Deconstructing Feather

Bill Lattner (@wlattner)
PyData Chicago, 2016

Building a Data-Driven World™

Why?

- Exchange tabular data between Python, R, and others
- Fast read/write
- Represent categorical features
- *It's about the metadata*[1]

1. http://wesmckinney.com/blog/feather-its-the-metadata/

CSV Files

com·plex·i·ty

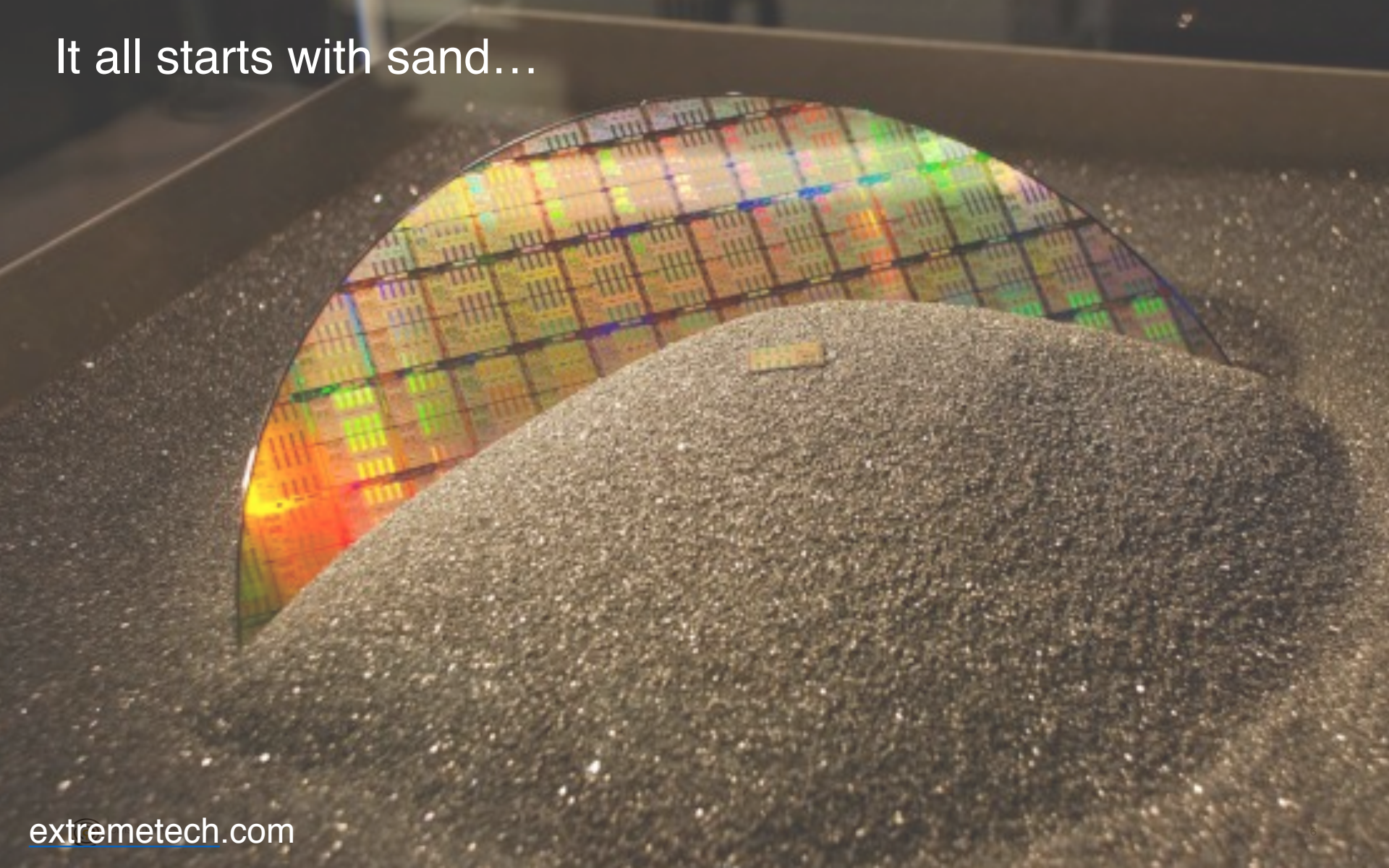
/kəmˈpleksədē/

noun

```
pandas.read_csv(filepath_or_buffer, sep=', ', delimiter=None, header='infer',
                names=None, index_col=None, usecols=None, squeeze=False, prefix=None,
                mangle_dupe_cols=True, dtype=None, engine=None, converters=None,
                true_values=None, false_values=None, skipinitialspace=False,
                skiprows=None, skipfooter=None, nrows=None, na_values=None,
                keep_default_na=True, na_filter=True, verbose=False,
                skip_blank_lines=True, parse_dates=False, infer_datetime_format=False,
                keep_date_col=False, date_parser=None, dayfirst=False, iterator=False,
                chunksize=None, compression='infer', thousands=None, decimal='.',
                lineterminator=None, quotechar='"', quoting=0, escapechar=None,
                comment=None, encoding=None, dialect=None, tupleize_cols=False,
                error_bad_lines=True, warn_bad_lines=True, skip_footer=0,
                doublequote=True, delim_whitespace=False, as_recarray=False,
                compact_ints=False, use_unsigned=False, low_memory=True,
                buffer_lines=None, memory_map=False, float_precision=None)
```

Computers!

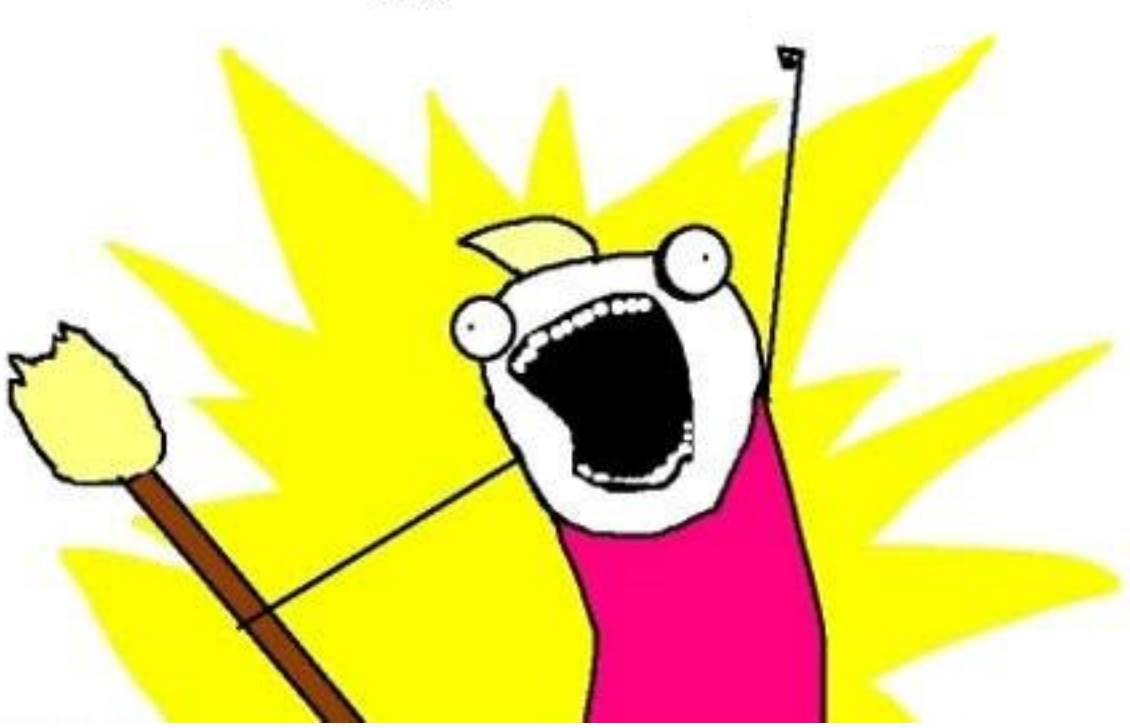
It all starts with sand…



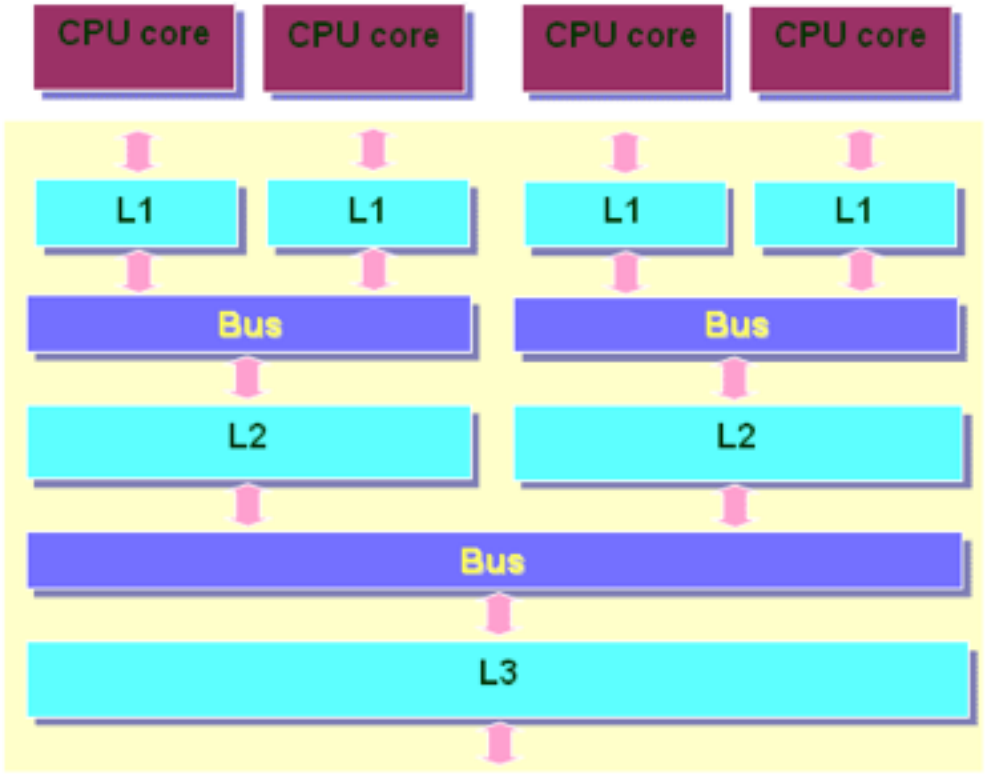
extremetech.com

How we think they work

O(1) all the memory access



How they actually work



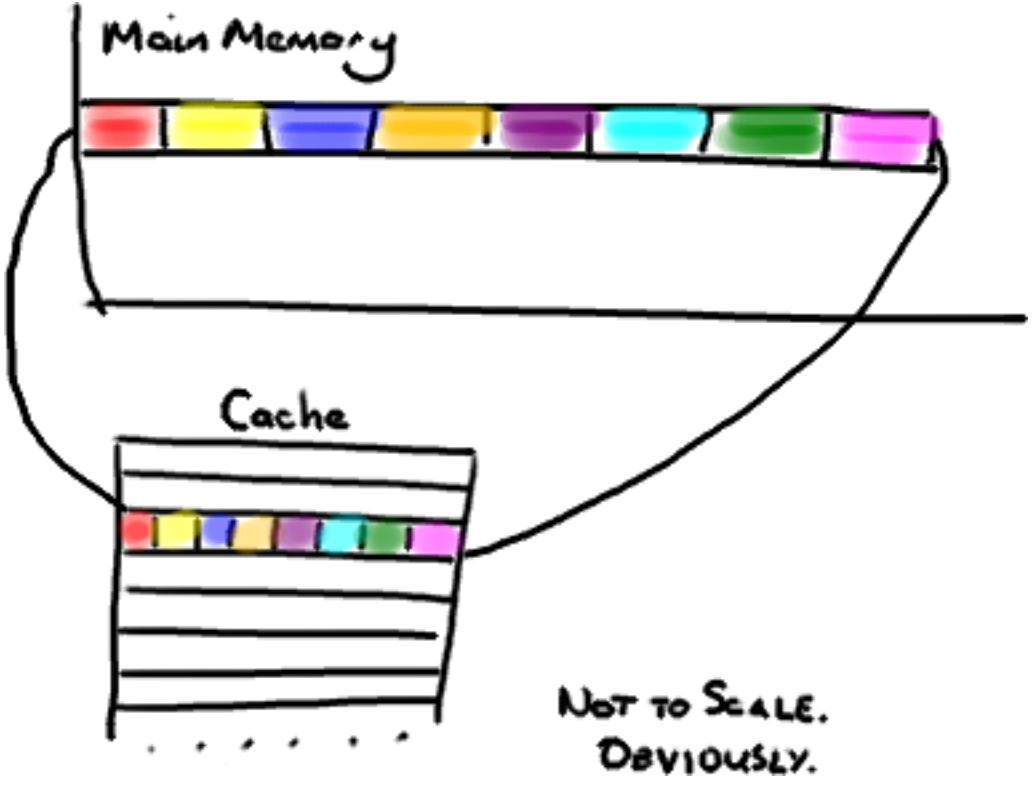
https://software.intel.com/sites/default/files/m/d/4/1/d/8/196578_196578.gif

How they actually work

L1 cache reference	0.5 ns
Branch mispredict	5 ns
L2 cache reference	7 ns
Mutex lock/unlock	100 ns
Main memory reference	100 ns
Compress 1K bytes with Zippy	10,000 ns
Send 2K bytes over 1 Gbps network	20,000 ns
Read 1 MB sequentially from memory	250,000 ns
Round trip within same datacenter	500,000 ns
Disk seek	10,000,000 ns
Read 1 MB sequentially from network	10,000,000 ns
Read 1 MB sequentially from disk	30,000,000 ns
Send packet CA->Netherlands->CA	150,000,000 ns

http://static.googleusercontent.com/media/research.google.com/en//people/jeff/stanford-295-talk.pdf

How they actually work



http://mechanitis.blogspot.com/2011/07/dissecting-disruptor-why-its-so-fast_22.html

What this means

Data layout needs to be tailored for expected read/write operations.

Memory access cost (latency) depends on location ***and*** predictability.

Sequential access FTW!!!

Feather (https://github.com/wesm/feather)

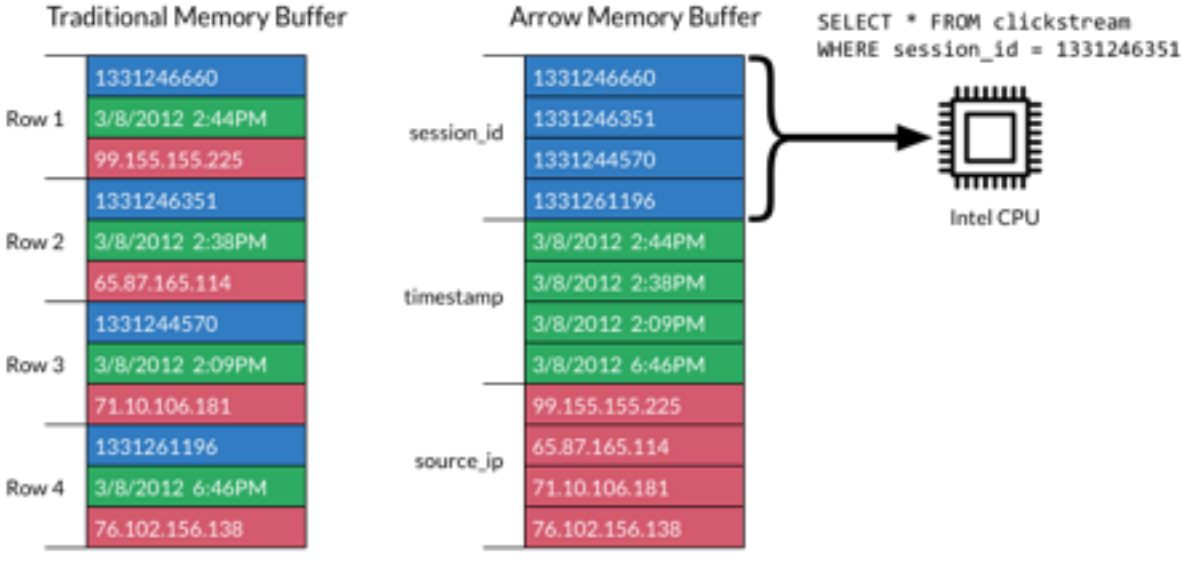
The idea

On disk representation of tabular data should be similar to the in memory representation.

Columnar layout is a good fit for analytic workflows.

The idea

	session_id	timestamp	source_ip
Row 1	1331246660	3/8/2012 2:44PM	99.155.155.225
Row 2	1331246351	3/8/2012 2:38PM	65.87.165.114
Row 3	1331244570	3/8/2012 2:09PM	71.10.106.181
Row 4	1331261196	3/8/2012 6:46PM	76.102.156.138



https://arrow.apache.org/

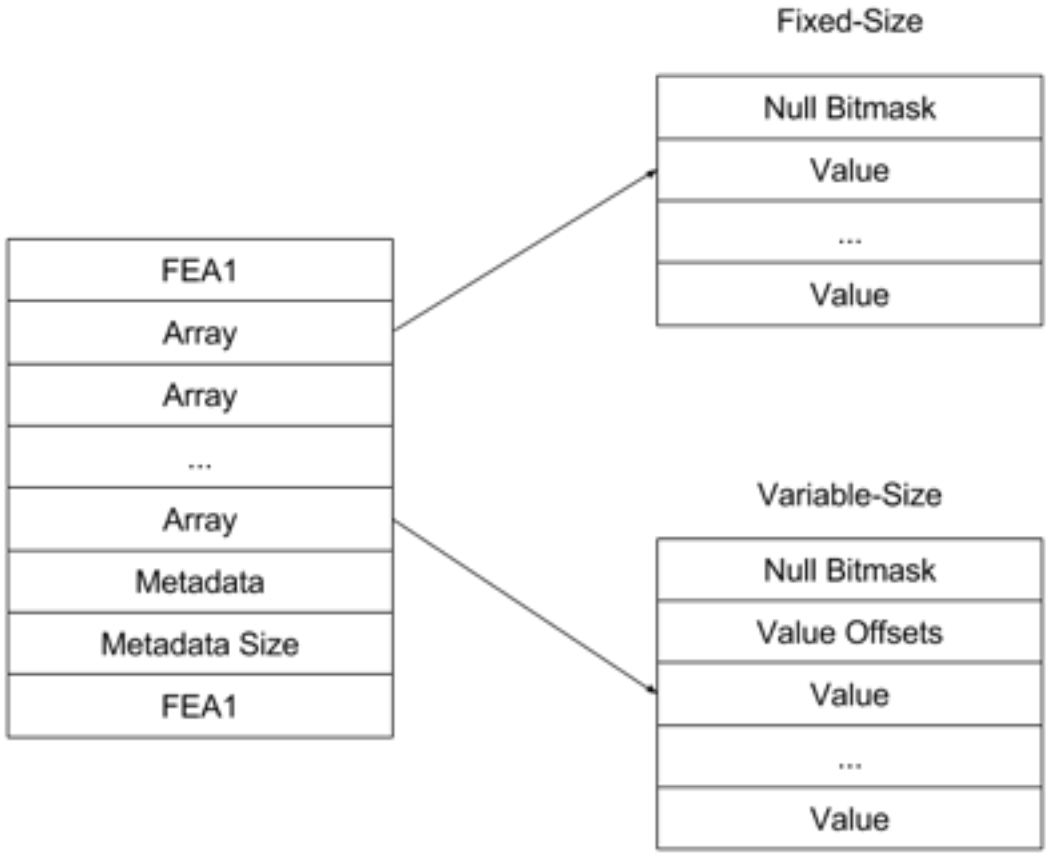
sim·plic·i·ty

/simˈplisədē/

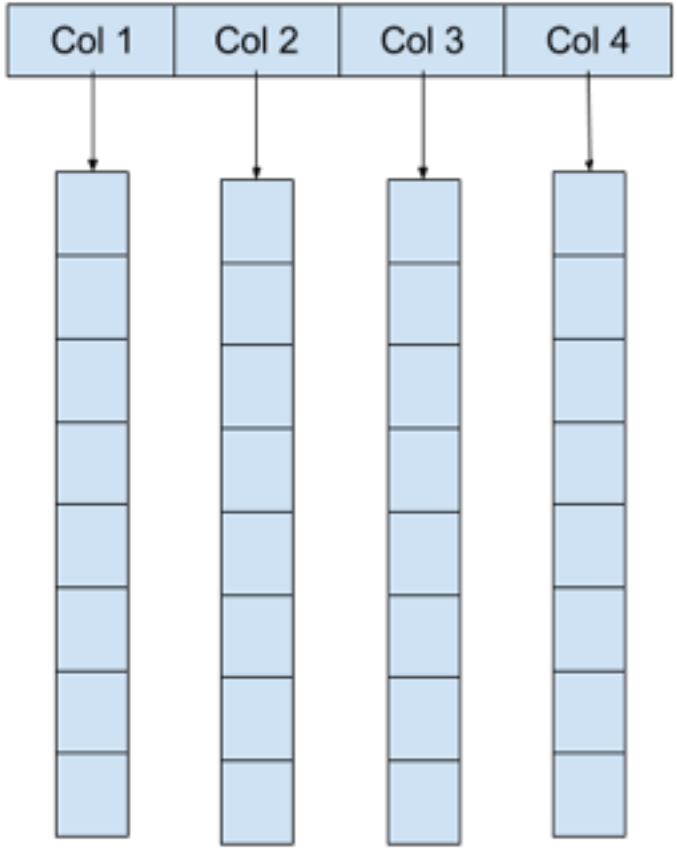
noun

```
1 feather.read_dataframe(path, columns=None)
```

The details


Fixed-Size
Null Bitmask
Value
...
Value
FEA1
Array
Array
...
Array
Metadata
Metadata Size
FEA1
Variable-Size
Null Bitmask
Value Offsets
Value
...
Value

Compare to a dataframe in R



Live Code

The future

Zero-Copy

- In-place operations
- Share operational code between languages
- Zero parsing or copying to Pandas memory representation, mmap the feather file

De facto interchange format

- input to tools like Scikit-Learn or StatsModels
- output from like PostgreSQL

Thanks