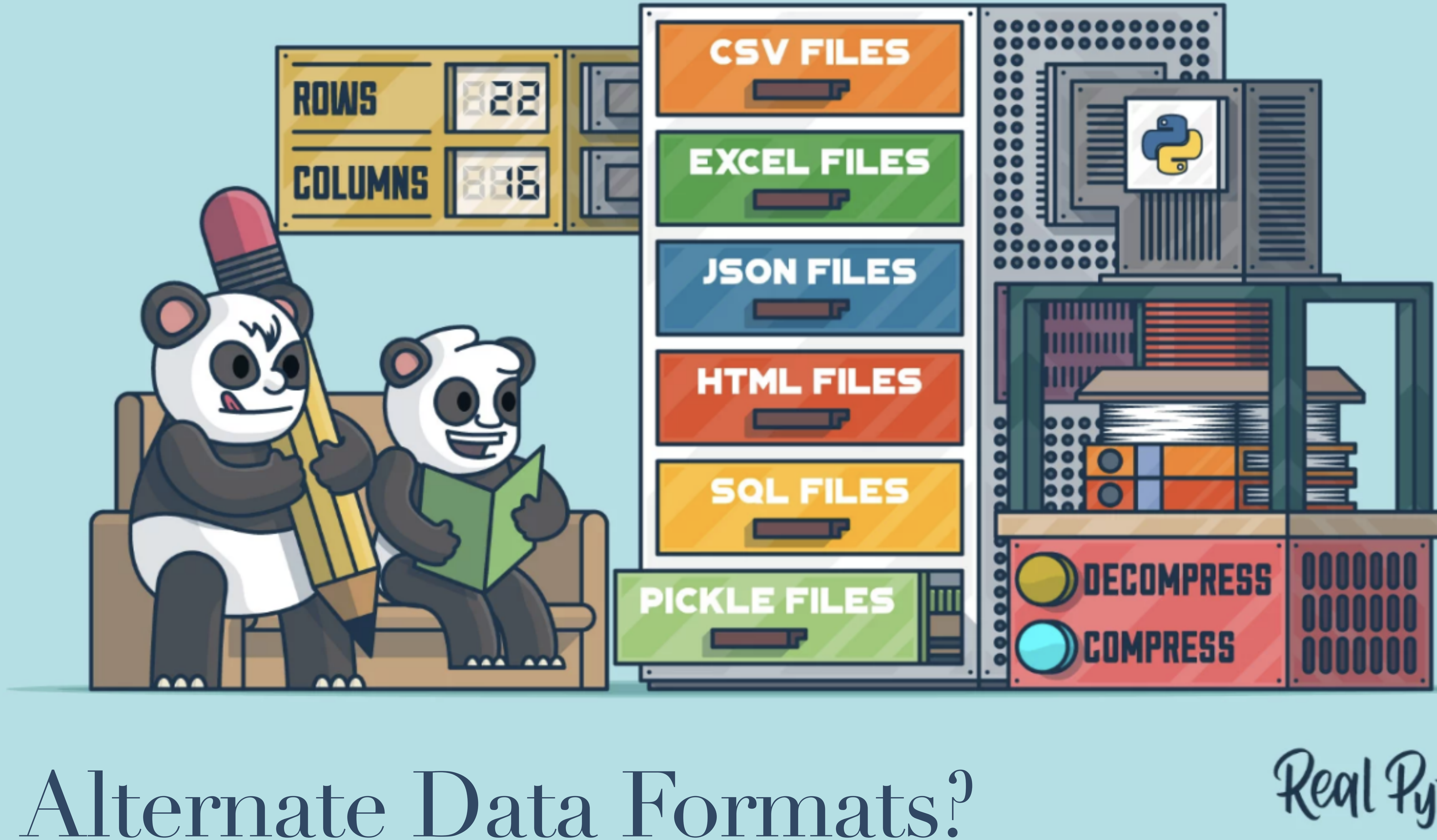


Alternate Data Formats?

HPS Software Meeting

July 15, 2020

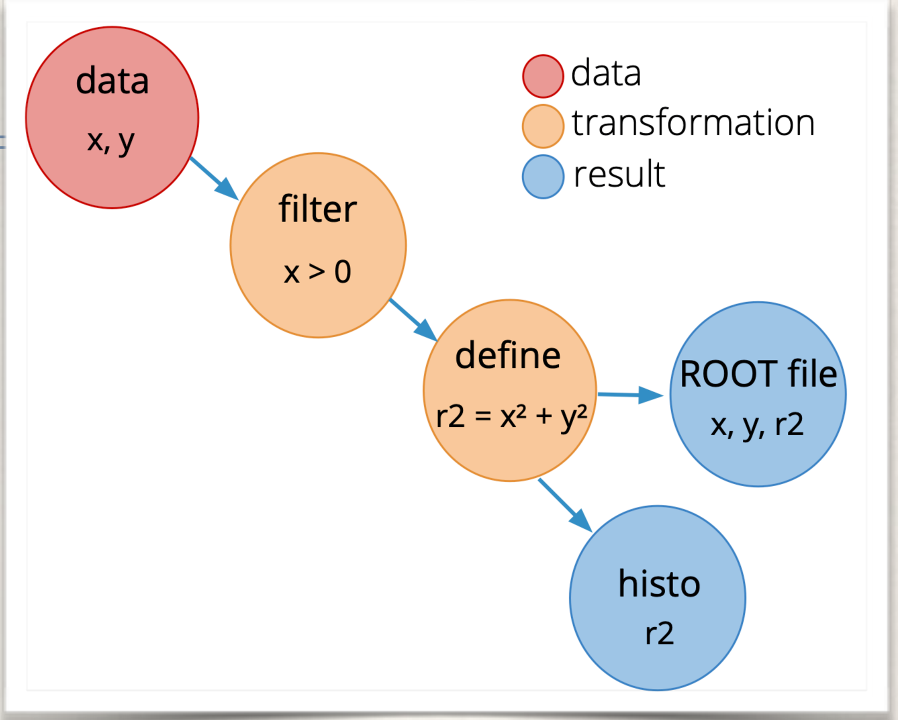
Simple Column Based Formats

- Instead of writing data as serialized class structures, write the data as arrays of primitives.
  - Each event contains named:
    - primitives - run number, event number, …
    - lists - particle_energy, particle_type, ecal_cluster_energy, …
    - lists of lists - particle_indexes_to_tracks, track_covmatrix, …
  - Minimally needed:
    - int, double, vector<int>, vector<double>, vector<vector<int> >, vector<vector<double> >
- Examples of simple column based data formats:
  - PAW's n-tuples, Sho's "tuple"
  - Python: Pandas Data Frames.
  - ROOT: RDataFrame
    - Works with any format TTree, but is *A LOT* easier with a simple column based format.
  - CLAS12: HIPO

Pro/Con of simple formats

- **PRO:**
  - It becomes very easy to add or drop some of the data. Just add or drop the column.
    - Existing code does not break, unless you drop a column it needed.
  - Most implementations of column based data sets are very fast.
    - Only read the actual data you need, not the whole class.
  - Very easy to access the information.
- **CON:**
  - Data is less organized, depending entirely on intelligent naming of the columns.
  - References are index based, so care must be taken that the referenced data does not change order.

ROOT - RDataFrames



- Transaction based data analysis.
- Advertised as: "modern, high-level, type-safe, parallel"
  - Scales well to multi-core processing.
- Works with C++ and/or Python.
  - Admittedly, the Python will likely be a mixed Python and C++.
- Works well with simple data formats.
  - Can work with complicated class structured TTree, but is more difficult.
    - Does not seem to work at all with TRef or TRefArray. (?)
- This seems to be where the ROOT analysis platform is going.
- see: https://root.cern/doc/master/classROOT_1_1RDataFrame.html