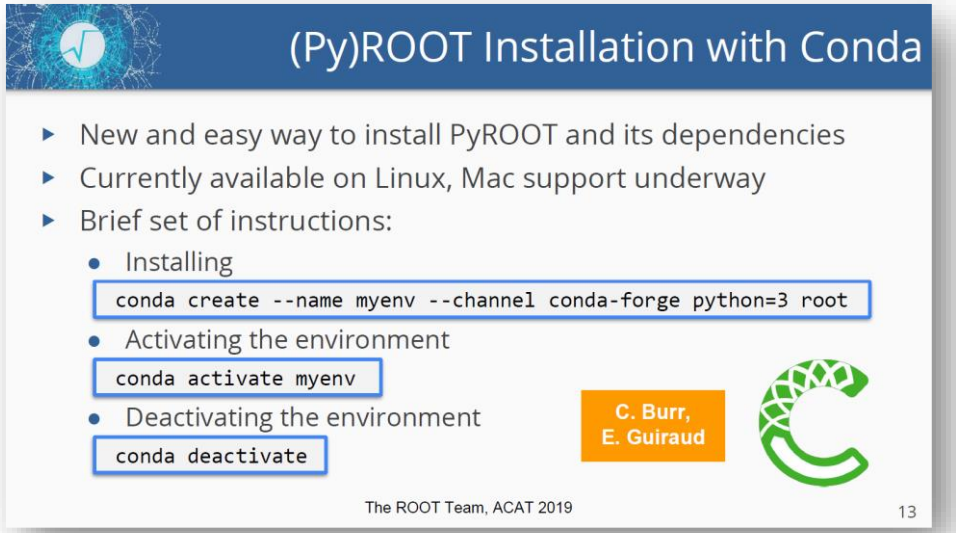
Taken from
Hans Dembinski, *User Feedback from LHCb*,
ROOT Users' workshop, Sarajevo, Sep. 2018

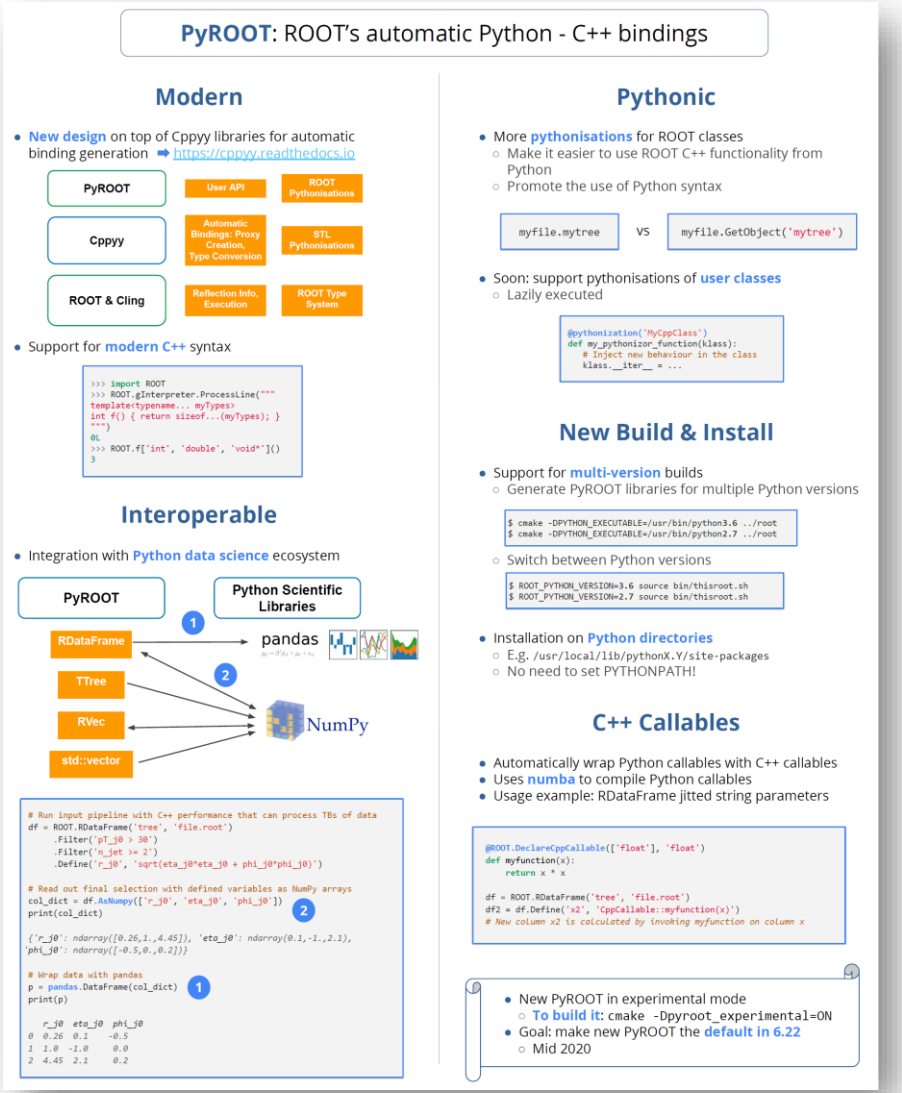
ROOT & Python

- (Py)ROOT has evolved enormously over the last few years !

- Some sources of material on latest goodies:
  - ICHEP 2020 talk on "Hello RNTuple and friends: what the new ROOT means for your analysis"
  - CHEP 2019 talk on "A New PyROOT: Modern, Interoperable and more Pythonic" :

- A game-changer in my opinion –installation via Conda !
  - Came largely from the community and not the ROOT team!





Python increasingly present in analysis tools used in publications

Full analysis likelihoods published on HEPData

- ❑ Test theory against LHC data
- ❑ All that's needed captured in a convenient format

- ❑ "Full likelihoods in all their glory" on HEPData
  - "While ATLAS had published likelihood scans …
    those did not expose the full complexity of the measurements"

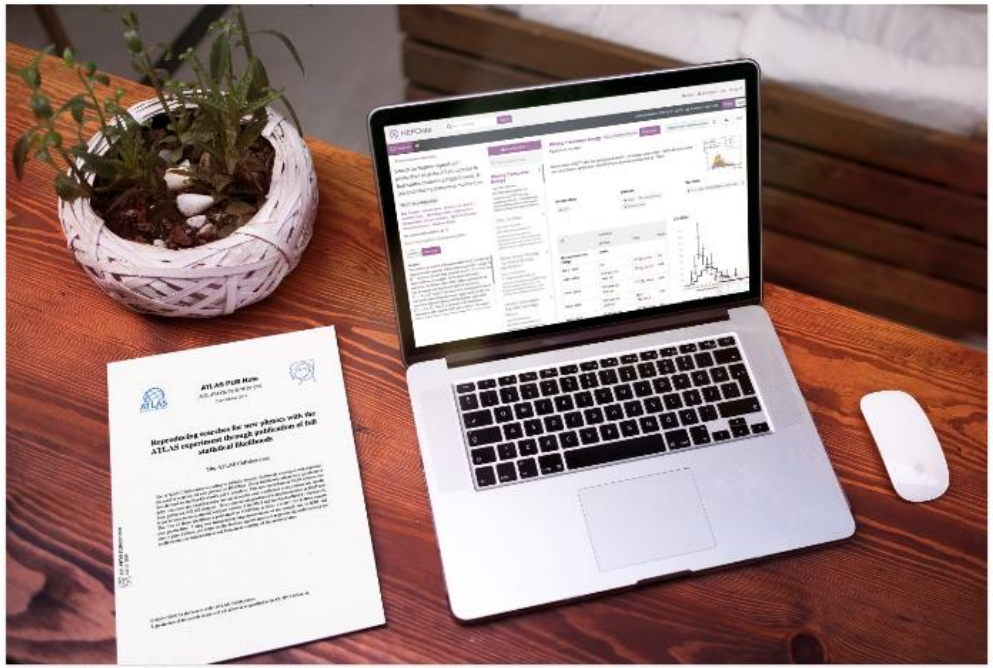
Work done with

- ❑ `RooStats` (C++)
- ❑ `pyhf` (Python)

New open release allows theorists to explore LHC data in a new way

The ATLAS collaboration releases full analysis likelihoods, a first for an LHC experiment

9 JANUARY, 2020 | By Katarina Anthony



Explore ATLAS open likelihoods on the HEPData platform (Image: CERN)

What if you could test a new theory against LHC data? Better yet, what if the expert knowledge needed to do this was captured in a convenient format? This tall order is now on its way from the ATLAS collaboration, with the first open release of full analysis likelihoods from an LHC experiment.

Taken from https://home.cern/news/news/knowledge-sharing/new-open-release-allows-theorists-explore-lhc-data-new-way

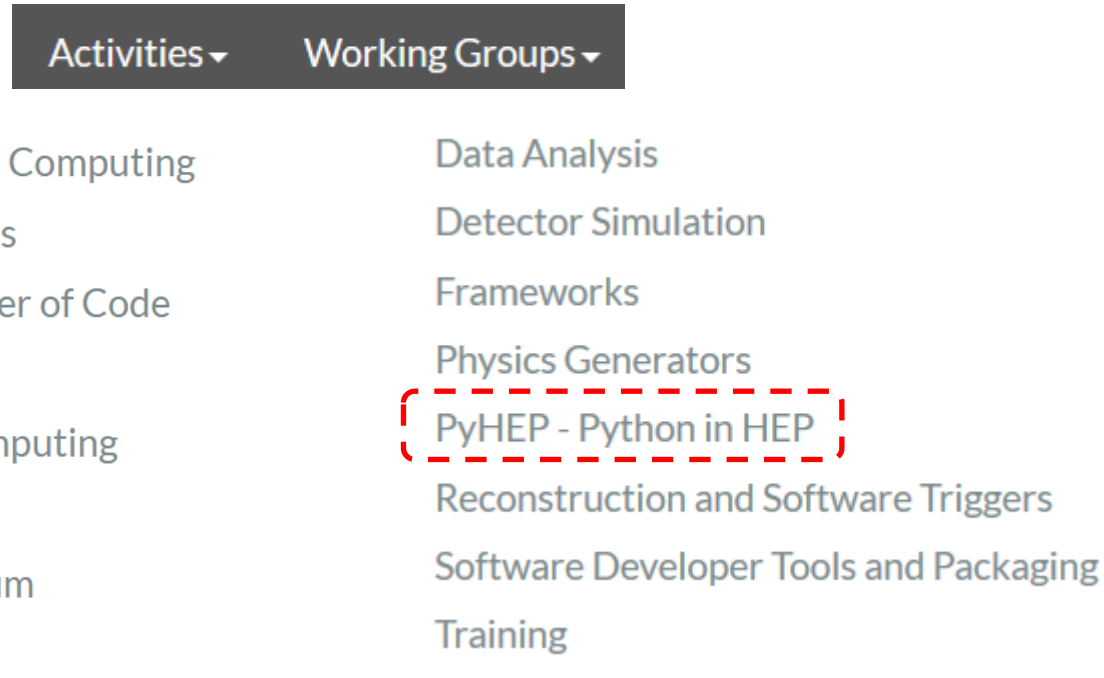
HSF & PyHEP

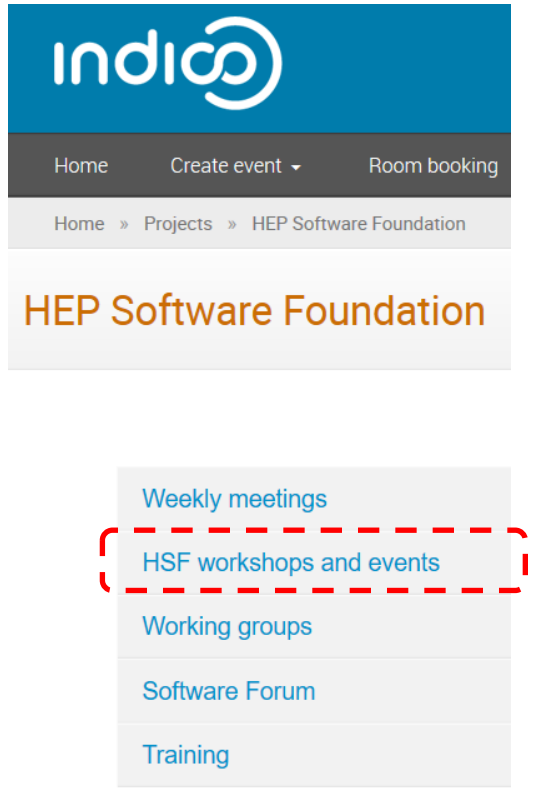
- The HSF PyHEP – “Python in HEP” Working Group
- PyHEP series of workshops
- Community projects towards a HEP Python ecosystem

- BTW, *a lot more information* in the back-up slides …

HSF – PyHEP ("Python in HEP") Working Group

- The "Python in HEP" WG effectively started in early 2018 as an activity group
  - I put it forward with the proposal of the 1st workshop, held as a pre-CHEP 2018 event
- It became "formally" a WG last year ☺





HSF – PyHEP ("Python in HEP") Working Group

- **Lots of ways to communicate !**
  - **The main (Gitter) channel now has over 160 people registered**

Note: "embraces the broad community". "Broader" is even better ☺ !

The PyHEP working group brings together a community of developers and users of Python in Particle Physics, with the aim of improving the sharing of knowledge and expertise. It embraces the broad community, from HEP to the Astroparticle and Intensity Frontier communities.

The group is currently coordinated by Ben Krikler (CMS, LZ), Eduardo Rodrigues (LHCb) and Jim Pivarski (CMS). All coordinators can be reached via hsf-pyhep-organisation@googlegroups.com.

Getting Involved

Everyone is welcome to join the community and participate by means of the following:

- Gitter channel PyHEP for any informal exchanges.
- GitHub repository of resources, e.g., Python libraries of interest to Particle Physics.
- Twitter Handle: #PyHEP

Extra Gitter channels have been created by and for the benefit of the community:

- PyHEP-newcomers for newcomers support (very low entry threshold).
- PyHEP-histogramming for discussions around histogramming.
- mpl-hep for Matplotlib proposals related to Particle Physics.

PyHEP Series of Workshops

Community projects towards a HEP Python ecosystem for data analysis

❑ Citing Gordon Watts (ACAT 2019) – how can we tackle the following issues?

- Increased LHC dataset sizes and CPU requirements
- Flat budgets & stable or decreasing staffing
- New software tools and communities inside and outside HEP
- High turn-over inside HEP
- Educational responsibility

Tackle them as a community !

(Note that much of this is not HEP specific ;-))

❑ PyHEP WG serves as a forum for discussion, means to exchange experiences and material

❑ Our workshops present many of these packages and provide educative material

⇒ *strong link with Training WG ☺*

Various projects have seen the light:

❑ Coffea

❑ FAST-HEP

❑ Scikit-HEP

❑ zfit

None existed in early 2016

❑ https://github.com/CoffeaTeam

❑ https://github.com/FAST-HEP

❑ https://github.com/root-project/

❑ https://scikit-hep.org/

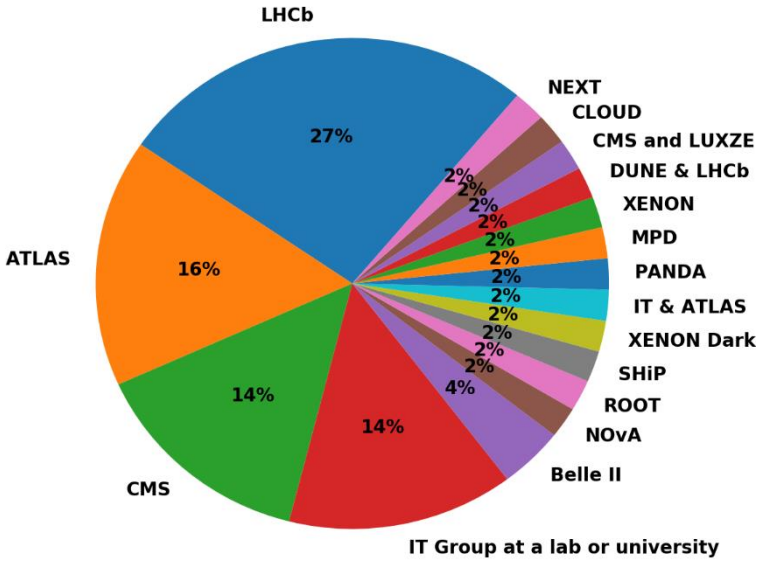
❑ https://github.com/zfit

PyHEP workshops – a (not so) new series of workshops

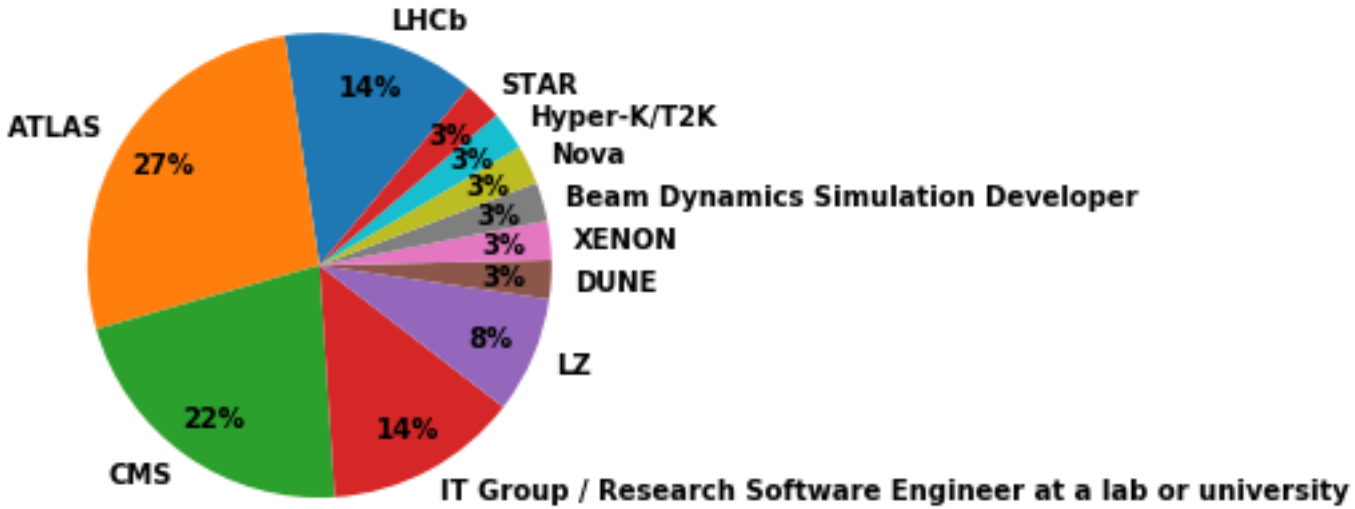
The **PyHEP workshops** are a series of workshops initiated and supported by the HEP Software Foundation (HSF) with the aim to provide an environment to discuss and promote the usage of Python in the HEP community at large. Further information is given on the PyHEP WG website.

❑ **Community diversity is paramount – great to see such a very diverse set of participants !**

PyHEP 2018
Sofia, Bulgaria

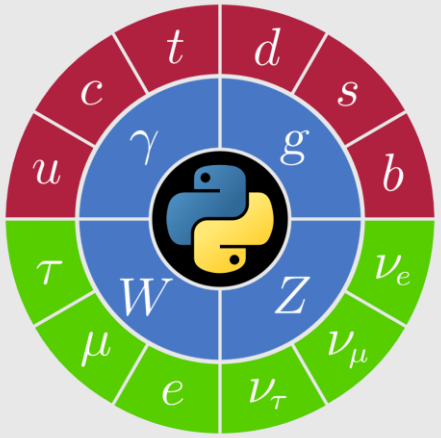


PyHEP 2019
Abingdon, U.K.



(Both pie charts taken from the pre-workshop questionnaires)

PyHEP 2020 Workshop



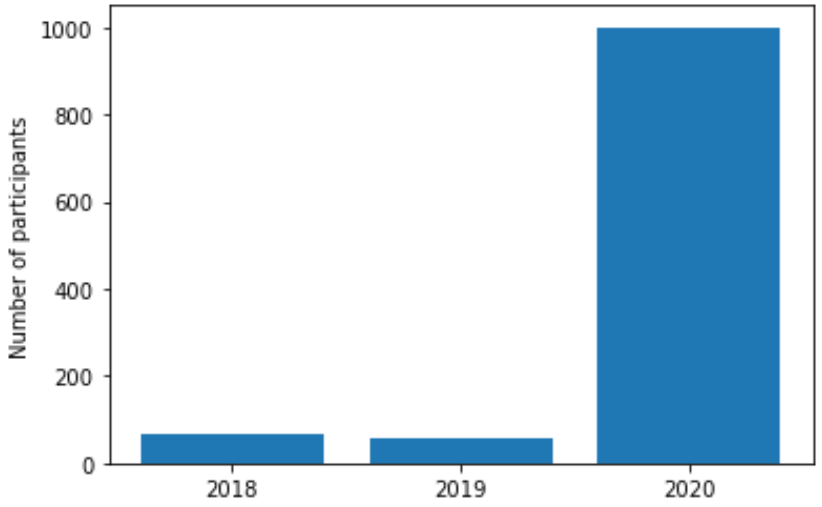
We now even have a logo ☺ !

- ❑ A special cuvée
- ❑ On organisational aspects
- ❑ Highlights

- ❑ BTW, *a lot more information* in the back-up slides …

PyHEP 2020, a special cuvée

- 3rd edition was meant to be in the US for the first time, co-locating with the important SciPy 2020 conference
  - We even had a nice poster ;-)!
- We engaged with this very large scientific community
  - Had several talks from HEP colleagues @ SciPy 2020
- But we both had to organise as a virtual event given the worldwide situation with COVID-19
- Truly global event with participants from all over the world (benefit from running virtual)
  - Impressive level of interest with 1000 registrations (limited to) (72, 55 in previous years)



PyHEP 2020 – Indico page, organising team, sponsors

PyHEP 2020 (virtual) Workshop

13-17 July 2020
US/Central timezone

- Overview
- Call for Abstracts
- Timetable
- Registration
- Participant List
- Poster
- Surveys
- Code of conduct
- EDI statement
- Workshop photos

Contact us
pyhep2020-organisation...

Organising Committee

Eduardo Rodrigues - University of Liverpool (Chair)
Ben Krikler - University of Bristol (Co-chair)
Jim Pivarski - Princeton University (Co-chair)
Matthew Feickert - University of Illinois at Urbana-Champaign

Local organisation

Chris Tunnell - Rice University
Peter Onyisi - The University of Texas at Austin

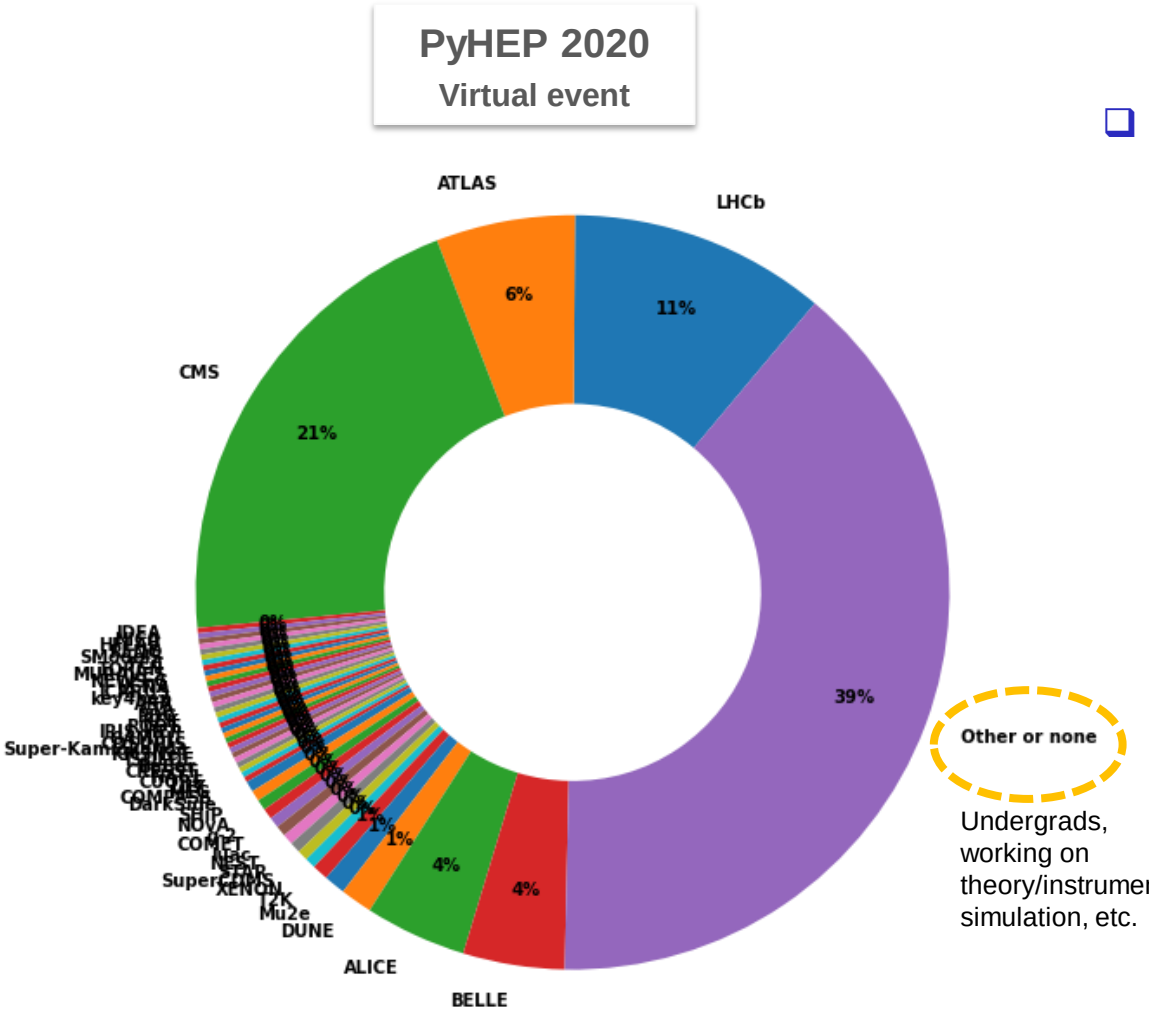
Sponsors

The event is kindly sponsored by

iris hep – Institute for Research & Innovation in Software for High Energy Physics
University of Liverpool
python SOFTWARE FOUNDATION
Software Sustainability Institute
Fermilab

- **Great list of kind sponsors is a proof of workshops being relevant and attracting attention – my personal opinion ;-)**

PyHEP 2020 stats – diversity and inclusion



- Great to see such a diverse set of participants !

(Pie chart and “logo art” with information taken from the pre-workshop questionnaire)

PyHEP 2020 organisational aspects – overview

- **Sessions & presentations**
  - Spread in sessions for “Atlantic”- and “Pacific”-friendly time zones
  - We strongly encouraged notebook presentations, available in public Github repositories with a Binder launch button
  - All presentational material posted on workshop agenda and later given a DOI with Zenodo, in a dedicated “pyhep2020 community” – formal citation, replaces proceedings
  - All talks got recorded, captioned and later uploaded to the HSF YouTube channel – dedicated playlist “PyHEP 2020 Workshop”

- **Zoom video conferencing system**
  - With capacity for 1000 participants
  - Public room but PIN provided via email

- **Slack channels**
  - Various channels:
    - By topic, mapping to sessions, discussions encouraged here
    - Announcements, for actual announcements
    - Random, used to encourage community spirit and add social context

- **Questions & answers with slido**
  - Used *slido* to crowd-source questions, to prioritise the most popular ones upvoted by participants
  - Session chair shares link to questions at end of presentation
  - Most popular ones get answered/discussed
  - At end of Q&A all questions are copied to Slack in the appropriate topical channel
    ⇒ participants can continue to discuss and exchange
  - A few polls also run via slido

- **Communication also on**

PyHEP 2020 organisational aspects – agenda (1/2)

Workshop agenda (1/2)

Keynotes

- Rubin Observatory: the software behind the science (Nate Lust)
- Python & HEP: a perfect match, in theory (David Straub)

Tutorials

- Uproot & Awkward Arrays (Jim Pivarski)
- Jagged physics analysis with Numba, Awkward, and Uproot on a GPU (Joosep Pata)
- Ganga: flexible virtualization for user-based large computations (Ulrik Egede)
- A prototype U.S. CMS analysis facility (Oksana Shadura)
- Columnar analysis at scale with Coffea (Mat Adamec)
- Introduction to automatic differentiation (Lukas Heinrich)
- High-performance Python (Henry Schreiner)
- Model-building & statistical inference with zfit and hepstats (Jonas Eschle)
- pyhf: accelerating analyses and preserving likelihoods (Matt Feickert)
- ThickBrick: optimal event selection and categorization in HEP (Prasanth Shyamsundar)

Typically 45 minutes

Eduardo Rodrigues PyHEP 2020, All@Home, 13 July 2020 6/11

PyHEP 2020 organisational aspects – agenda (2/2)

Workshop agenda (2/2)

Talks

- NanoEvents object (Nick Smith)
- TITANIA: how to structure dector monitoring (Jakub Kowalski, Maciej Witold Majewski)
- A new PyROOT for ROOT 6.22 (Enric Tejedor Saavedra)
- Resample: bootstrap and jackknife from Python (Hans Dembinski)
- Design pattern for analysis automation using Luigi (Marcel Rieger)
- ServiceX: on-demand data transformation & delivery (Kyungeon Choi)
- Integrating Coffea and WorkQueue (Cami Carballo)
- High granularity calorimeter (HGCAL) test beam analysis using Jupyter (Matteo Bonanomi)
- neos: physics analysis as a differentiable program (Nate Simpson)
- SModelS: a tool for interpreting simplified-model results (Wolfgang Waltenberger)
- TensorFlow-based maximum likelihood fits for high-precision Standard Model measurements at CMS (Josh Bendavid)
- Error computation in iminuit and MINUIT: how HESSE and MINOS work (Hans Dembinski)
- zfit with TensorFlow 2.0: dynamic and compiled HPC (Jonas Eschle)
- Machine learning for signal-background separation of nuclear interaction vertices in CMS (Anna Kropivnitskaya)
- The boost-histogram package (Henry Schreiner)
- Providing Python bindings for complex and feature-rich C and C++ libraries (Martin Schwinzerl)
- Integrating GPU libraries for fun and profit (Adrian Oeftiger)
- mplhep: bridging Matplotlib and HEP (Andrzej Novak)
- ROOT preprocessing pipeline for machine learning with TensorFlow (Matthias Komm)
- Integrated data acquisition in Python (Charles Burton)

Typically 20+10 minutes

Eduardo Rodrigues PyHEP 2020, All@Home, 13 July 2020 7/11

On 2021 plans – interested? Let's discuss and engage with each other

PyHEP 2021 workshop

- Pretty clear that we will organise such an event
- Likely that we tag along with SciPy again, hence do our workshop just after SciPy, for example
  - SciPy scheduled for July 5-11, undecided if f2f or virtual AFAIK
- We plan again a good mix of talks and tutorials. Your engagement/participation is welcome !

PyHEP WG topical meetings

- Interest in topical meetings was one of the workshop feedbacks
- Idea came up to follow somewhat the theme of Python 3 Module of the Week,
  but with a spirit adapted to our needs, hence rather a "Python Module of the Month",
  presentations with a focus on libraries relevant to data analysis in Particle Physics …
  … and why not relevant also to Nuclear Physics?
- First tutorial-like presentation will be on Feb. 3rd on Numba, see Indico

Let’s collaborate!

Your perspective will be enriching

- HEP Software Foundation (HSF)
  - HSF general forum hsf-forum@googlegroups.com
- HSF PyHEP Working Group
  - (main) Gitter channel
  - GitHub repository “Python in HEP” resources
- Topical meetings and PyHEP 2021
  - 1st topical meeting on Feb. 3rd

The HEP Software Foundation (HSF)

- The goal of the HEP Software Foundation (HSF) is to facilitate coordination and common efforts in software and computing across HEP in general
  - Our philosophy is bottom up, a.k.a. *Do-ocracy*
  - Also work in common with like-minded organisations in other science disciplines

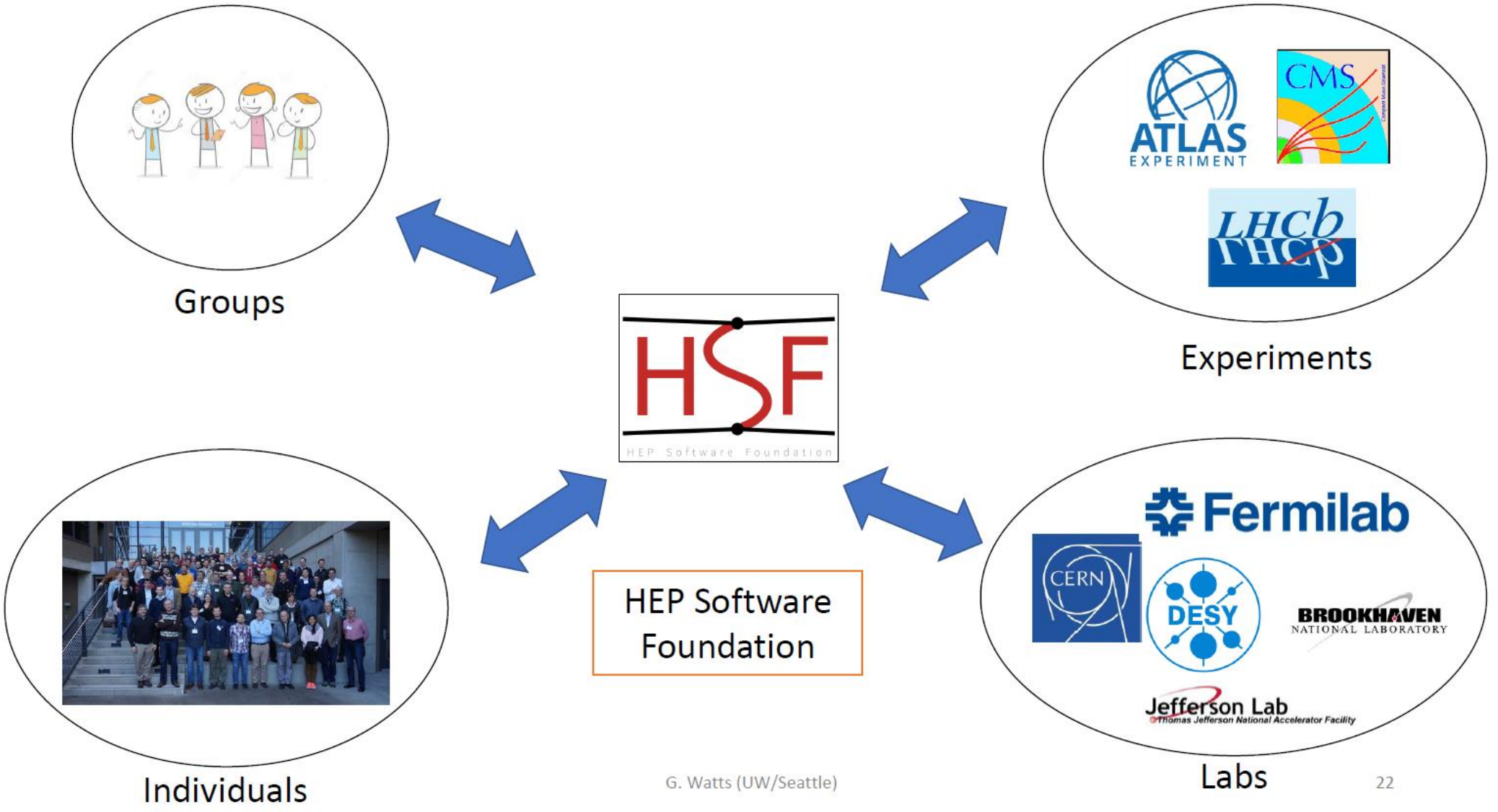
- Founded in 2014, explicitly to address current and future computing & software challenges in common

- Finalised in Dec. 2017 a Community White Paper (CWP) "A Roadmap for HEP Software and Computing R&D for the 2020s"
  - Almost all major domains of HEP Software and Computing covered
  - Large support for the document from the community (> 300 authors from >120 institutions)
  - Comput Softw Big Sci (2019) 3, 7; arXiv:1712.06982

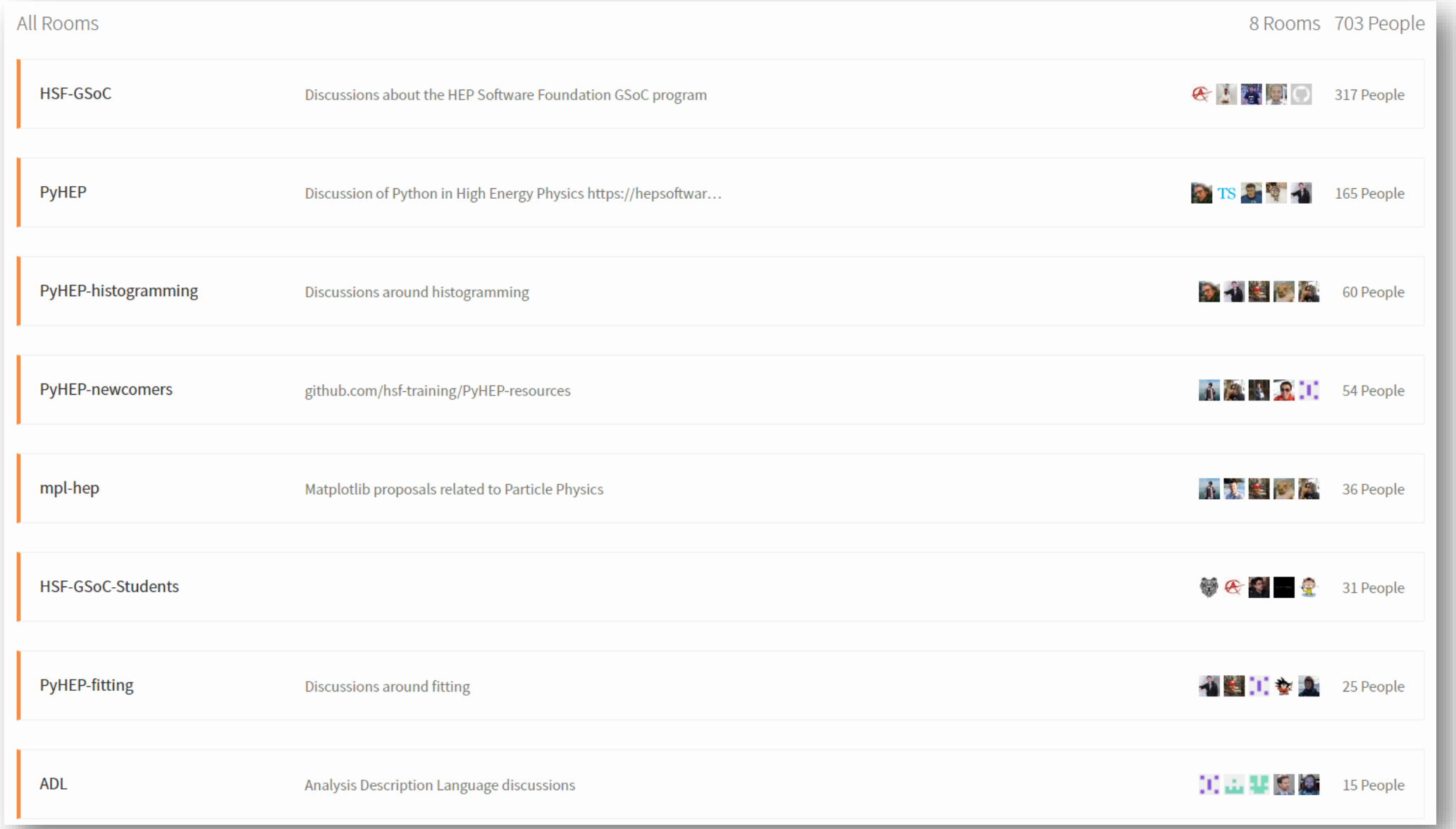
- The CWP was a major accomplishment made by the community, with HSF "coordination"
- But it was a milestone, not a final step
- HSF activities post-CWP are very diverse …

- 2020: new community document "HL-LHC Computing Review: Common Tools and Community Software", Stewart, Graeme Andrew *et al*. (2020, May 1). Zenodo. http://doi.org/10.5281/zenodo.3779250, HSF-DOC-2020-01

HSF - "facilitate coordination and common efforts"



HSF – Gitter channels



PyHEP series of workshops

PyHEP 2018

Sofia, Bulgaria

PyHEP 2019

Abingdon, U.K.

PyHEP 2020

- *Was meant to be held in Austin (Texas), U.S.A., in July 11-13*
- *Next to SciPy 2020 conference, to enhance cross-community exchange*
- *Run as a virtual event, as most conferences this year*

PyHEP workshops – diverse topics presented/discussed

PyHEP 2018
Sofia, Bulgaria

- Historical perspective / overview
- HEP python software ecosystem
- Analysis & HEP frameworks
- PyROOT and Python bindings
- Distribution and installation
- Python 2 to 3
- Open discussion on education and training

+

Keynote presentation on JupyterLab

PyHEP 2019
Abingdon, U.K.

- Accelerators-enabled code
- Analysis platforms
- Analysis fundamentals
- HEP Python software ecosystem
- High-level analysis tools
- Histogramming
- Packaging, distribution, CI
- PyROOT
- Research software
- Statistics
- Visualisation

- **Lightning talks**

- ❑ **Organisation:**
  - **- Topical sessions, all plenary**
  - **- 1/3 of time devoted to discussions rather than presentations**
- ❑ **Pre- and post-workshop surveys**
- ❑ **Live notes taken during the sessions**

PyHEP 2020 logistics – Jupyter notebook talks / tutorials with Binder

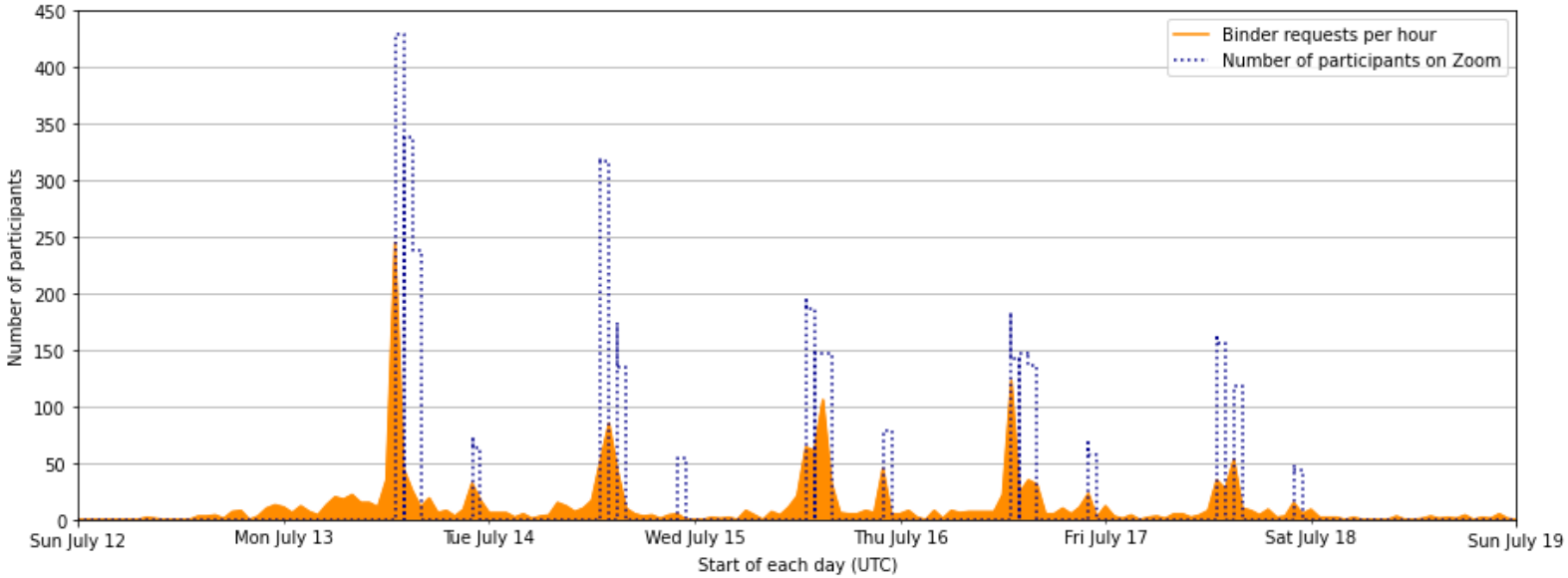
- We relied on Binder to have interactive computing experiences for all Jupyter notebook presentations
- Speakers with notebooks were requested to have a "launch binder" badge in their talk repositories
- Binder:
  - Free open-source project and service from the Jupyter team
  - Runs on donated compute resources from the Binder Federation
- We used both Binder Federation and CERN Binder Hub resources (for those with CERN accounts)
  - Got in touch with Binder team to have resources allocated to talk repositories at the relevant time !
  - It worked very well – thank you MyBinderTeam
  - Binder was a leitmotif during the workshop:
- With Zenodo + Binder, all code from the workshop should be reproducible into the future
  ⇒ "living workshop proceedings"!
- Find out more at mybinder.org

From PyHEPConf Twitter account

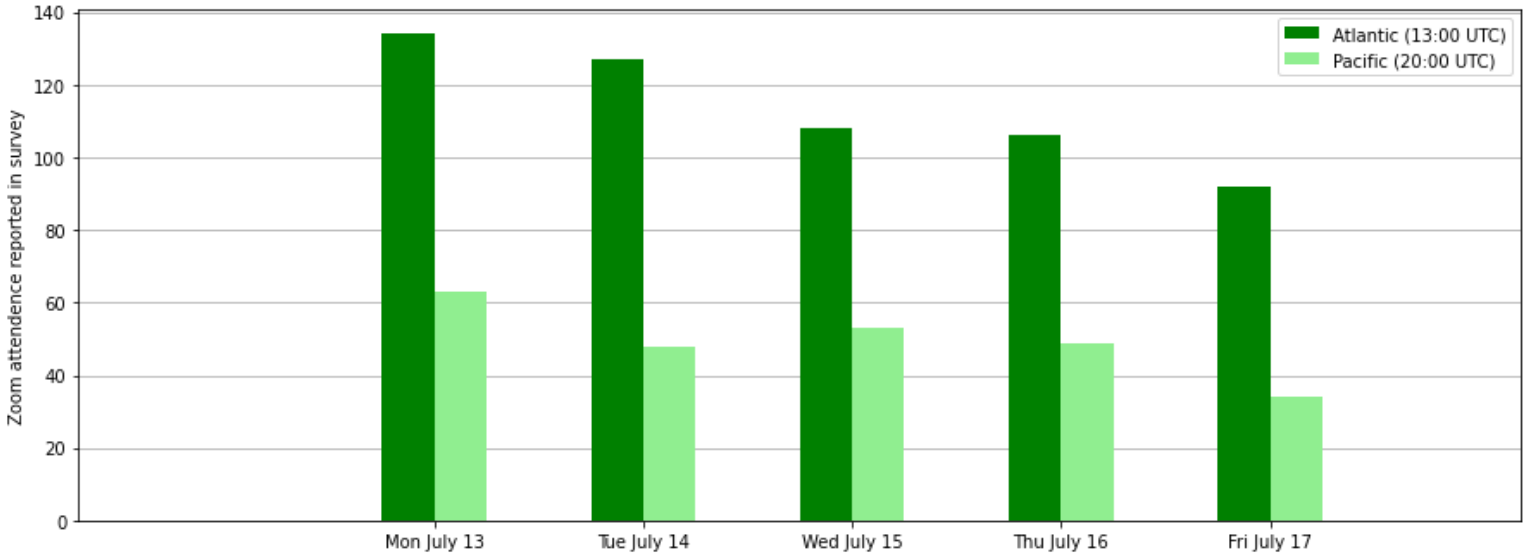
PyHEP 2020 logistics – session attendance & Binder usage

- Session participants
- Binder requests during sessions

⇒ Clear correlation !



- Number of participants per day & time zone, *as reported by those who filled in the post-workshop survey*
  - “Atlantic” time zone suited most

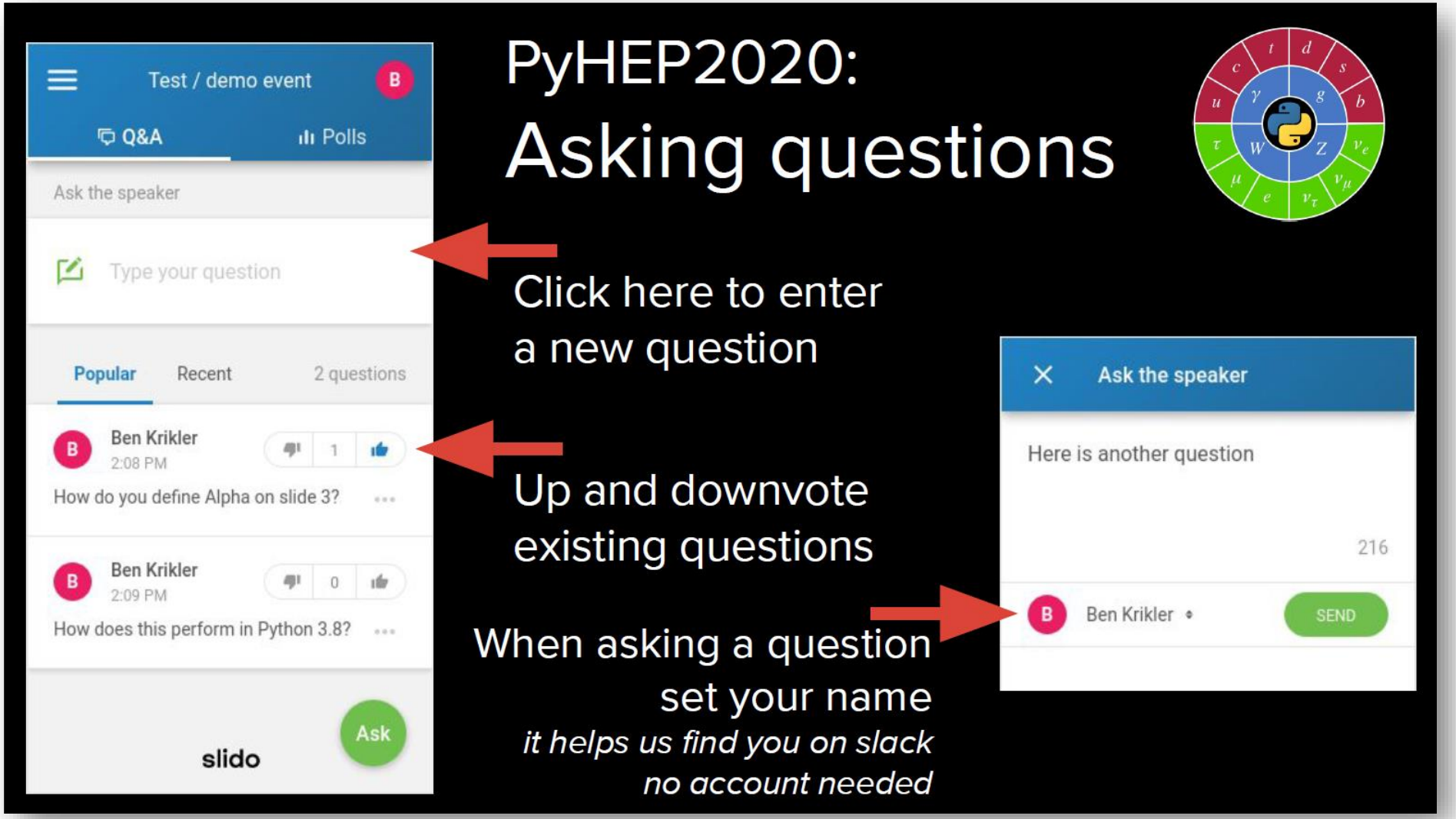


Study by Jim Pivarski

PyHEP 2020 logistics – how does slido work for Q&As

slido

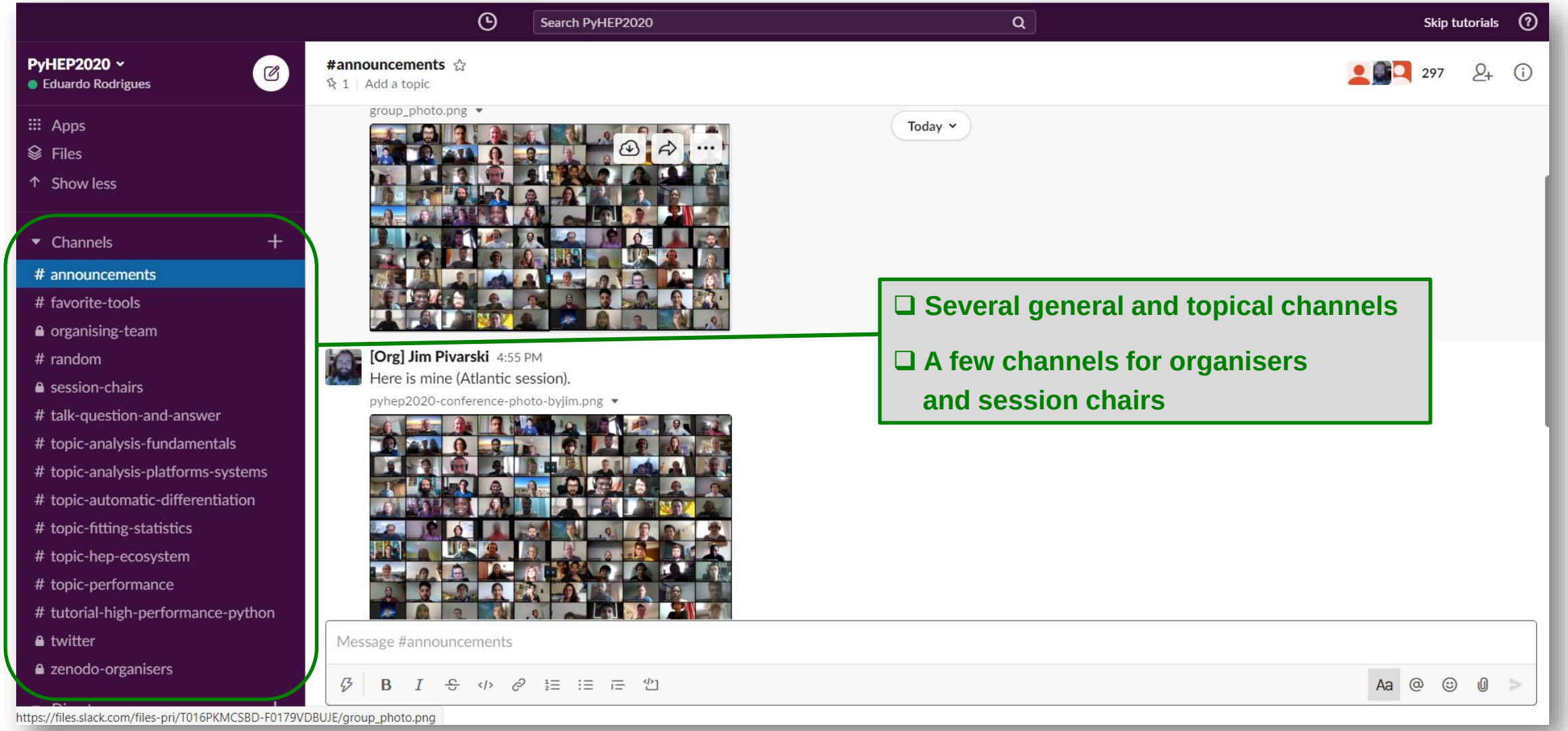
Easy to use Works with your live video No app downloads



PyHEP 2020 logistics – slido at work for Q&As and polls

As actually seen by participants

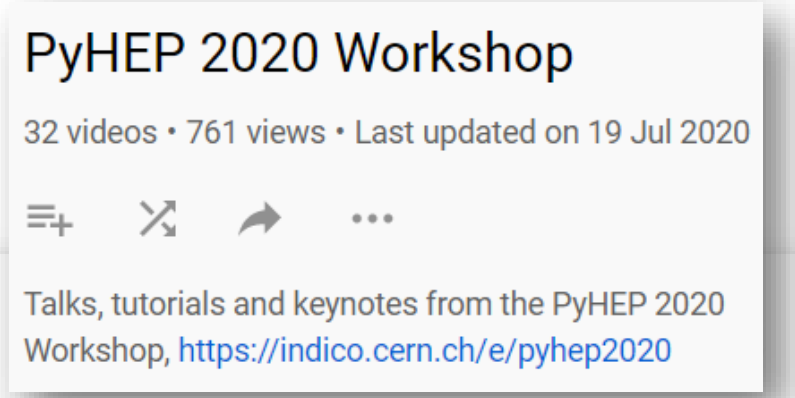
PyHEP 2020 logistics – Slack for discussion during/after sessions

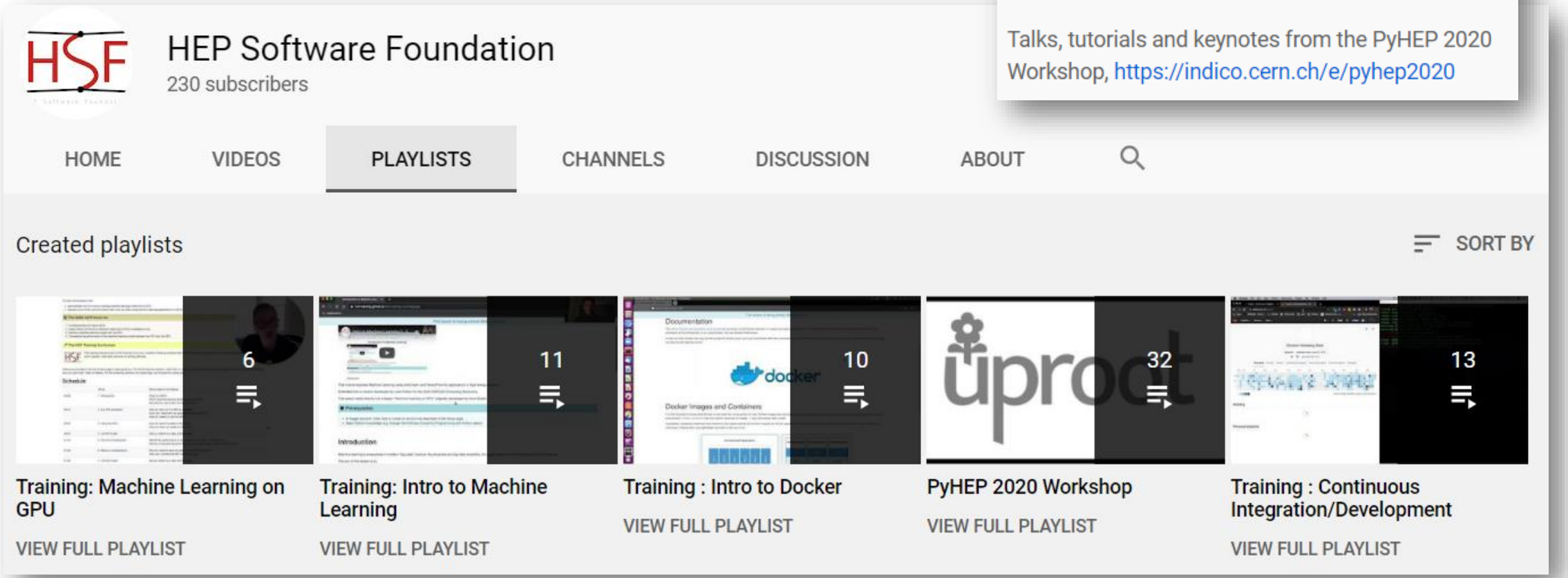


- Several general and topical channels
- A few channels for organisers and session chairs

PyHEP 2020 logistics – recordings on YouTube

- HSF has its own channel, with several playlists
- Recordings of all presentations on YouTube, captioned, in dedicated playlist



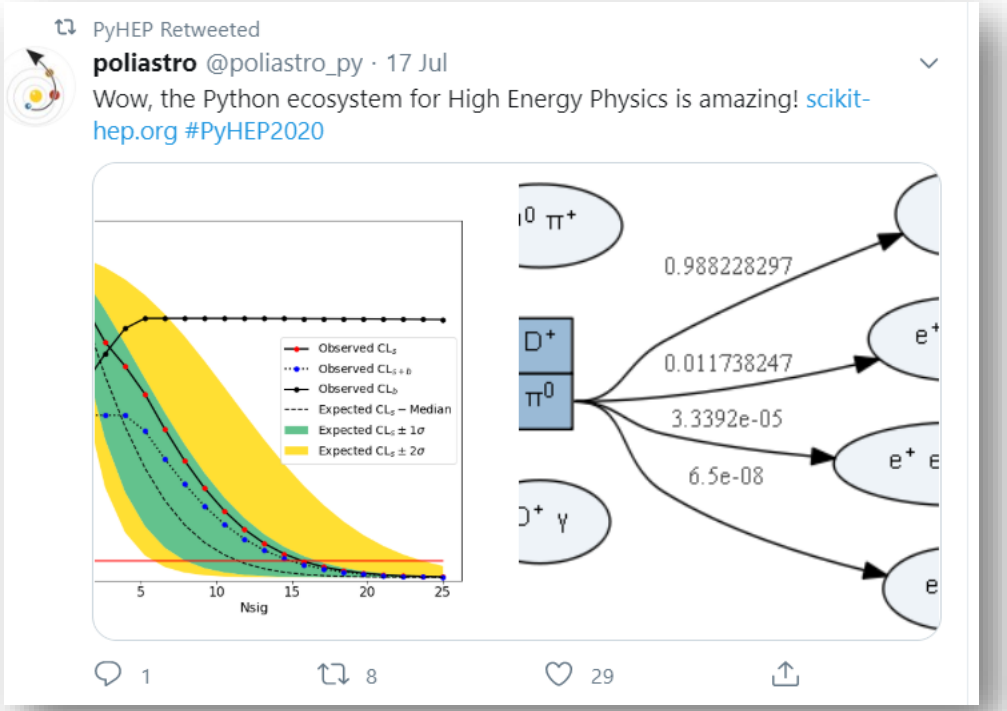


PyHEP 2020 logistics – we are even on

@PyHEPConf

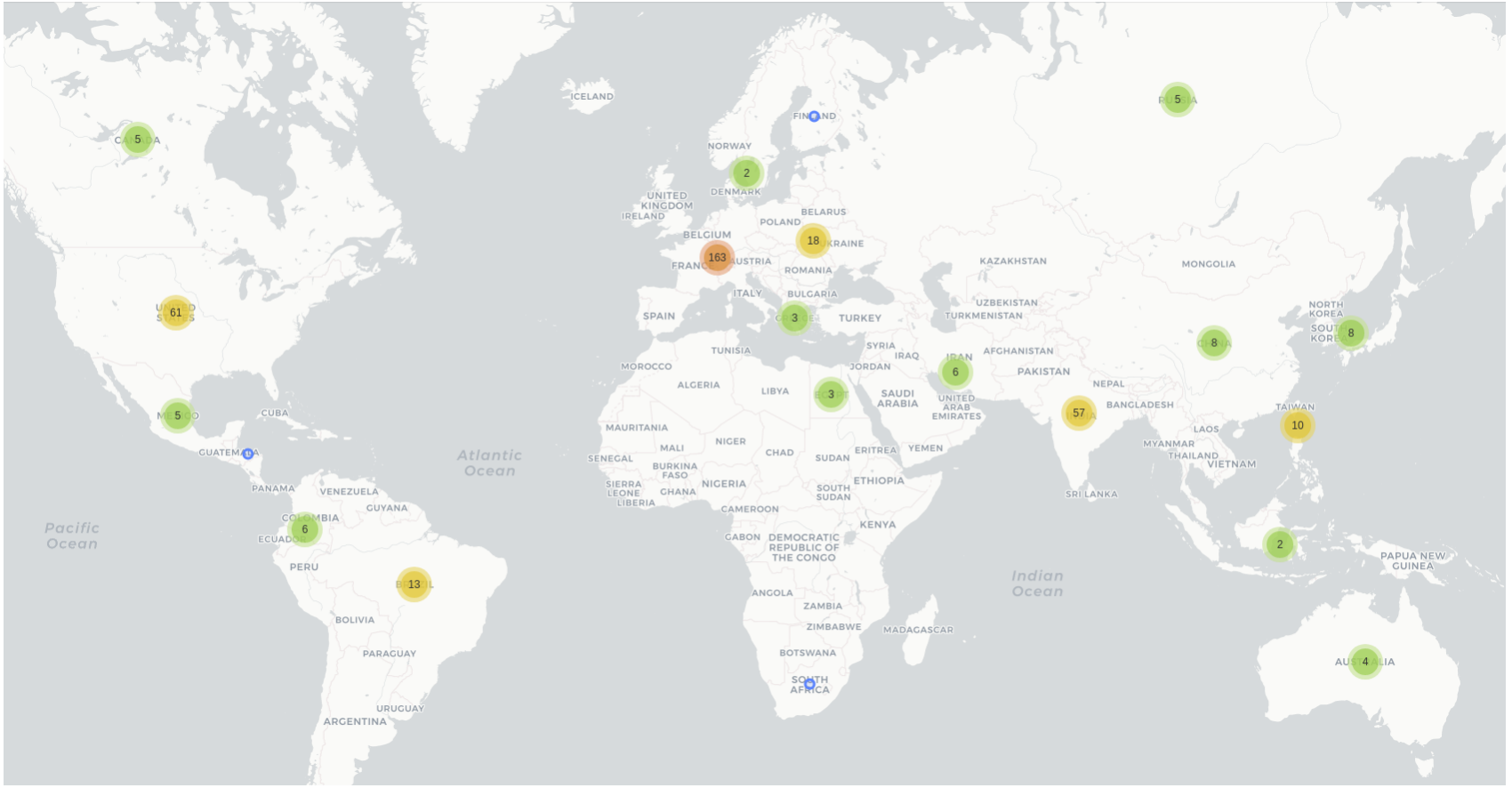
#PyHEP2020

A testimony from an astro-particle colleague …



PyHEP 2020 stats – diversity and inclusion

- **Diverse participation from all over the world !**



Plot by Jim Pivarski

- **Information taken from the 408/1000 responses received from the pre-workshop survey**

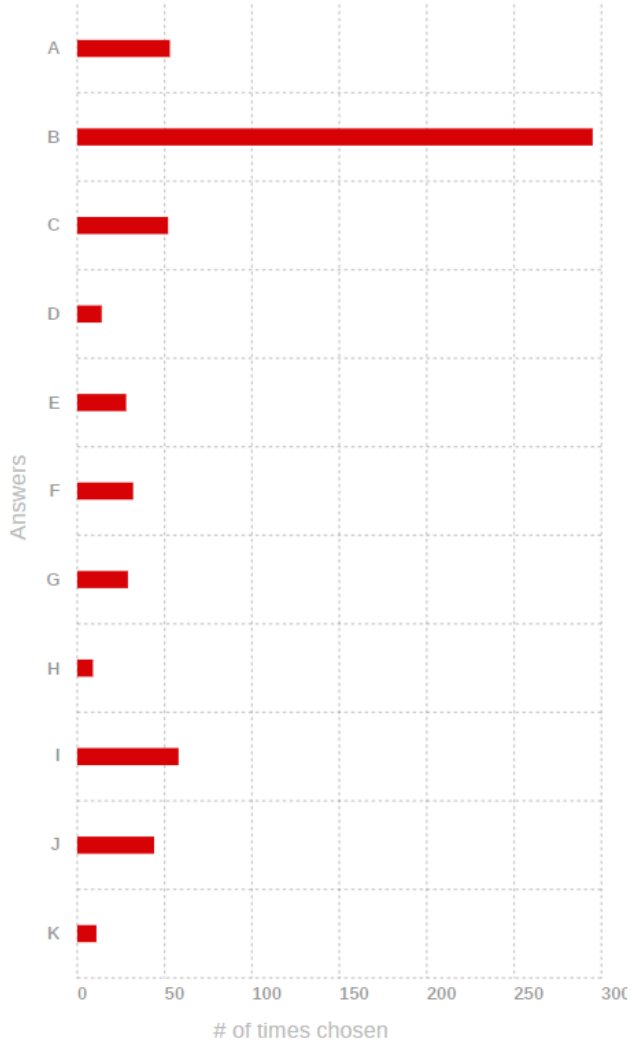
PyHEP 2020 stats – background of participants …

If you're involved in physics, what area(s) do you study?

Answered : *405*

You can answer this AND the area of computing (below) or only one, depending on what you do.

- **A. General physics (student)**: 53 (8.48%)
- **B. High-energy collider physics**: 295 (47.20%)
- **C. Neutrino physics**: 52 (8.32%)
- **D. Physics of nuclei or exotic atoms**: 14 (2.24%)
- **E. Precision frontier**: 28 (4.48%)
- **F. Direct dark matter searches**: 32 (5.12%)
- **G. Astroparticle physics**: 29 (4.64%)
- **H. Astronomy**: 9 (1.44%)
- **I. Theory/simulations**: 58 (9.28%)
- **J. Instrumentation**: 44 (7.04%)
- **K. Other, not listed above**: 11 (1.76%)



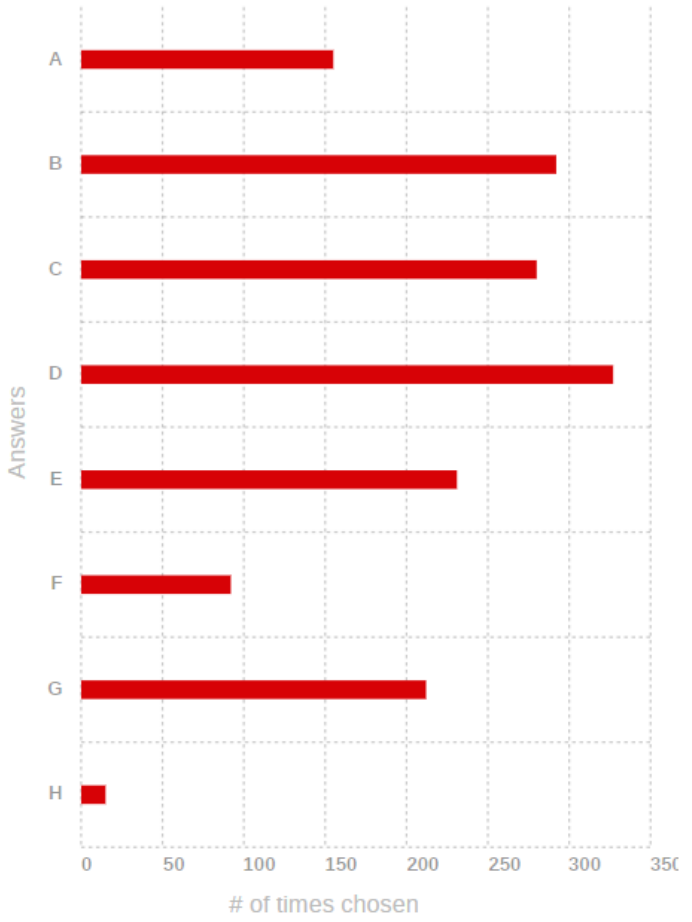
Taken from the pre-workshop survey (408 respondents)

PyHEP 2020 stats – … and their hopes

What are you hoping to learn from this workshop?

Answered: *405*

- **A. Python fundamentals (how to program in Python)**: 155 (9.66%)
- **B. General-purpose data analysis toolkits**: 292 (18.20%)
- **C. Machine learning/deep learning toolkits**: 280 (17.46%)
- **D. Particle physics analysis tools (other than ROOT)**: 327 (20.39%)
- **E. ROOT and PyROOT**: 231 (14.40%)
- **F. Collaboration-specific topics**: 92 (5.74%)
- **G. Software engineering skills (beyond the fundamentals)**: 212 (13.22%)
- **H. Other**: 15 (0.94%)



Taken from the pre-workshop survey (408 respondents)

PyHEP 2020 highlights – on workshop topics

Word cloud of abstracts

(Made with https://www.wordclouds.com/
removing author names, institutes and some other trivial words.)

- ❑ **Many topics**
- ❑ **Too much content to adequately review here!**

  - **Analysis fundamentals**
  - **Analysis platforms & systems**
  - **Automatic differentiation**
  - **Performance**
  - **Fitting & statistics**
  - **HEP analysis ecosystem**

  +

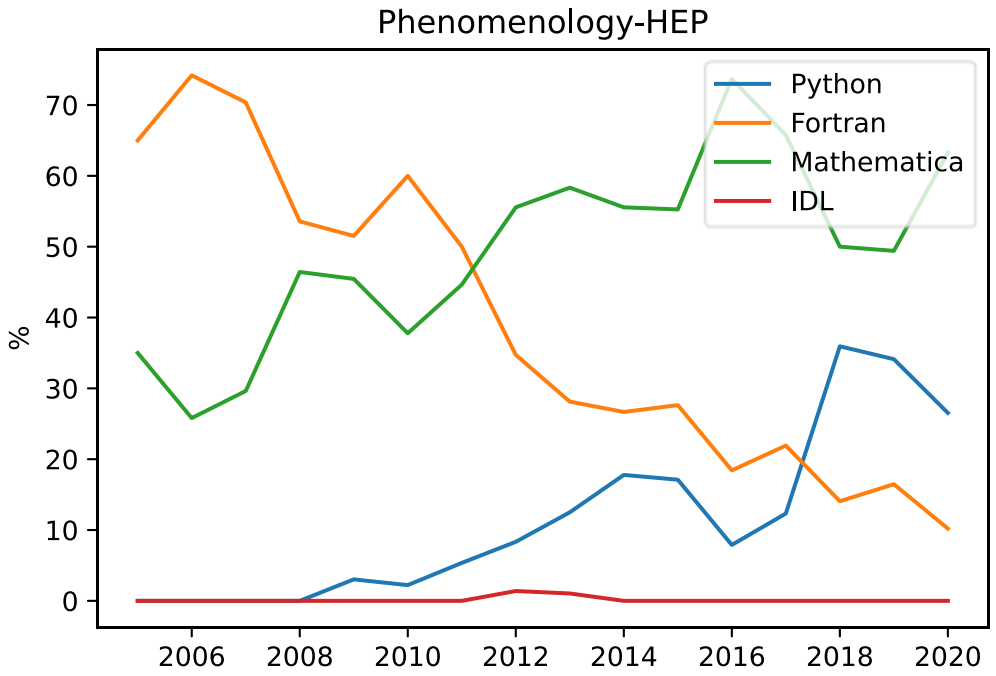
  2 keynote presentations (astronomy & pheno.)

- ❑ **Organisation:**
  - **Topical sessions, all plenary**
  - **Tutorials and standard talks**
  - **Much time devoted to discussions**
- ❑ **Pre- and post-workshop surveys**

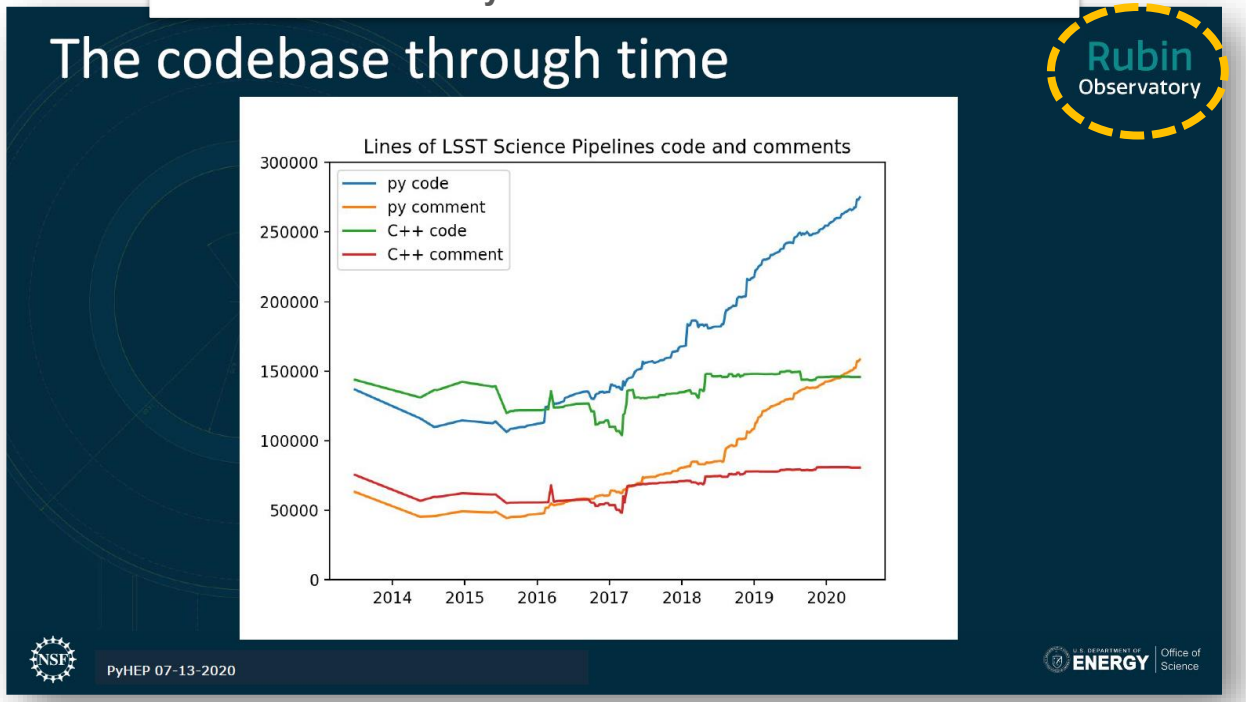
PyHEP 2020 highlights – keynote presentations

- **Python on the rise not just in experimental particle physics**

David Straub (flavour phenomenologist)
"Python & HEP: a perfect match, in theory"



Nate Lust (astronomy)
"Rubin Observatory: The software behind the science"



Challenges for Python in HEP-Ph

Python's full potential is harnessed when embracing the **open source paradigm**:

- Open source code
- Transparency (development, decision making, bugs!)
- Release early and often (software is **not** a paper!)
- Community

In HEP-Ph, there are very few open source projects in this sense, only "public codes".

PyHEP 2020 highlights – on trends

" Automatic differentiation is a method to compute exact derivatives of functions implements as **programs**. It's a widely applicable method and famously is used in many Machine learning optimization problems."

- **Auto-differentiation, specifically in the context of differentiable analysis, came out as an unforeseen "theme" and a new direction**
  - 1 tutorial and 1 talk on the subject

  - **Introduction to automatic differentiation (TUTORIAL)**
  - **neos: physics analysis as a differentiable program**

In HEP

Of course we can use automatic differentiation for neural networks. But other things in HEP also can make use of gradients. A prime example where this is the case is statistical analysis

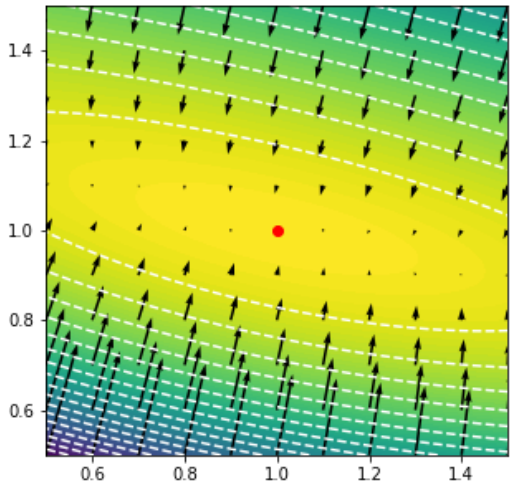
For a maximum likelihood fit we want to minimize the log likelihood

$\theta^* = \text{argmin}_\theta(\log L)$

```
import jax
import jax.numpy as jnp
import numpy as np
import pyhf
import matplotlib.pyplot as plt
```

```
pyhf.set_backend('jax')
```

Define the model, fit … and plot:



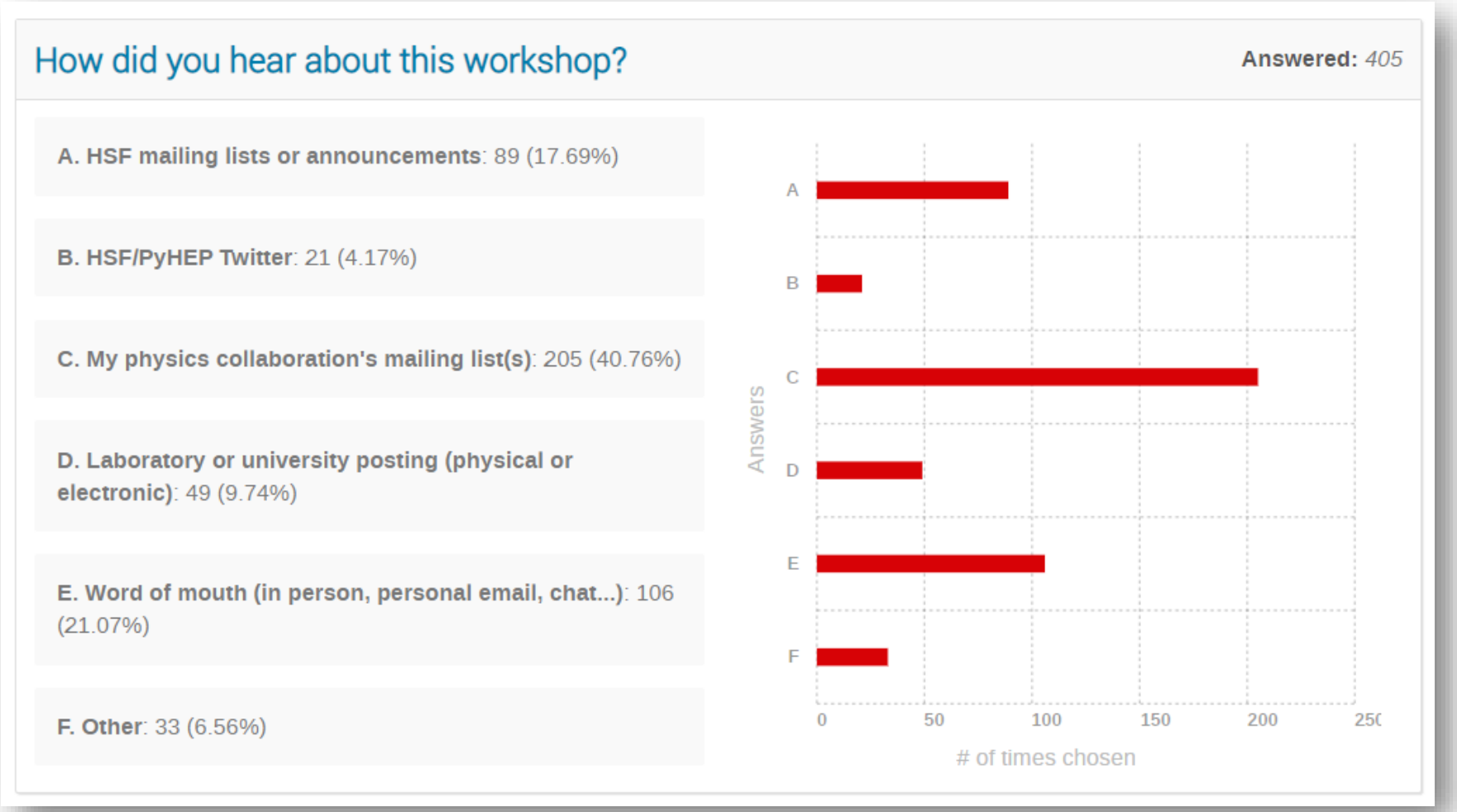
(Taken from the tutorial)

gradHEP is an effort to consolidate differentiable building blocks for analysis into a set of common tools, and apply them.

See the 'Differentiable computing' HSF activity to find ways to get involved -- all are welcome at this very early stage! :)

gradHEP

PyHEP 2020 organisational aspects – multi-channel advertising is crucial



Taken from the pre-workshop survey (408 respondents)

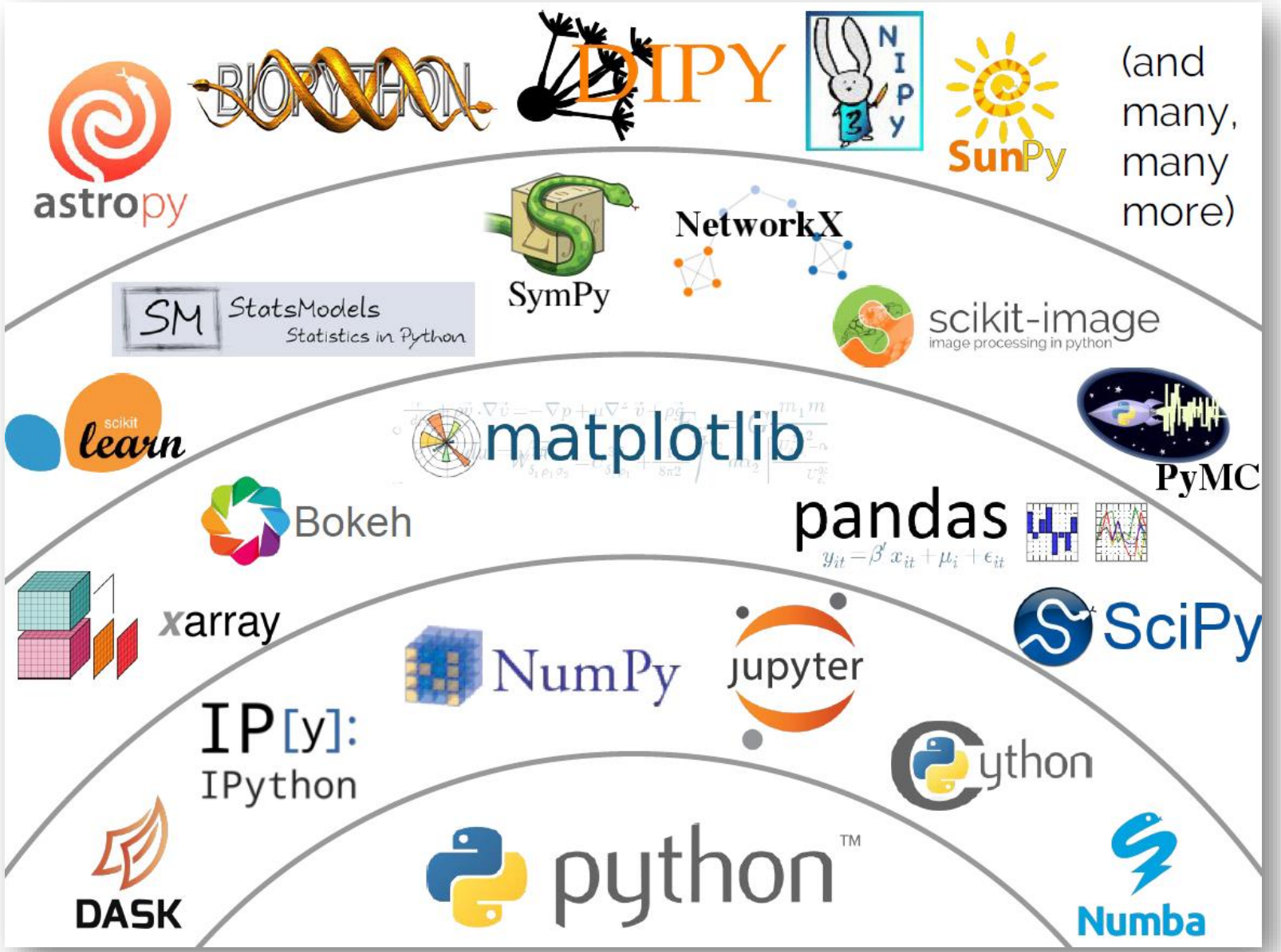
How's the Python scientific ecosystem like, outside HEP?

Domain-specific

Python's

Scientific

stack



What about HEP …?

Community projects towards HEP domain-specific Python tools ⇒ ecosystem

Jake VanderPlas,
The Unexpected Effectiveness of Python in Science,
PyCon 2017

Scikit-HEP project – the grand picture

- ❑ Create an ecosystem for particle physics data analysis in Python
- ❑ Initiative to improve the interoperability between HEP tools and the scientific ecosystem in Python
  - Expand the typical ~~toolkit~~ toolset for particle physicists
  - Set common APIs and definitions to ease "cross-talk"
- ❑ Promote high-standards, well documented and easily installable packages
- ❑ Initiative to build a community of developers and users
  - Community-driven and community-oriented project
  - Open forum to discuss
- ❑ Effort to improve discoverability of (domain-specific) relevant tools

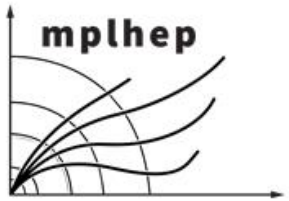
Collaboration

Reproducibility

Interoperability

Sustainability

Scikit-HEP project – overview of most popular and/or used packages



hist

hepunits

histoprint

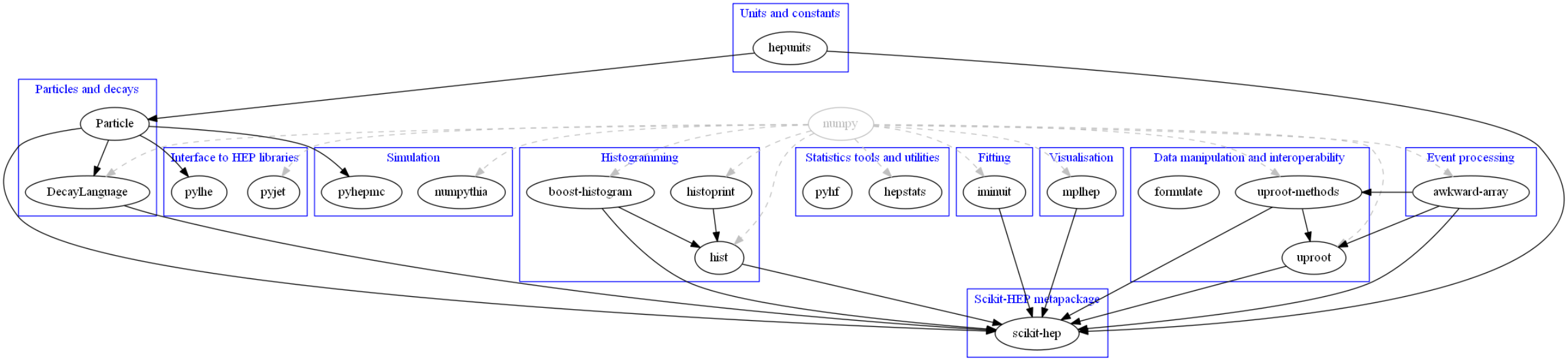
pylhe

[Not the full set of Scikit-HEP packages.]

Scikit-HEP project – overview of (most of the) packages

- Pattern of inter-package dependencies nicely "explains" why the project is a *toolset* and not a toolkit !

https://scikit-hep.org/



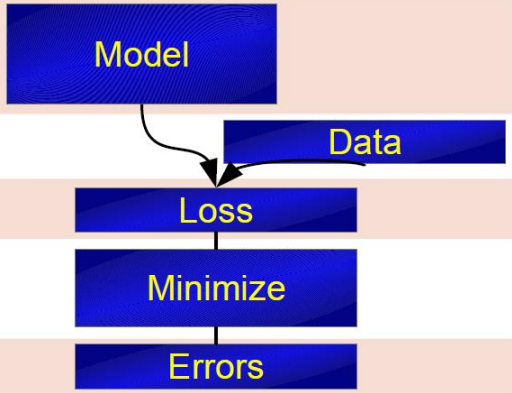
Not a comprehensive list. There are other packages: test data, tutorials, org stats, etc. (and some which tend to now be superseded, hence deprecated …)

The `zfit` project and package

- **Project: provide a stable fitting ecosystem, in close collaboration with the community**
- **`zfit` package:**
  - **Scalable, Pythonic, HEP specific features**
  - **Pure Python, no ROOT dependency, performant (TensorFlow as main backend)**
  - **Highly customisable and extendable**
  - **Depends on iminuit**

- **Simple example:**

```
obs = zfit.Space("x", limits=(-2, 3))

mu = zfit.Parameter("mu", 1.2, -4, 6)
sigma = zfit.Parameter("sigma", 1.3, 0.1, 10)
gauss = zfit.pdf.Gauss(mu=mu, sigma=sigma, obs=obs)

data = zfit.Data.from_numpy(obs=obs, array=normal_np)

nll = zfit.loss.UnbinnedNLL(model=gauss, data=data)

minimizer = zfit.minimize.Minuit()
result = minimizer.minimize(nll)

param_errors = result.error()
```



implement custom function

```
from zfit import ztf


class CustomPDF(zfit.pdf.ZPDF):
    _PARAMS = ['alpha']

    def _unnormalized_pdf(self, x):
        data = x.unstack_x()
        alpha = self.params['alpha']

        return ztf.exp(alpha * data)
```

```
custom_pdf = CustomPDF(obs=obs, alpha=0.2)

integral = custom_pdf.integrate(limits=(-1, 2))
sample   = custom_pdf.sample(n=1000)
prob     = custom_pdf.pdf(sample)
```

The coffea project

Coffea - Column Object Framework for Effective Analysis

Fermilab project to build an analysis framework on top of awkward array and uproot

Separation of "user code" and "executors"

- User writes a Processor to do the analysis
- Executor runs this on different distributed job systems, e.g.:
  - Local multiprocessing, Parsl or Dask (batch systems), Spark cluster

Coffea ***achieved 1 to 3 MHz*** event processing rates

- Using Spark cluster on same site as data at Fermilab

44

Slide taken from Ben Krikler

The FAST-HEP project

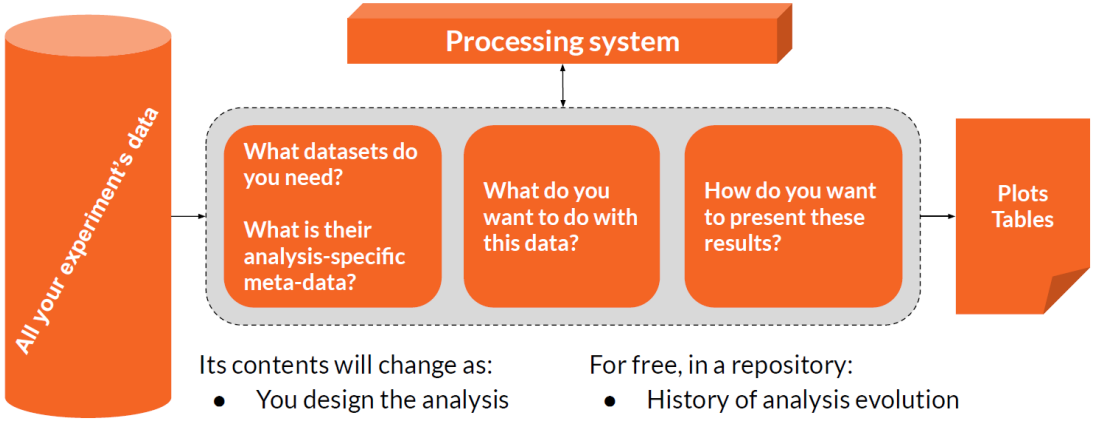
FAST-HEP

Toolkit to help high-level analyses, in particular, within particle physics

http://fast-hep.web.cern.ch fast-hep@cern.ch

- The main product should be the repository
  - Talking about contents – publication is another matter ;-)

Your analysis repository *is* your analysis



Its contents will change as:
- You design the analysis
- You get new / updated data

For free, in a repository:
- History of analysis evolution
- Continuous integration and validation

55

The FAST implementation

For tools:
use Python

NumPy

uproot

Awkward Array

NumExpr

at

For data:
use Pandas
Demoed at CHEP 2018

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

For descriptions:
use YAML...

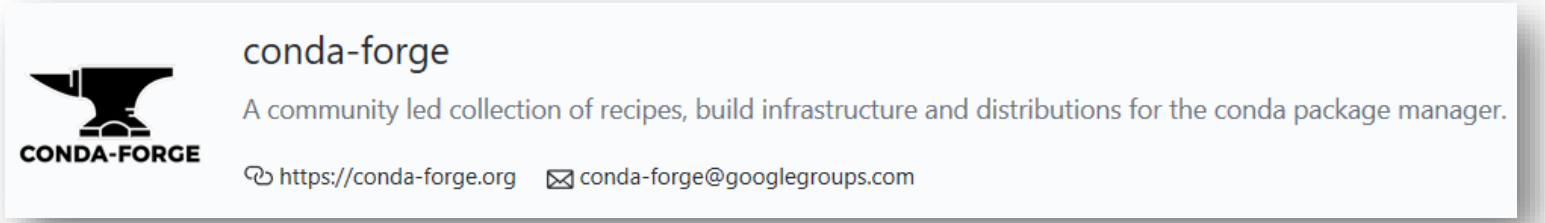
63

- Use a declarative programming approach:
  User sys WHAT, interpretation decides HOW

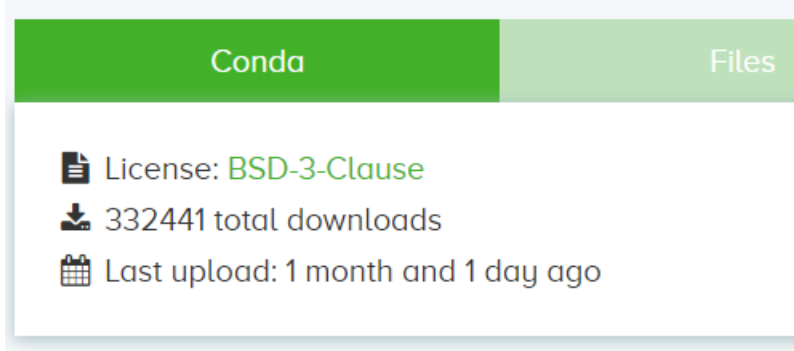
- Project towards an Analysis Description Language …

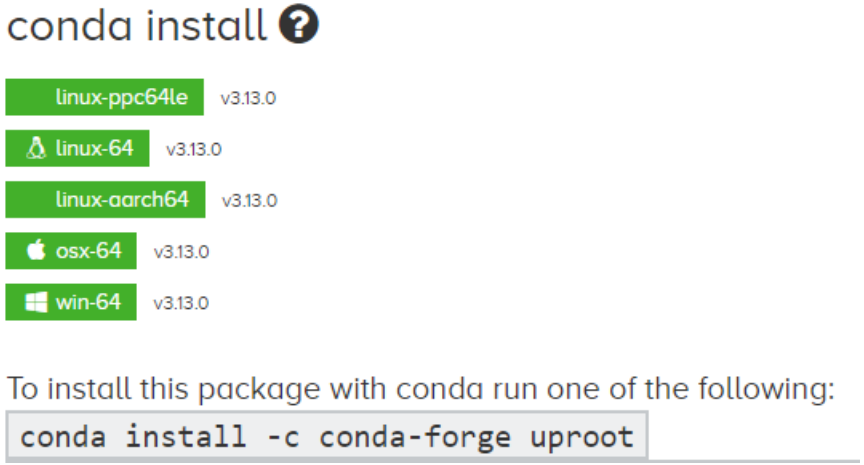
Material taken from Ben Krikler

Conda-forge – making it easy for users



- **Easy / trivial installation in many environments is a must !**
- **Much work has been done in 2019 to provide binary "wheels" on PyPI, and conda-forge packages for many of these new packages**
- **Example of `uproot`:**





PyROOT – focus of recent developments