YOU DON'T HAVE TO BE AN EXPERT

Stories from the Open Source Frontlines

Alenka Frim

OPEN SOURCE

> … open source software (OSS), a global public good that plays a vital role in the economy and is foundational for most technology we use today …

https://www.hbs.edu/ris/Publication%20Files/24-038_51f8444f-502c-4139-8bf2-56eb4b65c58a.pdf

> Open source fosters a collective approach to problem-solving.

https://www.turing.ac.uk/blog/open-source-software-why-it-matters-and-how-get-involved

> Developing and deploying open source software is no longer just a novel idea. It’s a strategic necessity in a fast-changing digital world.

https://www.bcg.com/publications/2021/open-source-software-strategy-benefits

KARAOKE

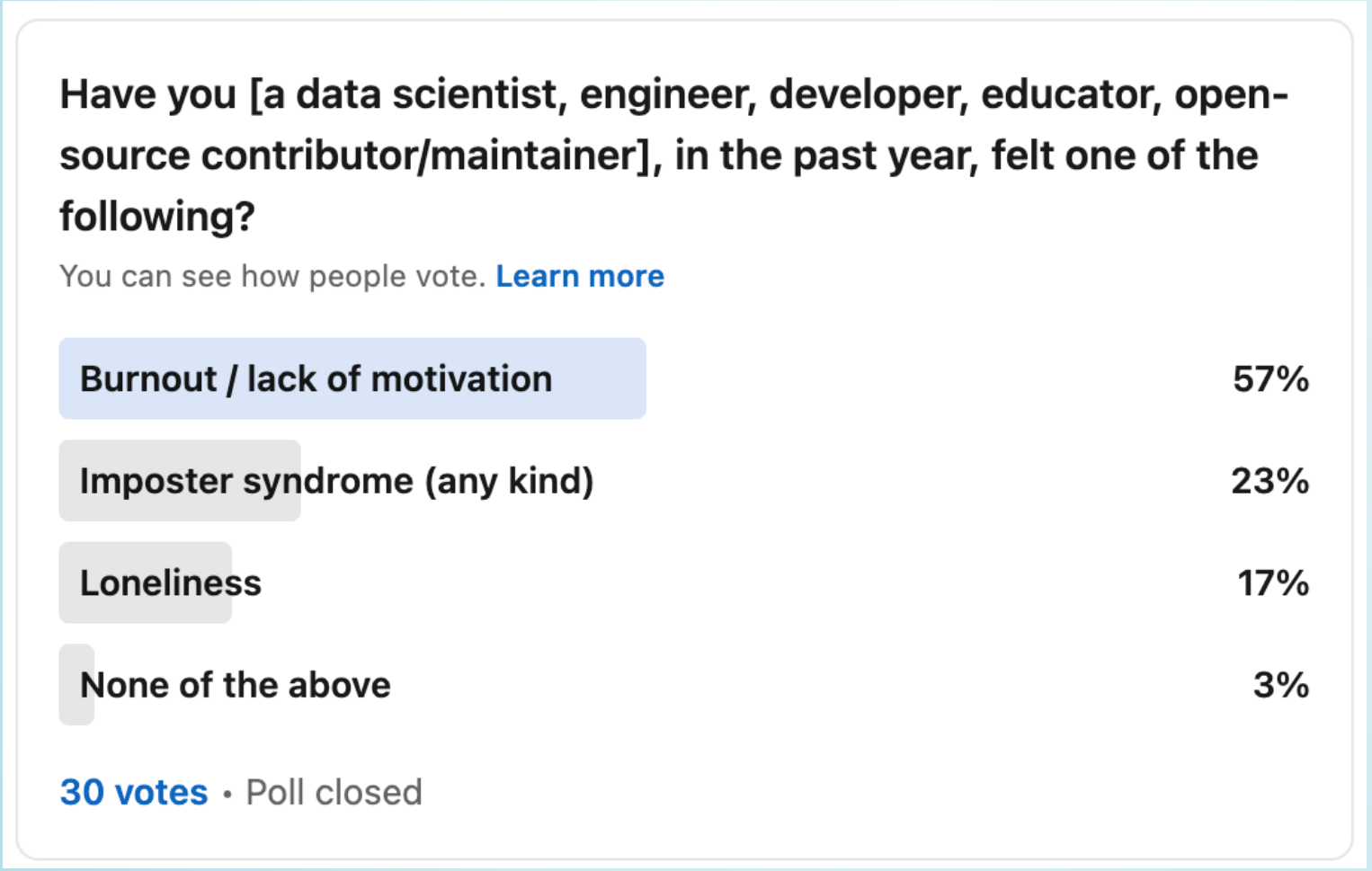
CAN THE DATA ECOSYSTEM BE OVERWHELMING?

Do you remember a time when you ...

... felt you were not smart enough?

... you were afraid of being wrong?

CAN THE DATA ECOSYSTEM BE OVERWHELMING?



DR. CAT HICKS

Psychologist for Software Teams

Developer Thriving

Learning Culture

> People are always looking for cues to weather or not they are in a safe place for learning

BUILDING PLAYFULNESS

TYPE CHECKING SUPPORT

- **Rok Mihevc** wanted to find a solution for the Type Checking support in PyArrow
- took the opportunity to brainstorm ideas on EuroPython sprints

- Worked towards POC with **Patrick J. Roddy**

TYPE ANNOTATIONS IN PYARROW

[Python] Gradually add type checks to Arrow, initial step #45

Open **rok** wants to merge 10 commits into `main` from `gradual_pyarrow_stubs`

Conversation 11 | Commits 10 | Checks 24 | Files changed 18

rok commented last month • edited Owner

This proposes adding type annotation to pyarrow by adopting pyarrow-stubs into pyarrow. To do so we copy a subset of pyarrow-subs's stubfiles into pyarrow tree. We then add annotation checks for some stubsfiles and some test files and make sure checks pass. Annotation checks should be expanded until all (or most) project files are covered in future work.

PR introduces:

1. adds pyarrow-stubs into `arrow/python/pyarrow/`
2. fixes pyarrow-stubs to pass mypy and pyright check
3. adds mypy and pyright check to CI (crudely)
4. adds a tool (update_stub_docstrings.py) to keep annotation docstrings in sync with source docstrings
5. adds docstring sync check to CI (crudely)

🎉 1 ❤️ 1 🚀 1

TYPE ANNOTATIONS IN PYARROW

```
import pyarrow.compute as pc


pc.
```

```
↑ f all(array, skip_nulls, min_count, opt…  pyarrow.compute
↓ f scalar(value)                           pyarrow.compute
↑ f array_take(array, indices, boundschec…  pyarrow.compute
↑ f sum(array, skip_nulls, min_count, opt…  pyarrow.compute
↑ f array_filter(array, selection_filter,…  pyarrow.compute
↓ f cast(arr, target_type, safe, options,…  pyarrow.compute
↓ f abs(x, memory_pool)                     pyarrow.compute
↓ v abs_checked                             pyarrow.compute
```

TYPE ANNOTATIONS IN PYARROW

```
import pyarrow.compute as pc
pc.strftime()
```

pyarrow.compute

```
def strftime(timestamps: Date32Scalar | Date64Scalar | Time32Scalar[Any]
TimestampScalar[Any] | DurationScalar[Any] | MonthDayNanoIntervalScalar
Date64Scalar | Time32Scalar[Any] | Time64Scalar[Any] | TimestampScalar[A
MonthDayNanoIntervalScalar] | ChunkedArray[Date32Scalar | Date64Scalar |
[Any] | TimestampScalar[Any] | DurationScalar[Any] | MonthDayNanoInterva
            /,
            format: str = "%Y-%m-%dT%H:%M:%S",
            locale: str = "C",
            *,
            options: StrftimeOptions | None = None,
            memory_pool: MemoryPool | None = None) -> StringScalar | St
```

Format temporal values according to a format string.

For each input value, emit a formatted string. The time format string and locale can be set usin
the "%S" (seconds) format code depends on the input time precision: it is an integer for timesta
with the required number of fractional digits for higher precisions. Null values emit null. An erro
timezone but it cannot be found in the timezone database, or if the specified locale does not ex

Params: `timestamps` – Argument to compute function.

`format` – Pattern for formatting input values

APACHE ARROW

Would I be any good at software development in Open source?

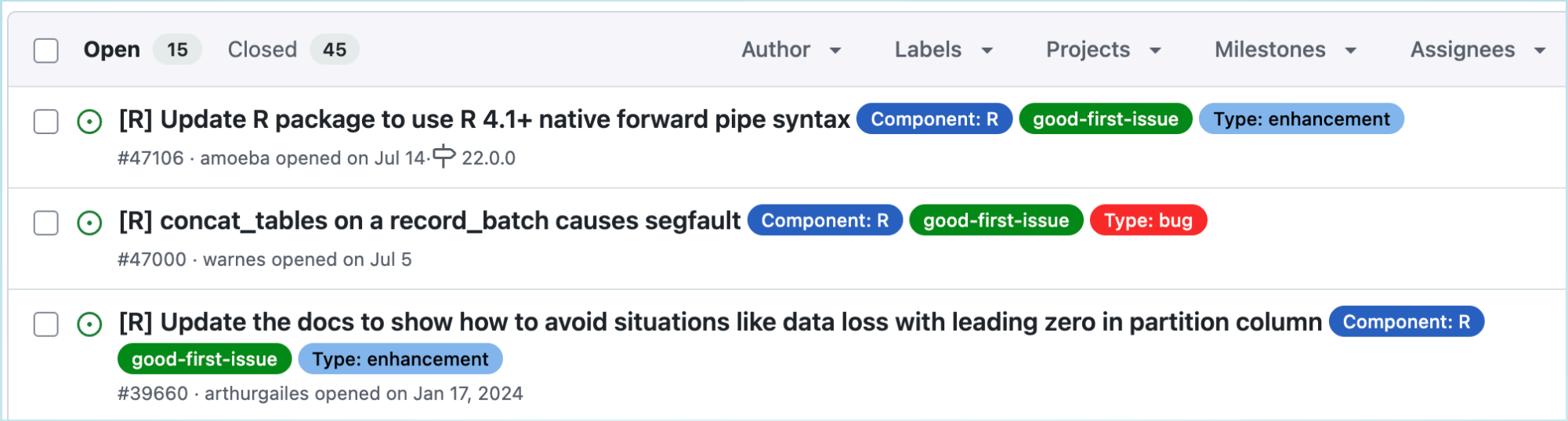
Official Implementations

The Apache Arrow project houses a collection of libraries for different programming languages. Use the links in the table below to access the documentation and source code for these libraries.

Language	Docs	Source
C++	C++ Docs	C++ Source
C GLib	C GLib Docs	C GLib Source
C#	C# Docs	C# Source
Go	Go Docs	Go Source
Java	Java Docs	Java Source
JavaScript	JavaScript Docs	JavaScript Source
Julia	Julia Docs	Julia Source
MATLAB	MATLAB Docs	MATLAB Source
Python	Python Docs	Python Source
R	R Docs	R Source
Ruby	Ruby Docs	Ruby Source
Rust	Rust Docs	Rust Source
Swift	Swift Docs	Swift Source

ARROW R PACKAGE

Lots of good first issues!



Lots of support from the maintainers!

apache/arrow **ARROW-13137: [C++][Documentation] Make in-table references consistent** 5

#10630 by AlenkaF was closed on Jul 5, 2021 • Approved

apache/arrow **ARROW-12867: [R] Bindings for abs()** Component: R 24

#10519 by AlenkaF was closed on Jun 21, 2021 • Changes requested

apache/arrow **ARROW-12198: [R] bindings for strptime** Component: R 46

#10334 by AlenkaF was closed on Jun 4, 2021 • Approved

HELPED ME BUILD MY COMMUNITY







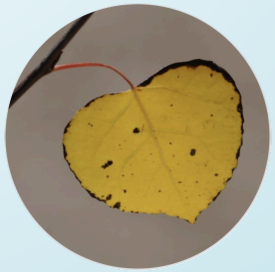













BUILDING CURIOSITY

APACHE ARROW FORMAT

> Apache Arrow defines a language-independent columnar memory format for flat and nested data, organized for efficient analytic operations on modern hardware like CPUs and GPUs.

APACHE ARROW FORMAT

> The Arrow memory format also supports zero-copy reads for lightning-fast data access without serialization overhead.

APACHE ARROW FORMAT

Apache Arrow defines a language-independent columnar memory format for flat and nested data, organized for efficient analytic operations on modern hardware like CPUs and GPUs.

The Arrow memory format also supports zero-copy reads for lightning-fast data access without serialization overhead.

YOGIC APHORISMS

> Yogaś citta vṛtti nirodhaḥ

Chitta is the consciousness which includes the mind, the intellect and the ego. Yoga is a method of silencing the vibrations of the chitta.

~ https://bksiyengar.com/modules/iyoga/iyoga.htm

Single reading

vs

re-reading
reading multiple texts

⭐ DOCUMENTATION IN OPEN SOURCE

- do the documentation!
- elevate documentation skills
- don’t look down on documentation work!

⭐ DOCUMENTATION IN OPEN SOURCE

- 💡 aphorism structure

  Aphorism → Translation → Commentary

PyArrow Tables

Instances of `pyarrow.Table`, a logical table data structure in which each column consists of one or more `pyarrow.Array` objects of the same type.

http://arrow.apache.org/docs/python/data.html#tables

BUILDING BRAVERY

IMPACT OF OS

and open standards!

⭐ TABULAR DATA STRUCTURES

Python

- select a dataframe library to work with tabular data

R

- built in columnar tabular data structure called `data.frame`

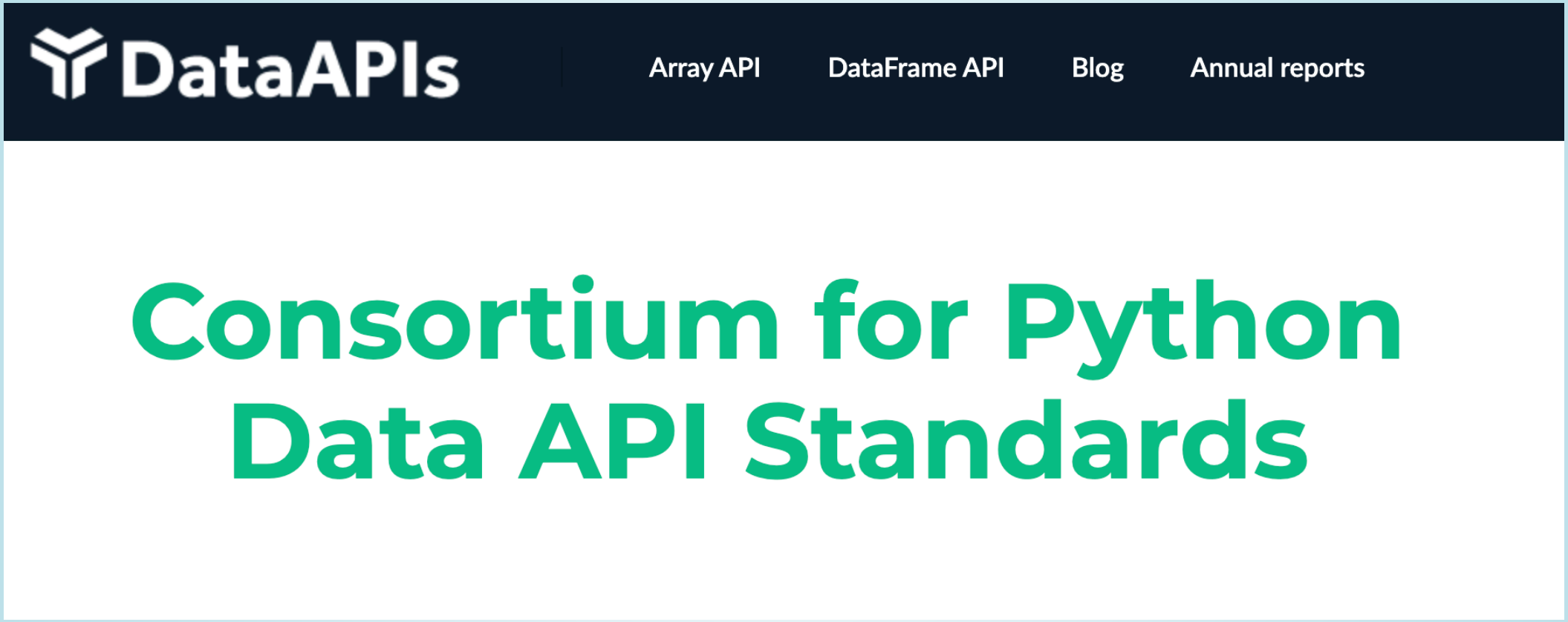
> Fragmentation of the Python dafatframe libraries.

DATAFRAME INTERCHANGE PROTOCOL

In 2021 **Ralf Gommers** from Quansight
took on the task to:

- facilitate the discussion
- to construct a **dataframe interchange protocol**
- allow converting one type of dataframe into another

DATAFRAME INTERCHANGE PROTOCOL



Consortium includes representatives from both open source Python libraries and the industry.

LIBRARY INSTEAD OF A PROTOCOL

Marco Gorelli

Narwhals, the dataframe “translator”

> compatibility layer between Python dataframe libraries

ADOPTING A WIDER STANDARD

Joris Van den Bossche and **Will Jones**

- wanted to use the Apache Arrow columnar format
- de facto standard
- all about zero-copy

Enable data interchange without the need to depend on PyArrow.

ARROW PYCAPSULE PROTOCOL

C data interface

Implemented in 2020 by **Antoine Pitrou**

Inspired by the Python buffer protocol.

Later additions: C stream interface and C device interface

Capsules

part of Python C API

Envelopes for the C Data Interface structs

More content on the Arrow PyCapsule Protocol:

The expanding Apache Arrow universe

Joris Van den Bossche | PyData Paris 2024

EXAMPLE BEFORE

```
df_polars = pl.DataFrame({"a": [1,2], "b": [0.1, 0.2]})

# Convert polars do datafusion
ctx = datafusion.SessionContext()
df_datafusion = ctx.from_arrow(df_polars.to_arrow())

# and convert back to polars
pl.DataFrame(df_datafusion.to_arrow_table())
```

EXAMPLE AFTER

```
df_polars = pl.DataFrame({"a": [1,2], "b": [0.1, 0.2]})

# Convert polars do datafusion
ctx = datafusion.SessionContext()
df_datafusion = ctx.from_arrow(df_polars)

# and convert back to polars
pl.DataFrame(df_datafusion)
```

COMMUNITY BENEFITS

Open **[Python] Promote usage of the Arrow PyCapsule Protocol (for the C Data Inteface)** #39195

kylebarron on Jul 23, 2024 · edited by kylebarron Edits Contributor ···

I've been working a bit to promote the protocol; here's a running tally:

implemented: (at least partially, return objects with pycapsule protocol and/or check for existence of protocol in constructors)

- pandas: ENH: support the Arrow PyCapsule Interface on pandas.DataFrame (export) pandas-dev/pandas#56587, ENH: support the Arrow PyCapsule Interface on pandas.Series (export) pandas-dev/pandas#59518, ENH: support Arrow PyCapsule Interface on Series for export pandas-dev/pandas#59587, ENH: add basic DataFrame.from_arrow class method for importing through Arrow PyCapsule interface pandas-dev/pandas#59696
- ibis: feat(pyarrow): support Arrow PyCapsule interface on `ibis.Table` objects ibis-project/ibis#9143, feat(pyarrow): support `__arrow_c_schema__` on `ibis.Schema` objects ibis-project/ibis#9665
- pyarrow
- nanoarrow
- arro3
- lonboard
- pyogrio
- gdal Python bindings: add a ogr.Layer.WriteArrow() method consuming __arrow_c_stream__ or __arrow_c_array__ interfaces OSGeo/gdal#9133, Python bindings: implement __arrow_c_stream__() interface for ogr.Layer OSGeo/gdal#9043

THANK YOU!

GitHub profile

Slides