# Python in quantitative finance

Wes McKinney[1]

[1]AQR

# Outline

1. Background
   - My perspective
   - Common financial research tools
   - Python's status
2. pandas library
   - Motivation
   - Interactive Demo
   - Related projects
3. Summary and questions

# What is quantitative finance?

- Applying mathematics to model market phenomena
- Identifying statistical or causal relationships in financial data sets
- Building systematic investing strategies
- In recent years: convergence of math, computer science, and economics

# About me and AQR

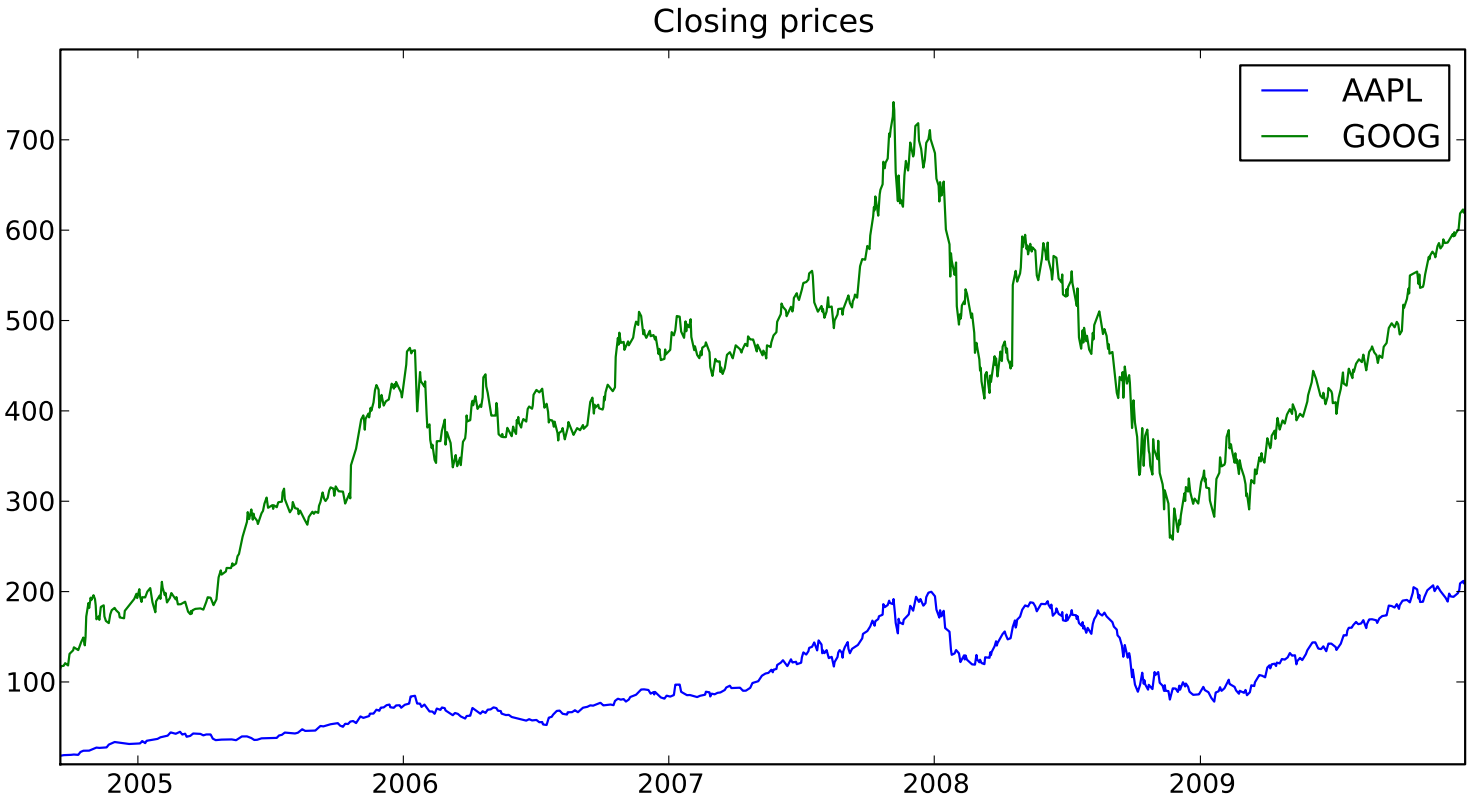
- Me: in the industry for 3 years, math background
- My company: AQR Capital Management
  - Founded in 1998
  - Manages both long-short hedge funds and more traditional long-only products
  - Combines economic intuition with statistical techniques to forecast price movements
  - Research tends to focus on longer-term relative value portfolios

# Common financial research tasks

- Data manipulation
  - Combine and transform raw data series
  - Handle missing observations, time series of different frequencies, other sources of heterogeneity
- Statistical estimation
  - Econometric analysis: linear regression and other more advanced models
  - Measuring and controlling risk: forecasting asset volatility
- Building investing strategies
  - Constructing tradable portfolios from raw data
  - Analyze strategy performance

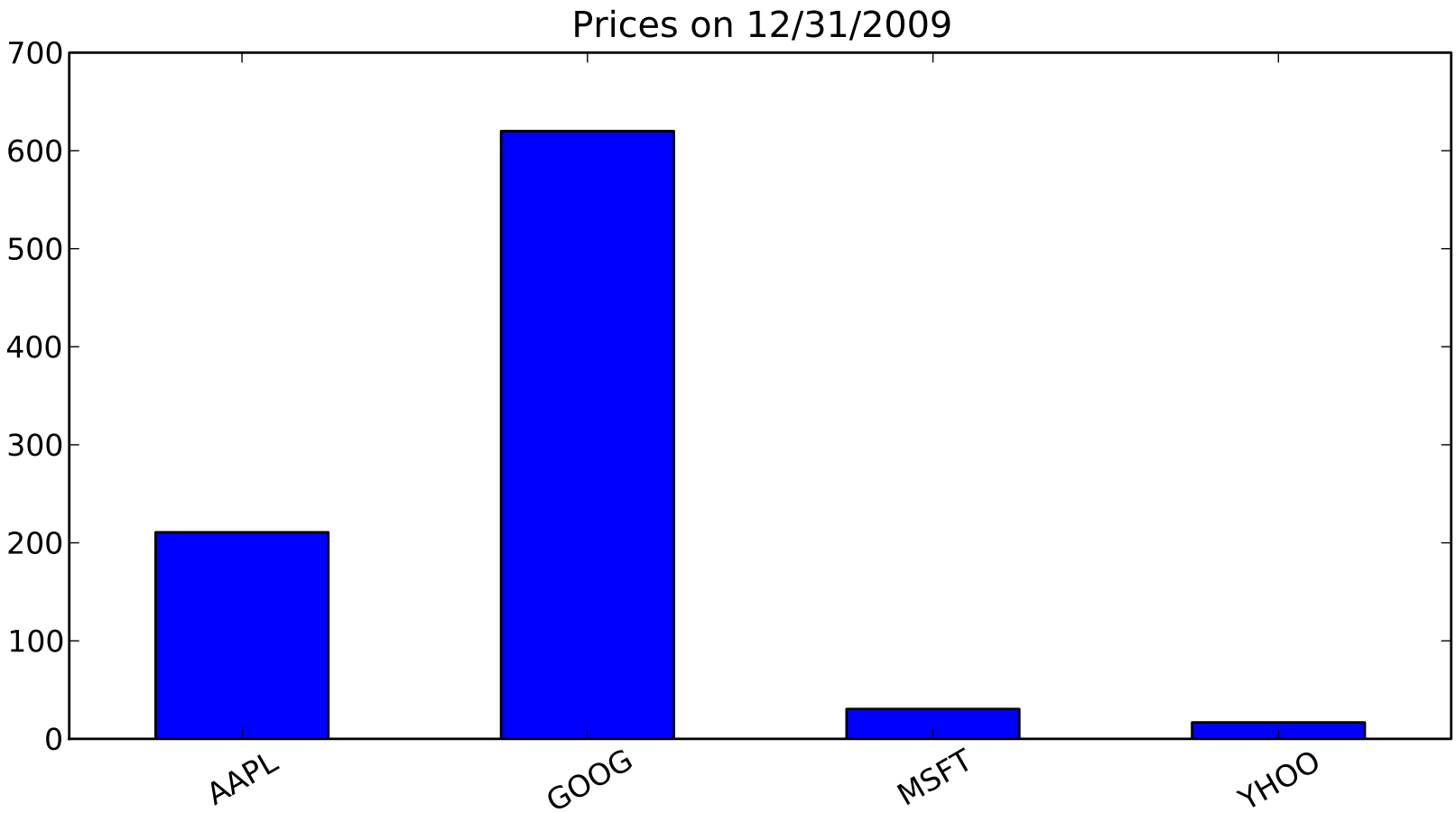
# Basic units of financial data

- Time series (TS): data points through time for one data field



# Basic units of financial data

- Cross section (XS): data points for many fields at **single** point in time



# Widely used research languages

- Commercial: MATLAB, Stata, eViews, etc.
- R
  - Popular among statisticians, econometricians, and time series analysis practitioners
  - Abundance of open-source packages on CRAN
  - Already available to Python programmers via **rpy**
- Research and implementation language are frequently not the same

# How about Python?

- Many excellent mature building blocks for doing finance work
  - numpy array / matrix functionality is flexible, powerful
  - scipy provides statistical functionality, optimization, other solvers
- IPython + matplotlib for interactive research / data visualization
- Can mix low-level code (C, Fortran, Cython / Pyrex) to boost performance
- Python language is great for building systems (expressiveness, debuggability, deployment)

# Where are Python and NumPy weak (for finance)?

- Relatively thin on statistical modeling / econometrics libraries
  - **scikits.statsmodels** and other projects are helping change that
- Many tools assume "clean" data sets; most often not the case
- Chicken and egg problem: less financial community presence
- Few tools specifically for time series and cross-sectional data

- **My goal:** help Python become a more compelling choice for finance work and other statistical applications

# Where are Python and NumPy weak (for finance)?

- Relatively thin on statistical modeling / econometrics libraries
  - **scikits.statsmodels** and other projects are helping change that
- Many tools assume "clean" data sets; most often not the case
- Chicken and egg problem: less financial community presence
- Few tools specifically for time series and cross-sectional data

- **My goal:** help Python become a more compelling choice for finance work and other statistical applications

# pandas project background

- Born of practicality at AQR in 2008
- Idea: data structures for labeled data; lightweight and easy-to-visualize
- Link identifiers (dates, tickers, etc.) to standard NumPy arrays
- Handles both TS and XS data with no fuss
- Makes very few assumptions about data cleanliness (built with NaN-handling from ground up)
- Core indexing mechanism deals with data alignment, easy reshaping
- Etymology: **pan**el **da**ta system
- In heavy production use at AQR

# pandas project background

- Born of practicality at AQR in 2008
- Idea: data structures for labeled data; lightweight and easy-to-visualize
- Link identifiers (dates, tickers, etc.) to standard NumPy arrays
- Handles both TS and XS data with no fuss
- Makes very few assumptions about data cleanliness (built with NaN-handling from ground up)
- Core indexing mechanism deals with data alignment, easy reshaping

- Etymology: **pan**el **da**ta **s**ystem
- In heavy production use at AQR

# What's in the library?

- Flexible NumPy-based data structures for 1, 2, and 3 dimensional data
- Common time series statistical operators
  - example: moving {sum, average, std, skewness}
- Nascent integrated econometrics / statistical regression library

# Demo

# Some related projects

- scikits.statsmodels
- scikits.timeseries
- tabular
- la (larry: labeled array object)

# Ideas for future

- Expand existing functionality to address other applications
- Implement more statistical models / wrap scikits.statsmodels classes
- Develop seamless **rpy** interface to leverage CRAN wealth
- Better / more efficient IO functions for getting data into pandas

# Questions / Thanks

- Contact: wesmckinn@gmail.com
- Website: pandas.sourceforge.net
- Download from Python package index
- AQR website: www.aqr.com

# AQR Disclaimer

The views and opinions expressed herein are those of the author and do not necessarily reflect the views of AQR Capital Management, LLC its affiliates, or its employees. The information set forth herein has been obtained or derived from sources believed by author to be reliable. However, the author does not make any representation or warranty, express or implied, as to the information's accuracy or completeness, nor does the author recommend that the attached information serve as the basis of any investment decision. This document has been provided to you solely for information purposes and does not constitute an offer or solicitation of an offer, or any advice or recommendation, to purchase any securities or other financial instruments, and may not be construed as such. This document is intended exclusively for the use of the person to whom it has been delivered by the author, and it is not to be reproduced or redistributed to any other person.