

Awkward Arrays: Accelerating scientific data analysis on irregularly shaped data

Framework, #2103945

PI: Jim Pivarski

What Awkward Array does

Example of an "awkward" array

```
array = ak.Array([
    {"x": 1.1, "y": [1]}, {"x": 2.2, "y": [1, 2]}, {"x": 3.3, "y": [1, 2, 3]}, [
    {"x": 4.4, "y": [1, 2, 3, 4]}, {"x": 5.5, "y": [1, 2, 3, 4, 5]}])
```

NumPy-like expression

```
output = np.square(array["y", ..., 1:])
```

Result

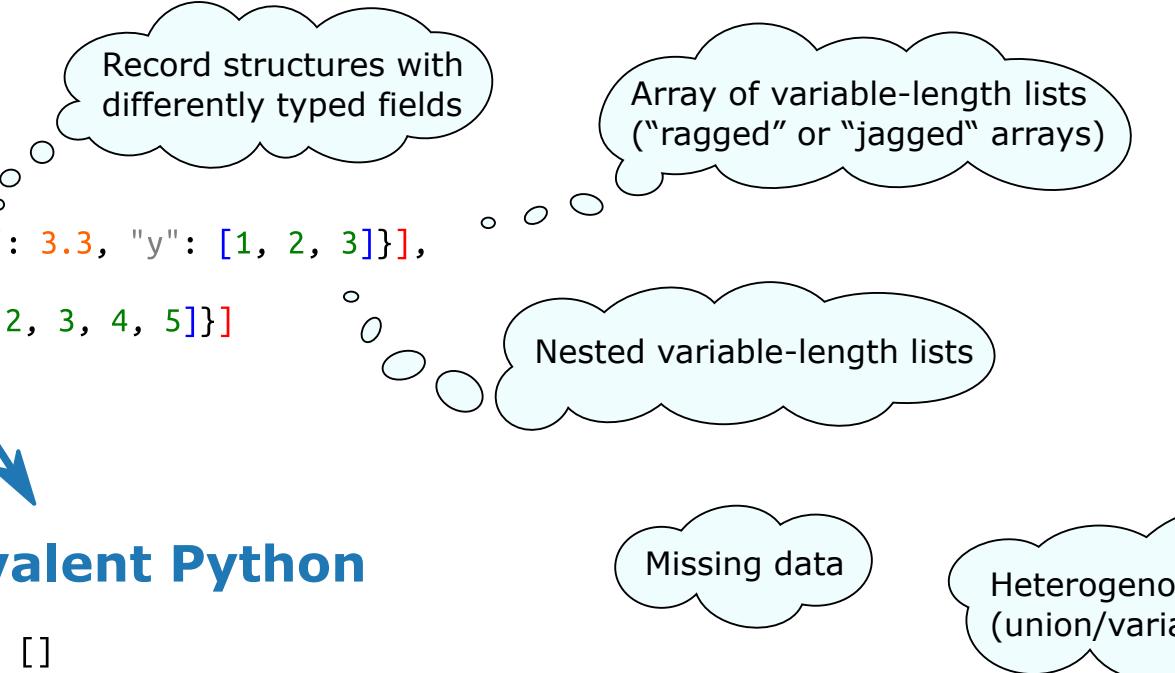
```
output.to_list()
[[[], [4], [4, 9]], [],
 [[4, 9, 16], [4, 9, 16, 25]]]
```

1.5 seconds
2.1 GB of memory

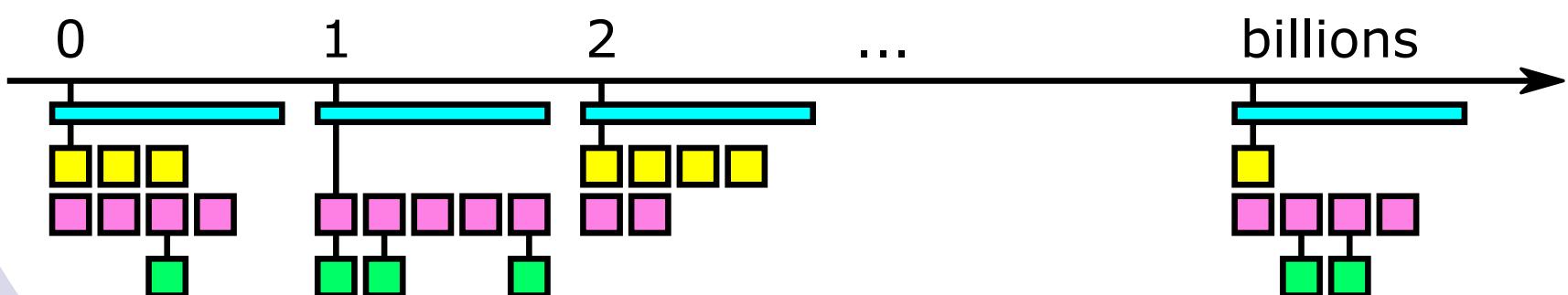
Equivalent Python

```
output = []
for sublist in python_objects:
    tmp1 = []
    for record in sublist:
        tmp2 = []
        for number in record["y"][1:]:
            tmp2.append(np.square(number))
        tmp1.append(tmp2)
    output.append(tmp1)
```

140 seconds
22 GB of memory



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.

Documentation

Instant browser demo

Try it

Get started Try it

Versatile Arrays

Awkward Arrays are general tree-like data structures, like JSON, but contiguous in memory and operate with compiled, vectorized code like NumPy.

They look like arrays:

Like NumPy they can:

ak.Array([1, 2, 3, 4, 5])

Like NumPy they can:

ak.Array([{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [{"x": 1, "y": [{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [1, 2, 3]}, {"x": 2, 3, 4}, {"x": 5, "y": [1, 2, 3, 4, 5]}]}]}]}]}]}])

Like NumPy they can:

ak.Array([{"x": 1, "y": [{"x": 1, "y": [{"x": 1, "y": [{"x": 1, "