Analyzing Data with Awkward Arrays

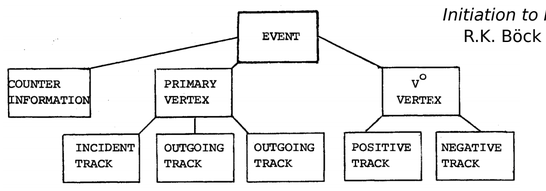
Team: Jim Pivarski and Ianna Osborne
Institutions: Princeton University

Institute for Research & Innovation in Software for High Energy Physics

Why it's needed

Particle physicists have always needed **big datasets** of **nested, variable-sized data**.

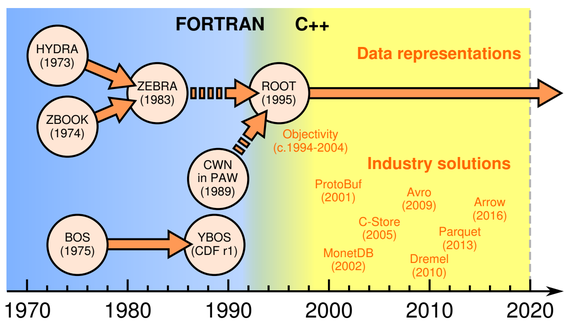
Figure from a 45-year old physics-software manual:



Initiation to HYDRA, R.K. Böck (1974).

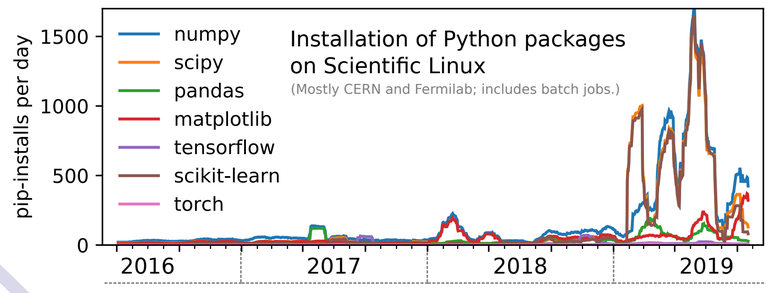
We'd draw similar figures today!

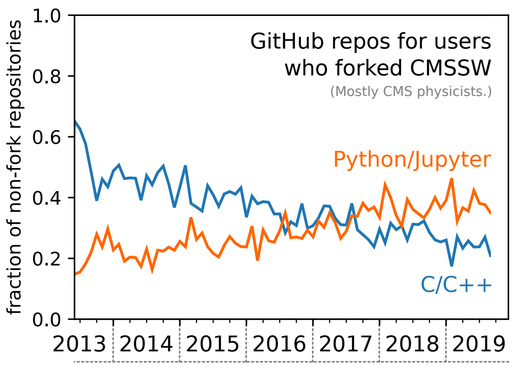
Traditionally, this problem was solved by making data analysts use ~~Fortran~~ C++.



Why now?

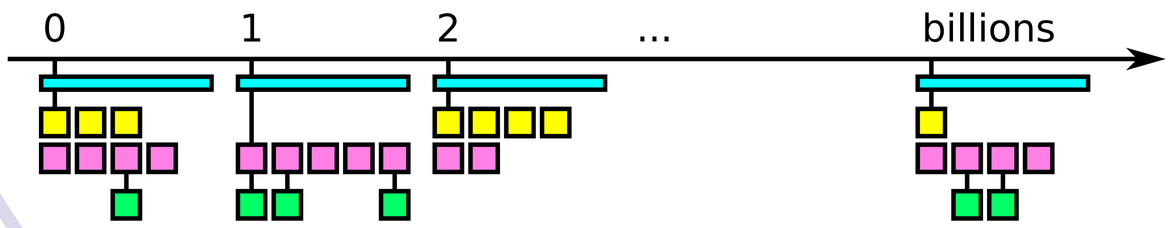
Python/NumPy is rapidly becoming a standard language for data analysis in particle physics.





Awkward Array

An array library for **nested, variable-sized data**, including arbitrary-length lists, records, mixed types, and missing data, using **NumPy-like idioms**.



Arrays are **dynamically typed**, but operations on them are **compiled and fast**.

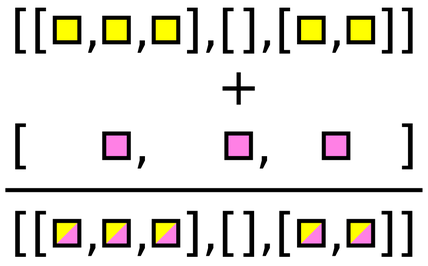
Coincides with NumPy when arrays are regular; generalizes when they're not.

But...

NumPy does not work on nested, variable-sized data!

What it does

Jagged broadcasting of NumPy “ufuncs”

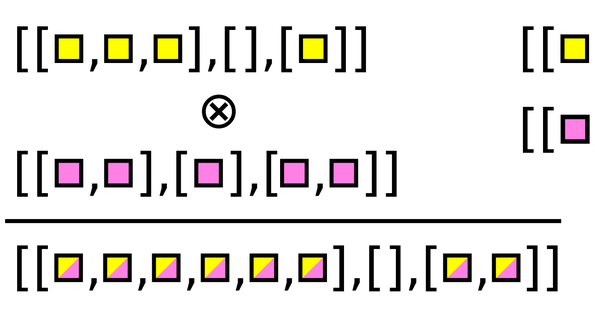


Advanced indexing

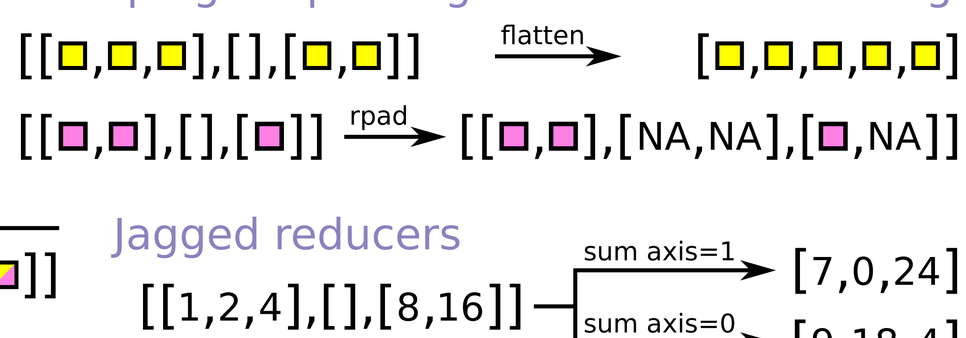
```
# select muons 0,1 from events
>>> events[:, "muons", [0,1]]

# select muons with pt > 50
>>> events[events["muons", "pt"] > 50]
```

Combinatorics



Reshaping for plotting and machine learning



Jagged reducers

ROOT & Arrow/Parquet I/O

Originally intended as an array type for ROOT files, Awkward Arrays are convertable to/from Apache Arrow and Parquet (sometimes zero-copy).

Interface with Numba

Awkward Arrays can be arguments and return values in Numba's JIT-compiled functions, enabling for-loop logic at the speed of compiled code.

...with Pandas

Awkward Arrays can be columns of a Pandas DataFrame.

...NumExpr, Autograd, ...

Jagged broadcasting is applied to all elementwise array calculations.

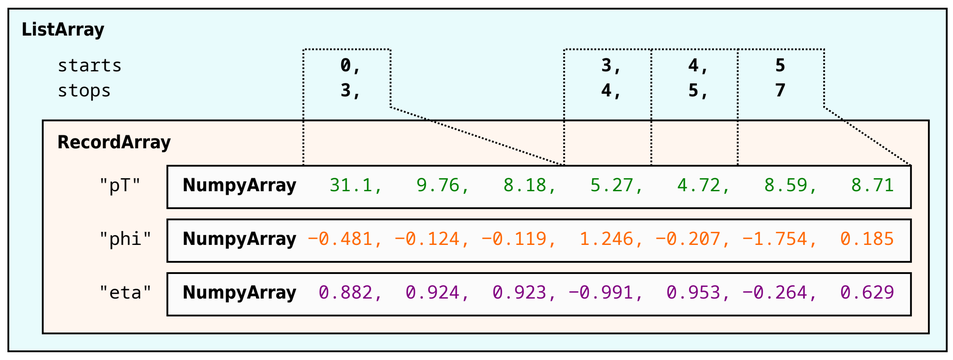
How it works

Arrays and their operations are **columnar**.

Consider these lists of particle objects:

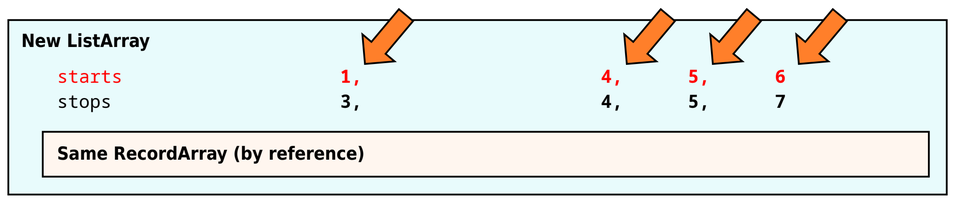
```
[[Muon(31.1, -0.481,  0.882), Muon(9.76, -0.124, 0.924), Muon(8.18, -0.119, 0.923)],
 [Muon(5.27,  1.246, -0.991)],
 [Muon(4.72, -0.207,  0.953)],
 [Muon(8.59, -1.754, -0.264), Muon(8.71,  0.185, 0.629)]]
```

We represent them in columnar arrays, contiguous by field:



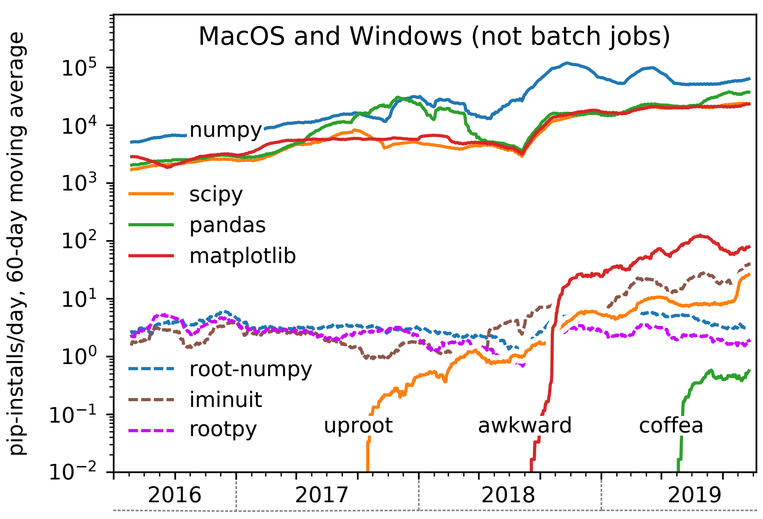
To transform the data, for example to remove the first element from each list, we only need to replace the ListArray:

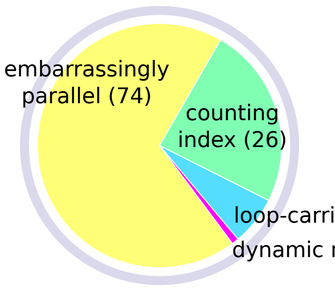
```
[[..., Muon(9.76, -0.124, 0.924), Muon(8.18, -0.119, 0.923)], [...], [...],
 [..., Muon(8.71,  0.185, 0.629)]]
```



Who uses it?

Mostly physicists, but a few geneticists and data scientists have expressed interest.





The library consists of a suite of “kernels” that transform arrays into arrays.

Most are embarrassingly parallel and are good candidates for GPU acceleration.

This project is supported by National Science Foundation under Cooperative Agreement OAC-1836650. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.