pandas: Rich Data Analysis Tools for Quant Finance

Wes McKinney

April 24, 2012, QWAFAFEW Boston

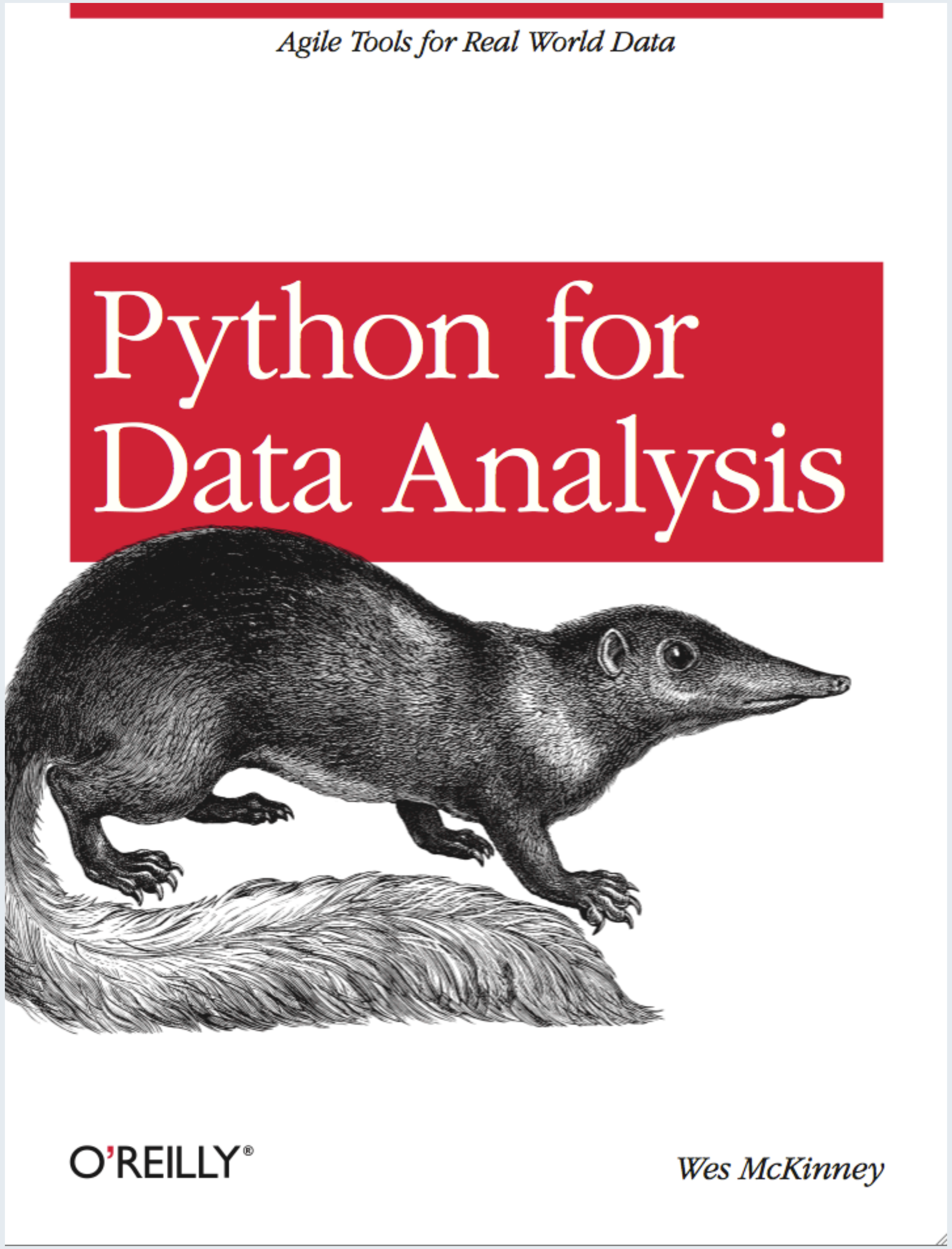
about me

WES MCKINNEY

- MIT ’07
- AQR Capital: 2007 - 2010
- Global Macro and Credit Research
- pandas: 2008 - Present
- wes (at) lambdafoundry.com
- Twitter: @wesmckinn

Upcoming book

- Python language essentials
- Core scientific libraries
- pandas
- Visualization
- Case studies
- Look out for Rough Cuts version on oreilly.com



Lambda Foundry

- http://lambdafoundry.com
- Incorporated January 2012
- Mission: Better solutions to data-driven business problems
- **RapidQuant**: Financial analytics libraries and research environment
- Open Source Development and Support
- Training and Consulting

RapidQuant Platform

- Integrated, interactive research environment and workbench
- Analytics libraries, standard data algorithms, and transforms
- Frameworks for backtesting, risk modeling, portfolio management
- Vendor data integration
- Optimized data interfaces
- Caching and data management
- Testing tools
- Distributed computing

Outline

- Why Python for Quants?
- Scientific Python Foundation
- Pandas essentials
- Time series
- Group-wise data manipulation
- Plotting and Visualization
- More examples

Research vs. Production

Research needs

- Rapid iteration, exploration
- Interactive Data Analysis
- Statistical modeling tools
- Backtesting framework
- Rich Visualization
- Reporting, Excel integration
- High perf computations

Production needs

- Model and data controls, versioning
- Rigorous testing, robustness
- Large-scale process management
- Modularity, extensibility
- Productive system dev tools
- Strong interoperability
- High perf computations

Python: one-language solution to the two-domain problem?

Python

- Easy to learn, but richly featured
- Concise, but highly readable
- “Python gets out of my way”
- Multi-paradigm: object-oriented, functional, procedural
- Easy integration with C / C++ / Fortran
- Mature scientific libraries and large, active community

Python for Systems

- Strong library support
- Excellent maintainability
- Debugging, profiling, static code analysis tools
- Numerous testing frameworks
- Deployment, packaging tools
- GUI toolkits, web development, network applications
- One of Google’s main languages

Core financial stack

- NumPy: multidimensional arrays, linear algebra
- pandas: data manipulation toolkit
- IPython: rich interactive environment
- SciPy: like MATLAB toolboxes
- statsmodels: statistics and econometrics
- Visualization: matplotlib

NumPy: Numerical Python

- Fast array processing library implemented in C
- ndarray: multidimensional array object
- Linear algebra operations
- Random number generation
- Efficient binary IO
- Other stuff: FFT, f2py, masked arrays, ...

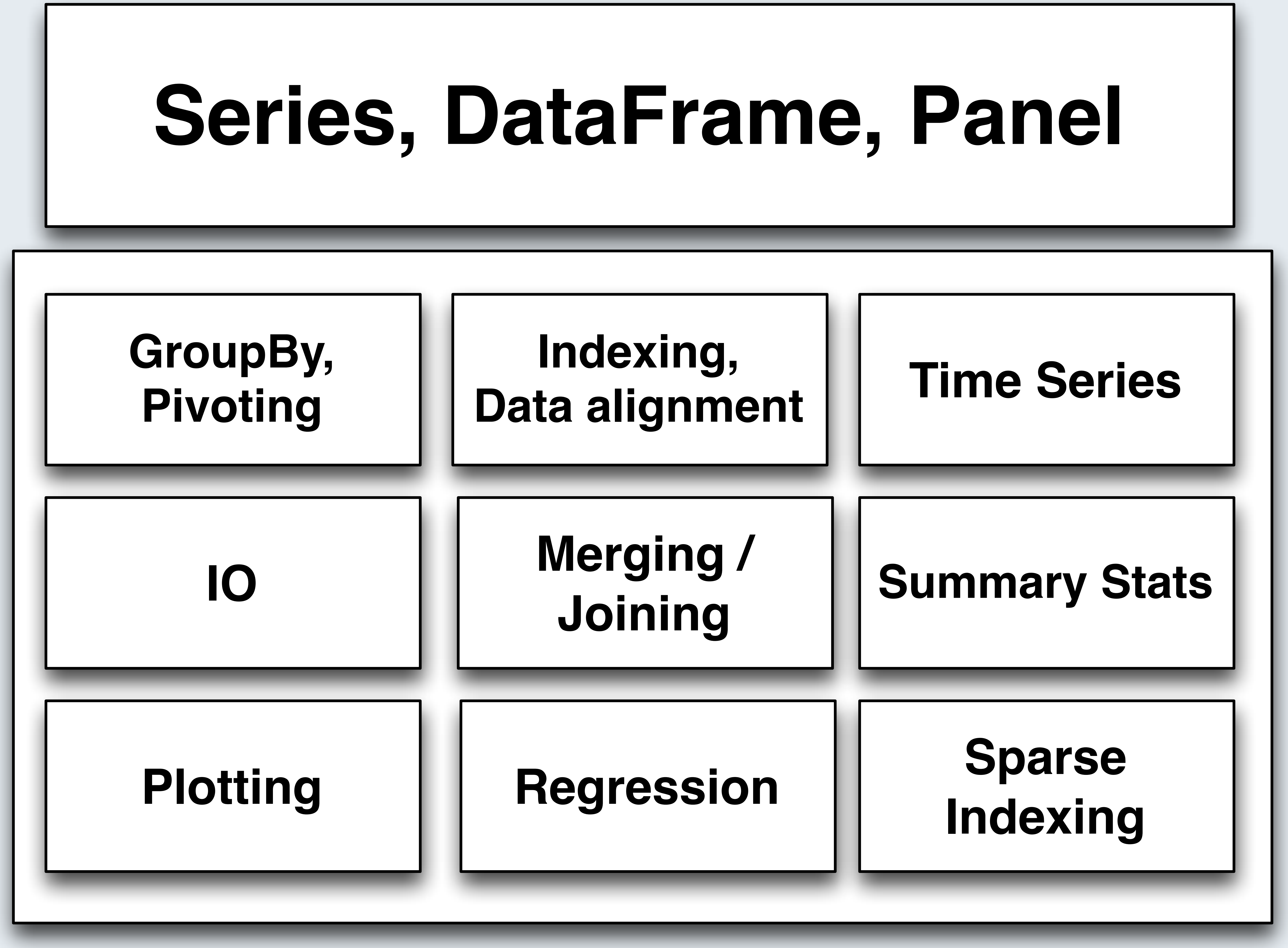
IPython

- Rich, interactive shell environment
- Terminal version + Rich Qt-based Shell with inline plotting
- Web Notebook Format
- Tab completion, introspection
- %run command
- Debugging and profiling tools

pandas

- Powerful data handling tool built on NumPy
- Started building in 2008 at AQR Capital, open sourced
- Has become widely used in quant finance
- Mature, well-tested codebase
- Powerful time series capabilities
- Widely used in production in the quant finance industry
- Upcoming 0.8.0 release: major time series improvements
- http://pandas.pydata.org

pandas



Series, DataFrame, Panel

GroupBy, Pivoting

Indexing, Data alignment

Time Series

IO

Merging / Joining

Summary Stats

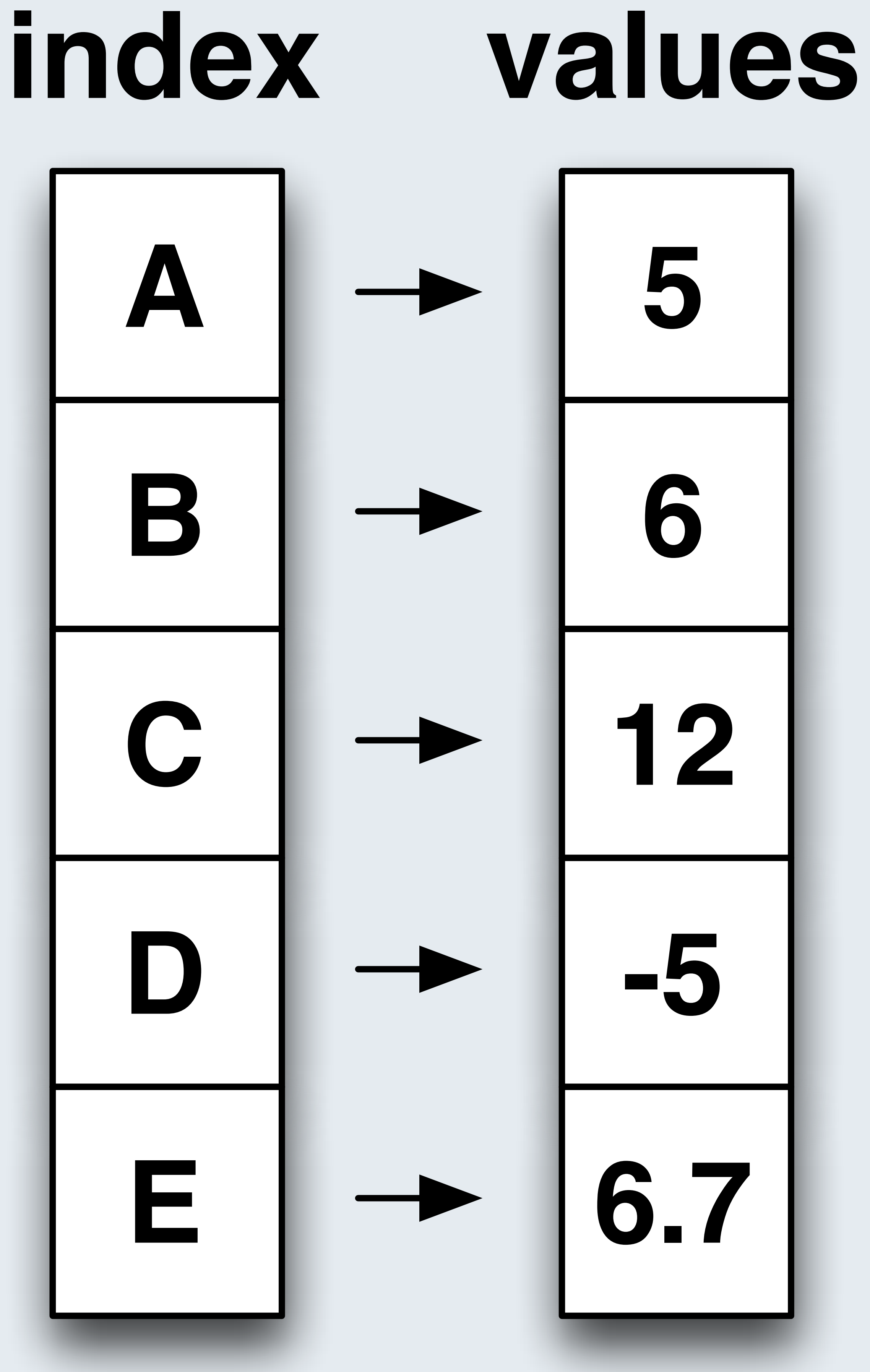
Plotting

Regression

Sparse Indexing

Series

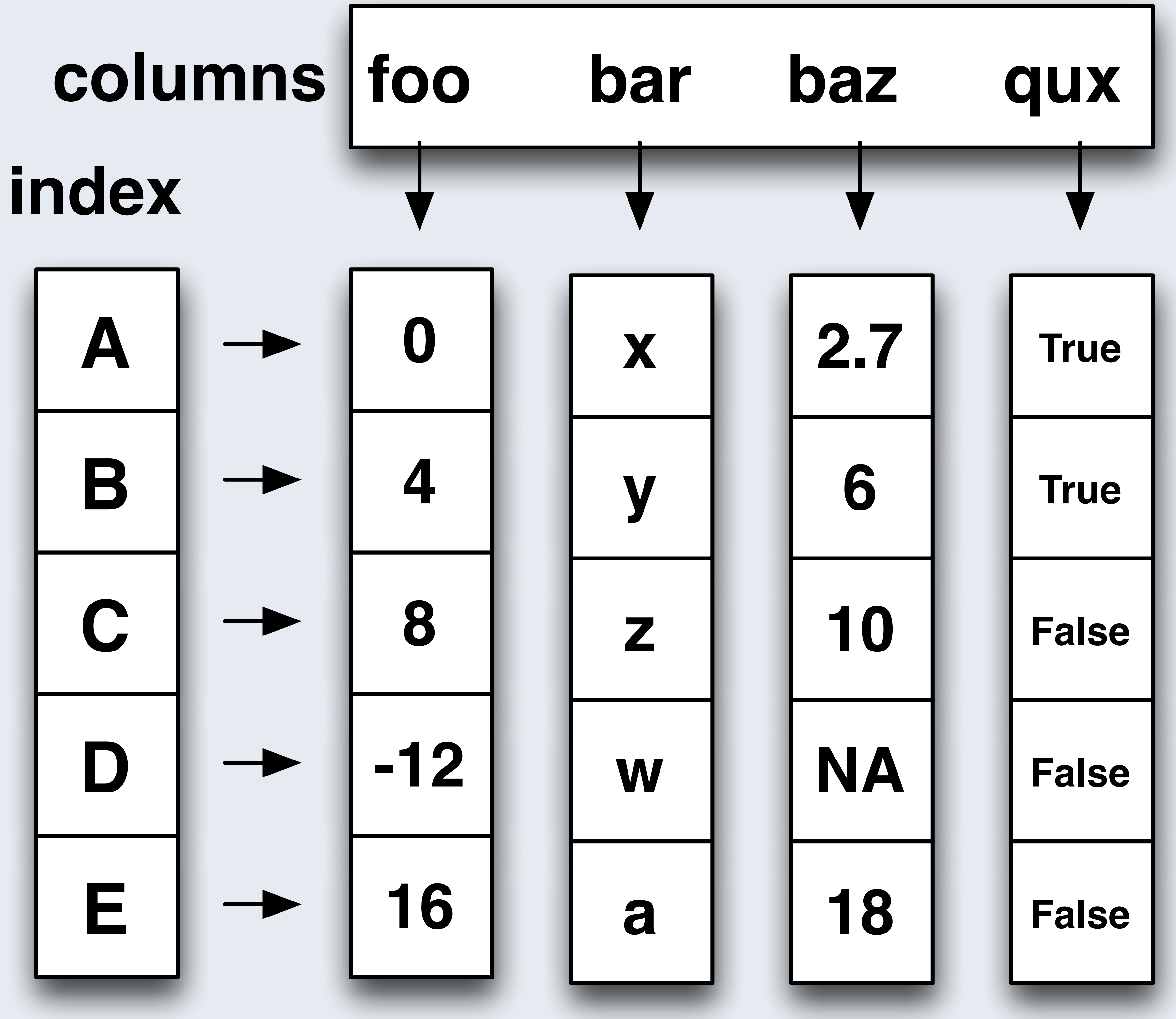
- 1D labeled array for cross-sections, time series
- Array of data, any type
- Array of labels, the “index”
  - Orderedness not required
- Index can be integers, time, assets, or any other identifiers



index	values
A	5
B	6
C	12
D	-5
E	6.7

DataFrame

- Table of Series objects
- Columns can be different types
- Shared row index
- Dict-like column insertion/deletion
- Select/slice data using row and / or column labels



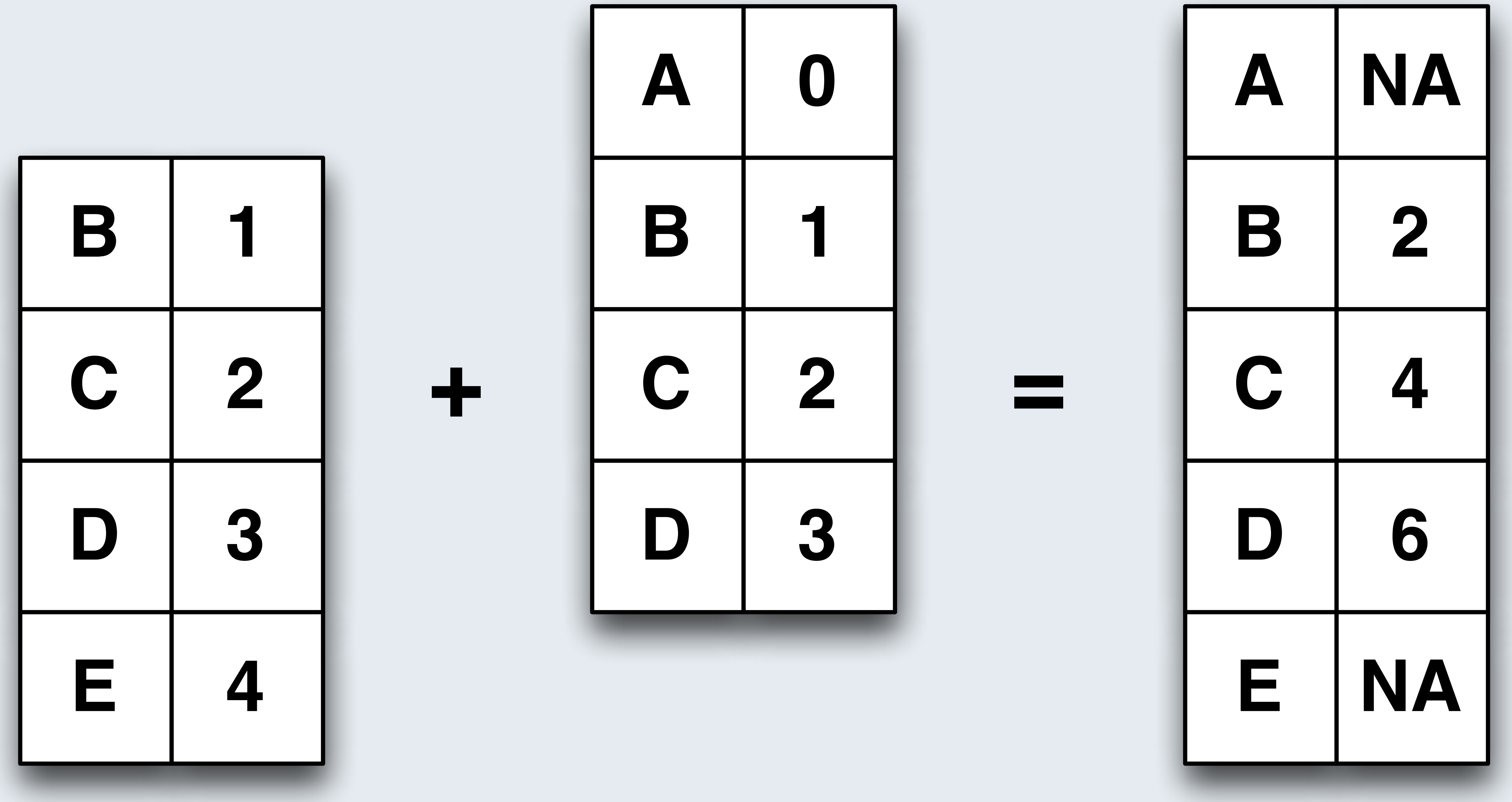
index	foo	bar	baz	qux
A	0	x	2.7	True
B	4	y	6	True
C	8	z	10	False
D	-12	w	NA	False
E	16	a	18	False

pandas input / output

- Read from / write to a variety of formats
- CSV, clipboard, fixed-width, generalized table
- Excel 2003, 2007
- DataFrame save and load via Pickling
- Managed storage solutions
  - HDF5: pandas.io.pytables
  - SQL: pandas.io.sql
- Web based API's like Yahoo! Finance, Fama-French, FRED

Data alignment

- Arithmetic auto-aligns data on label (ticker, timestamp, ...)
- DataFrame aligns on row and column labels



B	1
C	2
D	3
E	4

+

A	0
B	1
C	2
D	3

=

A	NA
B	2
C	4
D	6
E	NA

Indexing and selection

- Select rows/columns by position or label
- Slice chunks of objects without copying
- Insert and delete DataFrame columns
- Hierarchical indexing: multiple levels of keys on a single axis
- Many time series conveniences

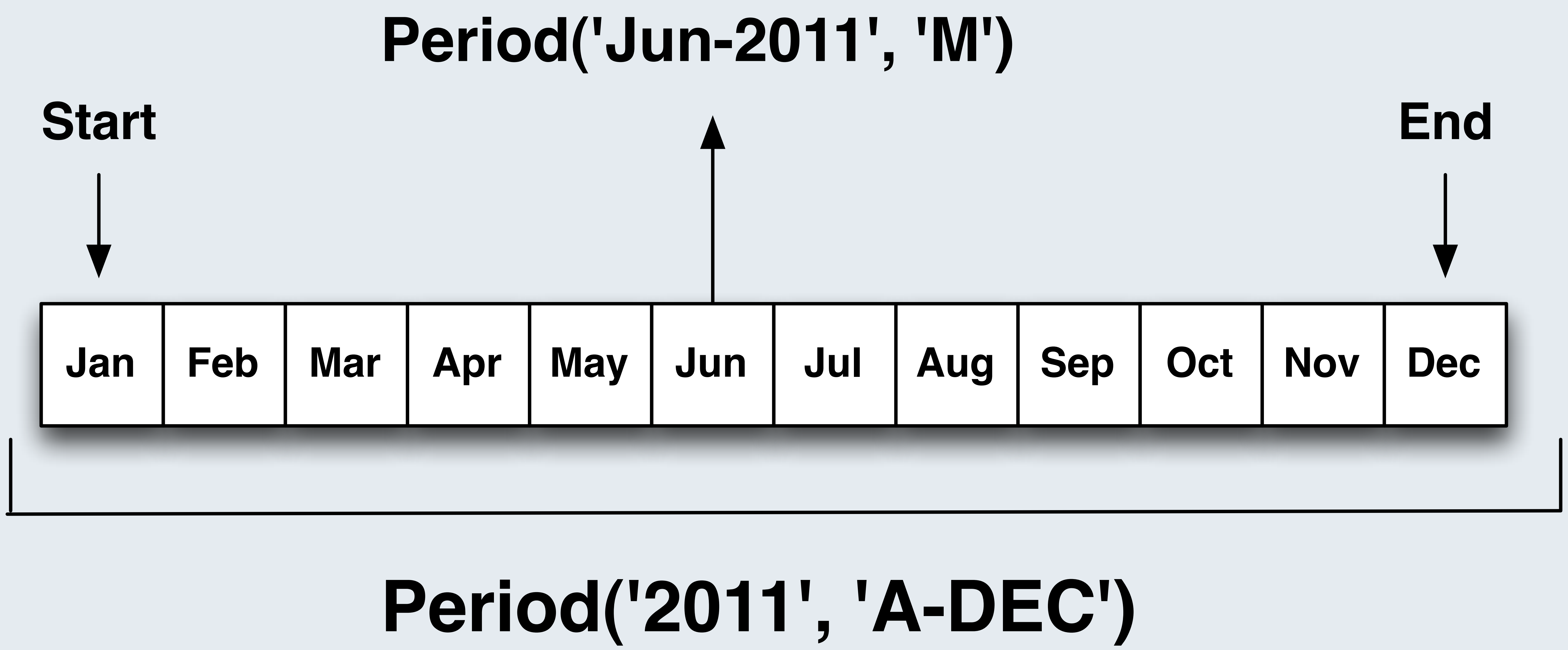
Time series

- Time series representations
- Fixed frequency and irregular data handling
- Date arithmetic
- Time zone handling
- Resampling: high to low, low to high
- Interpolating missing values
- Moving window functions

Date and time types

- **Timestamp**: specific moment in time
- **Period**: span of time
  - e.g. 2010, June 2007, 1997Q3
- **Interval**: defined by 2 timestamps
- **Timedelta or Duration:** a length of time
  - e.g. 3 days; 30 minutes; 2 hours

Period arithmetic



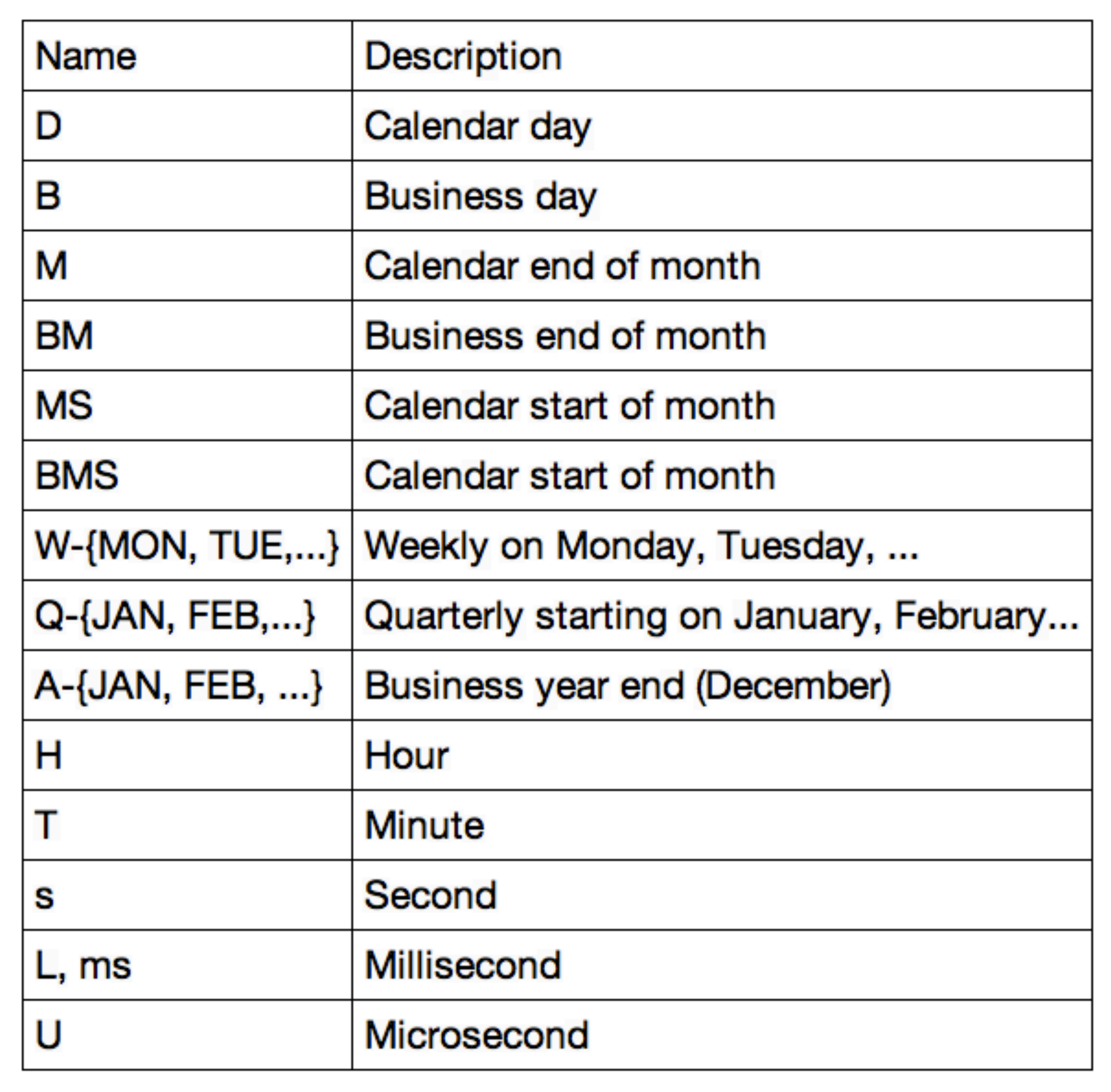
```
In [7]: p = Period('2011')

In [8]: p.asfreq('M', 'start')
Out[8]: Period('Jan-2011', 'M')

In [9]: p.asfreq('M', 'end')
Out[9]: Period('Dec-2011', 'M')
```

Fixed frequency time series

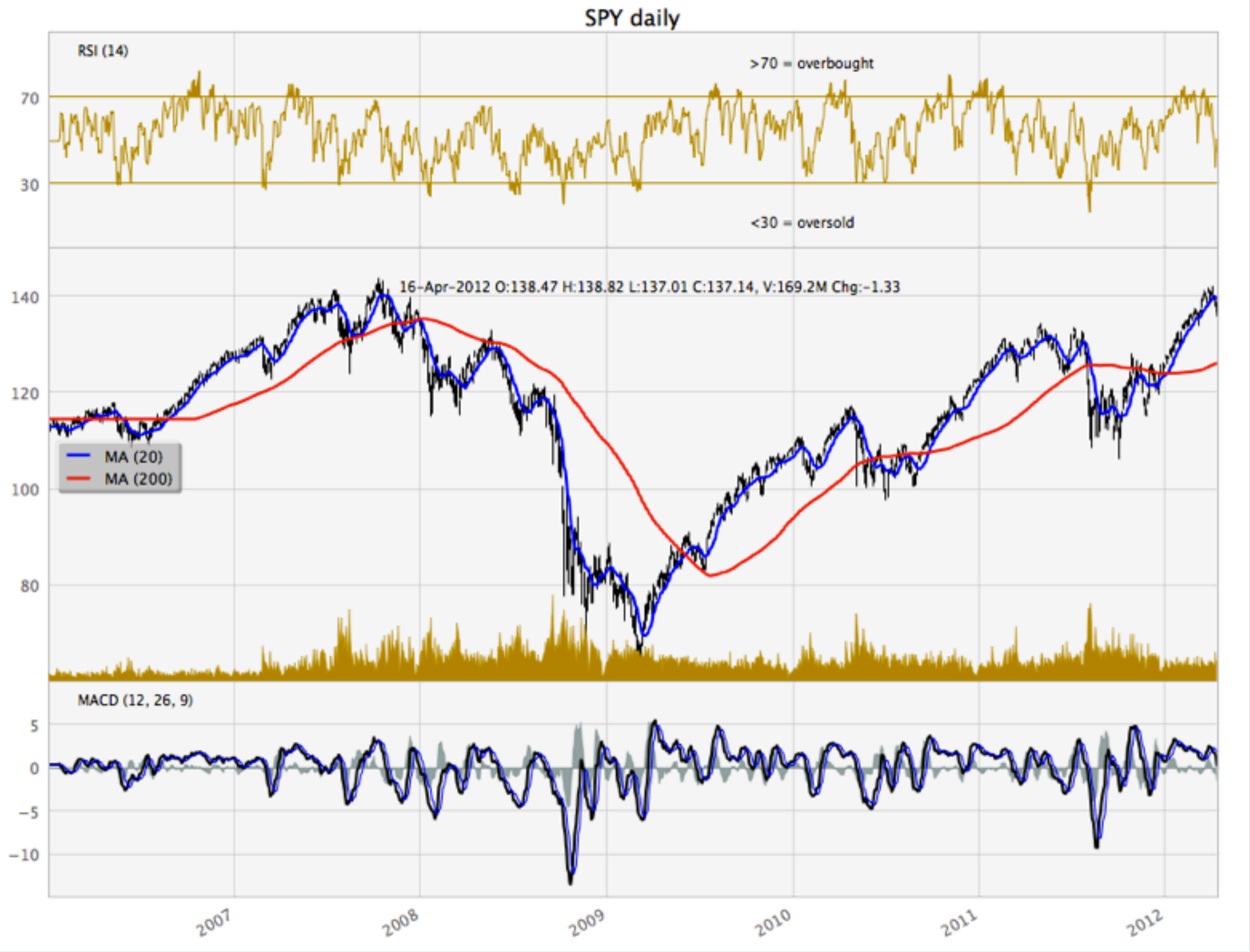
- Irregular by default, but can have a frequency
- Used for: shifting, frequency conversion, date arithmetic
- Upcoming changes in 0.8.0



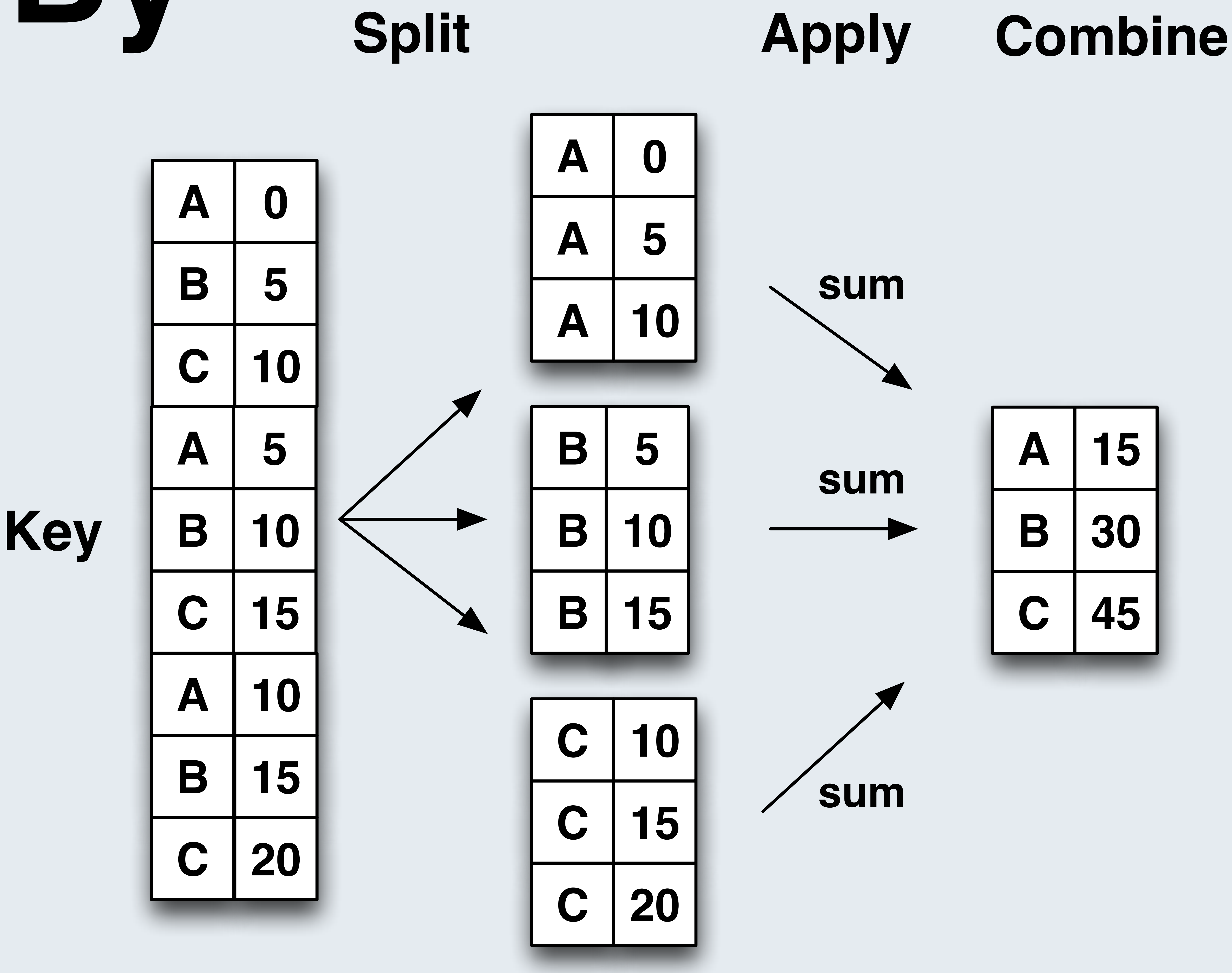
Name	Description
D	Calendar day
B	Business day
M	Calendar end of month
BM	Business end of month
MS	Calendar start of month
BMS	Calendar start of month
W-{MON, TUE,...}	Weekly on Monday, Tuesday, ...
Q-{JAN, FEB,...}	Quarterly starting on January, February...
A-{JAN, FEB, ...}	Business year end (December)
H	Hour
T	Minute
s	Second
L, ms	Millisecond
U	Microsecond

Visualization

- matplotlib: general purpose plotting
- IPython integrates with matplotlib
- Plot windows or inline plotting
- Many convenience functions functions in pandas
- Complex plots may take effort



Group By



Split

Apply

Combine

Key

A	0
B	5
C	10
A	5
B	10
C	15
A	10
B	15
C	20

A	0
A	5
A	10

B	5
B	10
B	15

C	10
C	15
C	20

sum

sum

sum

A	15
B	30
C	45

pandas roadmap

- Pandas for Big Data
- Integration with JavaScript visualization, e.g. d3
- More integration with statsmodels (econometrics) and scikit-learn (machine learning)
- ggplot2-like plotting interface
- Better text file processing capabilities

pandas vs. R

- More time series features, higher performance than zoo, xts, fts, its, etc.
- DataFrame merge performance 5-30x faster
- Better performance than plyr / reshape2 for reshaping and groupby operations
- Symmetric treatment of row- and column-oriented operations
- No ggplot2 equivalent; weak area for Python, have plans to work on this summer