cloudera®

DataFrames: The Good, Bad, and Ugly

Wes McKinney @wesmckinn

NY R Conference, 2015-04-25

Disclaimer: the views presented in this talk are my personal opinions and not necessarily those of Cloudera

This talk

- Some commentary on all the data frame interfaces out there
- Biased observations and cursory judgments
- Thoughts on crafting high quality data tools

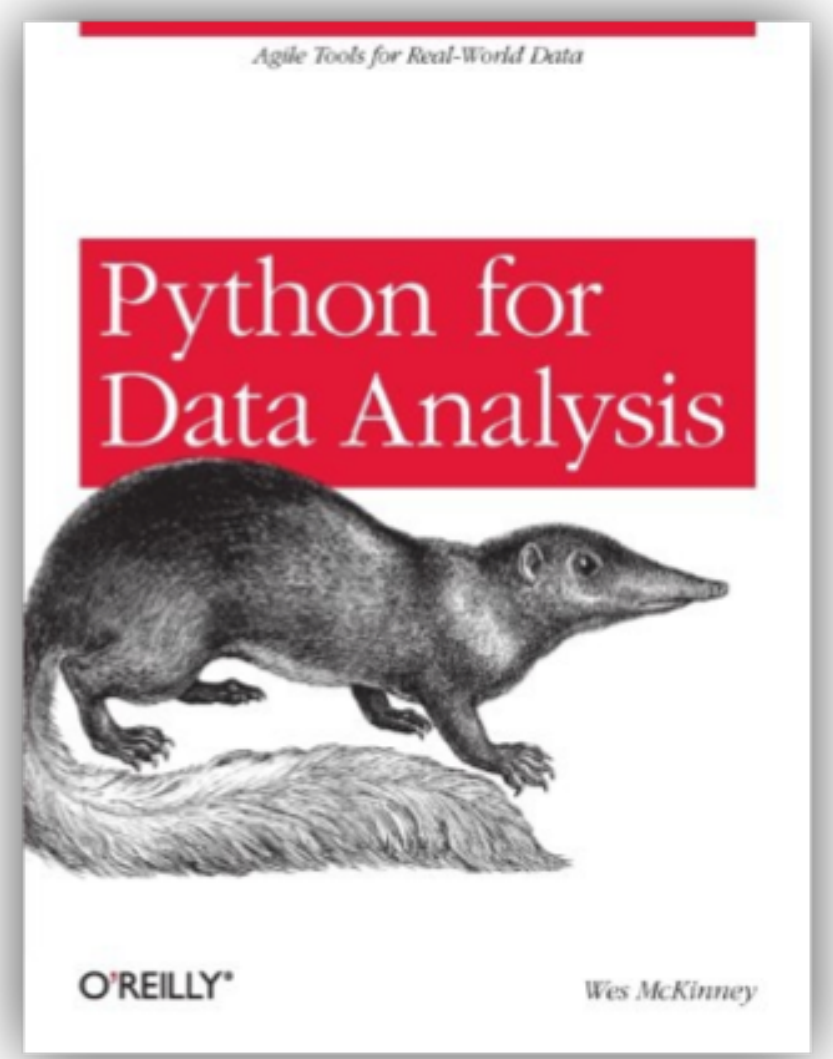
cloudera

Disclaimer #2: This is a nuanced discussion

cloudera

Who am I?

- Father of pandas (2008 -)
- Financial analytics in R / Python starting 2007
- 2010-2012
  - Hiatus from gainful employment
  - Make pandas ready for primetime
  - Write "Python for Data Analysis”
- 2013-2014: DataPad with Chang She & co
- 2014 - : Cloudera



cloudera

What’s in a DataFrame?

cloudera

What's in a DataFrame?
A table with some rows
By any other name
would analyze as sweet.

cloudera

Got a table?

cloudera

Got a table?
Put a DataFrame (interface) on it!

cloudera

What is this “data frame” that you speak of

- A table-like data structure
- An API / user interface for the table
  - Selecting data
  - Math and relational algebra (join, filter, etc.)
  - File / database IO
  - *ad infinitum*

Some axes of comparison

- Data structure internals (types, in-memory representation, etc.)
- Basic table API
- Relational algebra support
- Group-by / split-apply-combine API
- Performance, memory use, evaluation semantics
- Missing data
- Data tidying / ETL tools
- IO utilities
- Domain specific tools (e.g. time series)
- …

cloudera

The Great Data Tool Decoupling™

- Thesis: over time, user interfaces, data storage, and execution engines will decouple and specialize
- In fact, you should really want this to happen
  - Share systems among languages
  - Reduce fragmentation and “lock-in”
  - Shift developer focus to usability
- Prediction: we’ll be there by 2025; sooner if we all get our act together

Crafting quality data tools

- Quality / usefulness is usually forged by the fire of battle
- Real world use cases and social proof trump theory
- Eat that dog food
- When in doubt? Look at the test suite.

cloudera

R data frames

- Thin layer on top of R list type
  - Sequence of named vectors
  - Can have row names (any R vector)
- Simple column and row selection API
- Analytics, data transformation, etc. left to base package and add-on libraries
- Richness / usability comes largely from libraries

Some awesome R data frame stuff

- “Hadley Stack”
  - dplyr, tidyr
  - legacy: plyr, reshape2
  - ggplot2
- data.table (data.frame + indices, fast algorithms)
- xts : time series

cloudera

R data frames: rough edges

- Copy-on-write semantics
- API fragmentation / inconsistency
  - Use the “Hadley stack” for improved sanity
- Factor / String dichotomy
  - `stringsAsFactors=FALSE` a blessing and curse
- Somewhat limited type system

dplyr

- Composable table API
- Good example of what the “decoupled” future might look like
  - New in-memory R/RCpp execution engine
  - SQL backends for large subset of API

Spark DataFrames

- R/pandas-inspired API for tabular data manipulation in Scala, Python, etc.
- Logical operation graphs rewritten internally in more efficient form
- Good interop with Spark SQL
- Some interoperability with pandas
- Partial API Decoupling! (it still binds you to Spark)

pandas

- Several key data structures, data frame among them
- Considerably more complex internals than other data frame libraries
- Some good things
  - Born of need
  - A “batteries included” approach
  - Hierarchical axis labeling: addresses some hard use cases at expense of semantic complexity
  - Strong time series support

cloudera

pandas: rough edges

- Axis labelling can get in the way for folks needing “just a table”
- Ceded control of its type system / data rep’n from day 1 to NumPy
- Inefficient string handling (uses NumPy object arrays)
- Missing data handling less precise than other tools

Julia: DataFrames.jl

- Started by Harlan Harris & co
- Part of broader JuliaStats initiative
  - More R-like than pandas-like
  - Very active: > 50 contributors!
- Still comparatively early
  - Less comprehensive API
  - More limited IO capabilities

cloudera

Other data frames

- Saddle (Scala)
  - Dev'd by Adam Klein (ex-AQR) at Novus Partners (fintech startup)
  - Designed and used for financial use cases
- Deedle (F# / .NET)
  - Dev'd by AK's colleagues at BlueMountain (hedge fund)
- GraphLab / Dato
  - Really good C++ data frame with Python interface
  - Dual-licensed: AGPL + Commercial
- That's not all! Haskell, Go, usw…

cloudera

We’re not done yet

- The future is JSON-like
  - Support for nested types / semi-structured data is still weak
- Wanted: Apache-licensed, community standard C/C++ data frame that we all use (R, Python, Julia)
- Bring on the Great Decoupling

cloudera

cloudera

Thank you

@wesmckinn

