Pandas 2.0 – where next?

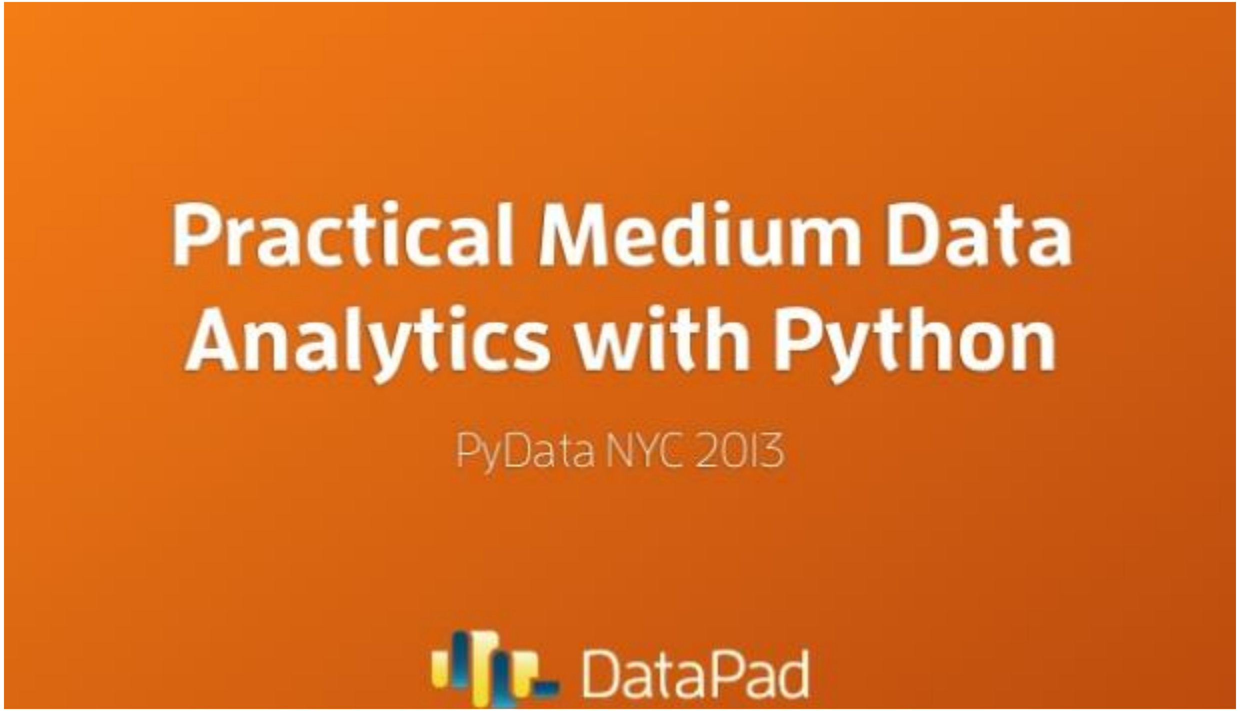
PyCon UK tutorial, 27 October 2017

Stephen Simmons

mail@stevesimmons.com
stephen.e.simmons@jpmorgan.com

https://github.com/stevesimmons/pyconuk-2017-pandas-and-dask

Wes McKinney’s talk at PyData NYC 2013



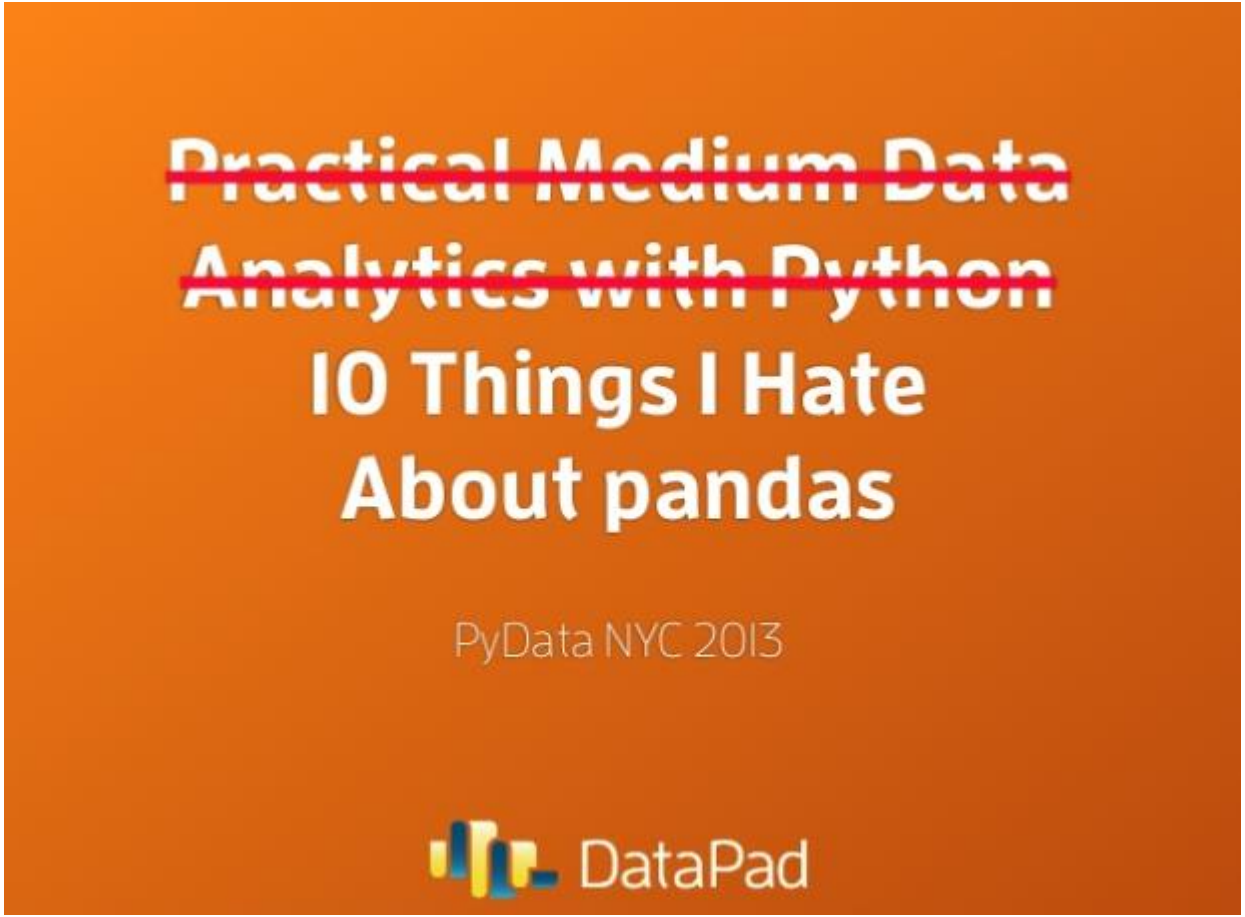
Practical Medium Data Analytics with Python

PyData NYC 2013

DataPad

Source: https://www.slideshare.net/wesm/practical-medium-data-analytics-with-python

Wes McKinney’s talk at PyData NYC 2013



~~Practical Medium Data Analytics with Python~~

10 Things I Hate About pandas

PyData NYC 2013

DataPad

“10 Things I Hate About Pandas” (Wes McKinney, 2013)

1. Internals too far from "the metal"
2. No support for memory-mapped datasets
3. Poor performance in database and file ingest / export
4. Warty missing data support
5. Lack of transparency into memory use, RAM management
6. Weak support for categorical data
7. Complex groupby operations awkward and slow
8. Appending data to a DataFrame tedious and very costly
9. Limited, non-extensible type metadata
10. Eager evaluation model, no query planning
11. "Slow", limited multicore algorithms for large datasets

Source: http://wesmckinney.com/blog/apache-arrow-pandas-internals/

Way forward: Apache Arrow + “libpandas”

Apache Arrow

- Columnar in-memory data format
- Fast: C++, cache-locality, zero-copy reads
- Aim is same memory format for Pandas, Spark, HBase, Parquet, …

v 0.7.1, Oct 2017

libpandas

- Use Arrow for in-memory data (no BlockManager)
- More predictable memory usage
- Simpler core API

Design phase

“Deferred” pandas API

- Expressions API for building calculation frameworks
- Use for out-of-core and distributed execution (using simplified Dask?)

For the future