Identifying Hidden Risk in Portfolios

A graph-based analysis of the stock market

Scott Stossel, Kevin Coppa, Daniel Wang, Jane Lai

Business Case Scenario

The Problem

Traditional diversification falls short:

- Investors rely on sector-based diversification
- Stocks from different sectors can still be correlated
- Hidden dependencies increase systemic risk

The Solution

Model stocks as a correlation network:

- Detect clusters of correlated assets with Louvain
- Use PageRank to find the most influential stocks
- Uncover cross-sector relationships not visible in standard models

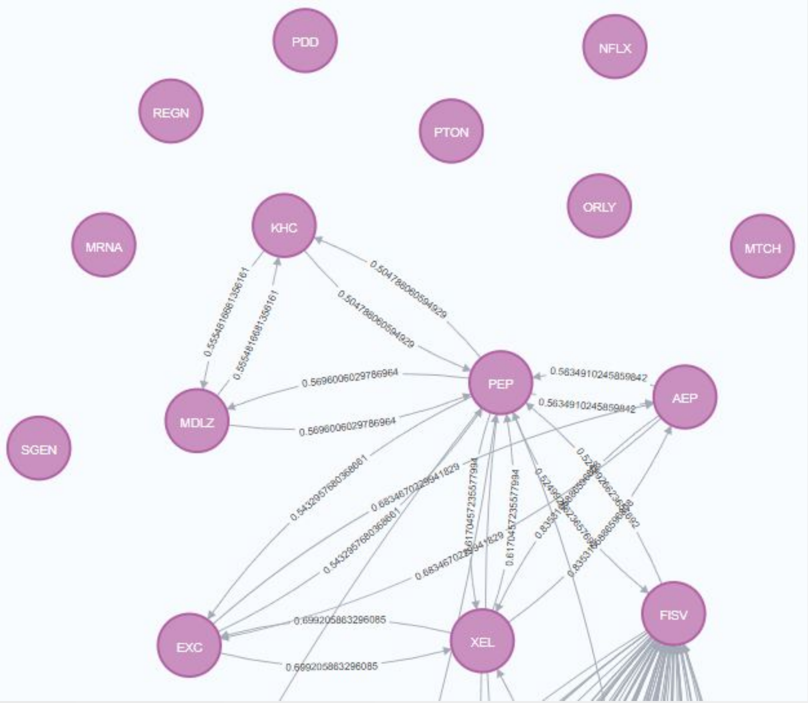
Graph Design

Each node represents a ticker in the SP500

Edges connect stocks with high correlations

Benefits:

- Identify stocks that move similarly
- Minimize overlapping risk exposure to particular clusters



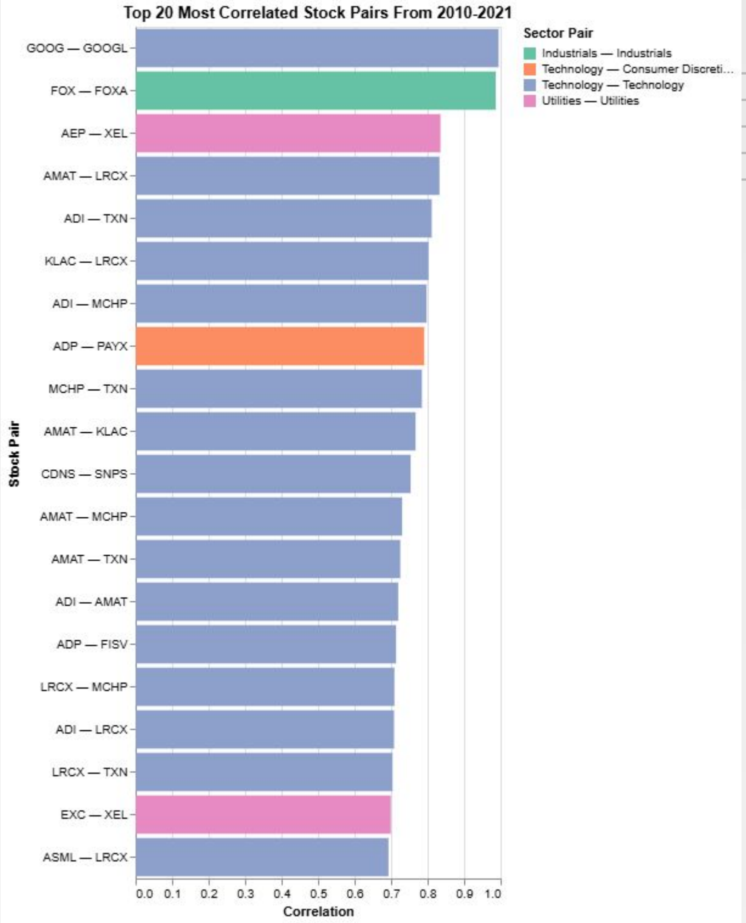
Community Detection - Leiden

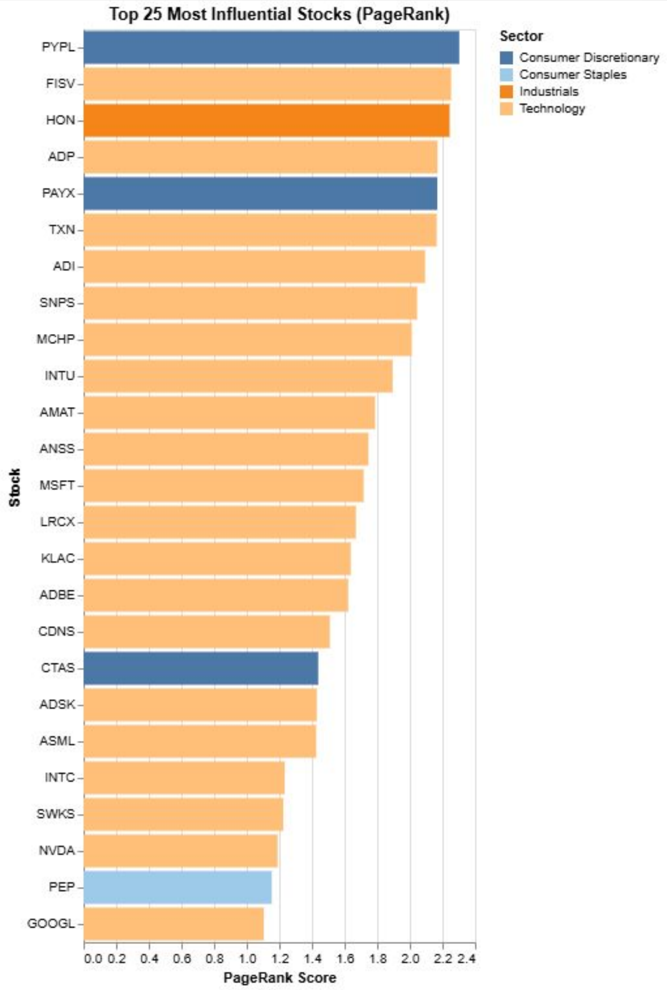
Benefits:

- Find densely connected groups of tickers
- Minimize risk from correlated price movements

Issues With Relationship Databases:

- Can’t directly represent relationships between two different rows
- Many joins between correlation tables would be needed





Centrality – PageRank

Benefits

- Finds influential tickers that represent market movers
- Limits their exposure to market or sector wide risk
- Increases exposure to market leaders

Relational Database Issues

- Don't efficiently support node's relationships and their relationships

Volatility Graph Design

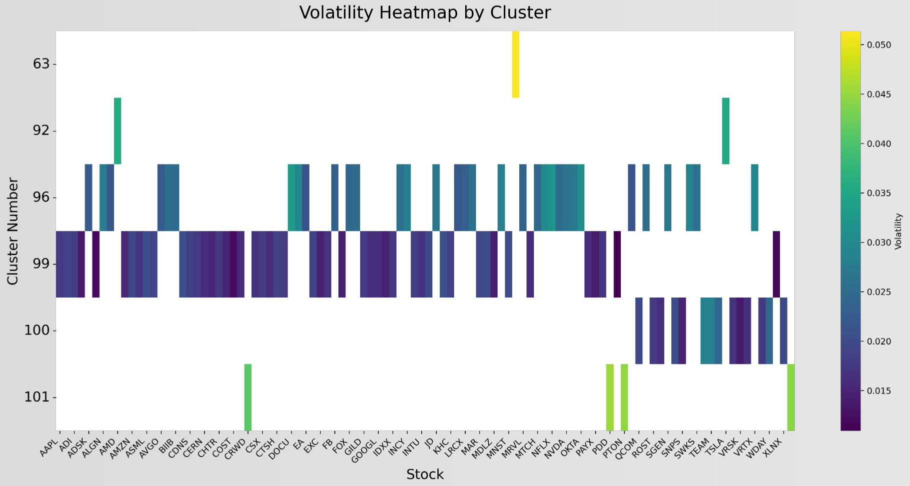
Each node represents a ticker in the SP500.

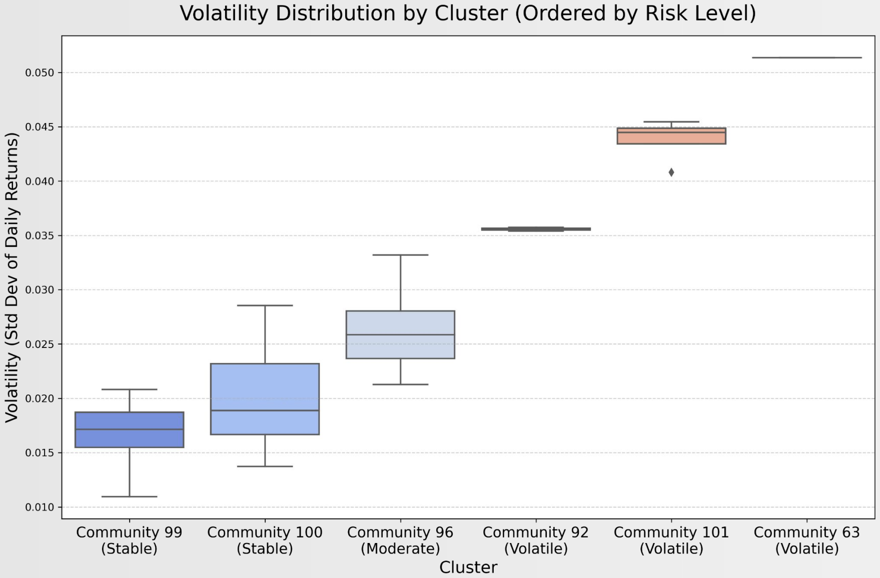
Edges connect stocks base on their volatility.

The design helps identify stocks that are more or less stable.

This helps investors understand stocks with different volatility levels.

KNN Clustering





Visualizing volatility levels for stocks in a given cluster

- Provides insights for diversification and risk-level

Variance in volatility among clusters

- Provides insight into how reliable the cluster is at categorizing volatility
- Degrees of volatility visualized

Use Case for MongoDB

Business Case:

Manage stock portfolio holdings with Louvain communities for diversification analysis

- Can store portfolio holdings with Louvain community labels (including all financial data) in one place
- Aggregate stock count per community (exposure analysis)
- Set diversification rules (e.g., max 3 stocks/community) and filter
- Track portfolio changes with time-stamped documents
- Flexible schema for financials, sectors, correlations, sentiment

Why MongoDB over a Relational DB?

- Relational DB:
  - Rigid schema not suited for nested or semi-structured data
  - Complex joins required for graph-like queries
  - Limited schema evolution
  - Not optimized for storing edges, community labels, or time-series documents

Use Case for Redis

Business Case:

Real-time enforcement of diversification rules and threshold alerts during portfolio adjustments

- Store community stock counts as keys for instant access
- Enforce diversification limits with quick key-based rule checks
- Use Pub/Sub to send immediate alerts
- Ultra-low latency → fast access (ie. trading apps)
- Complement MongoDB as a high-speed decision layer

Why Redis over a Relational DB?

- Redis stores data in RAM → instant updates
- Relational DB not suitable for for near-instant updates and conditional logic
- Relational DB lacks Pub/Sub feature, need external system

Conclusions

What we learned:

Graph algorithms like Louvain and PageRank expose risk concentration and influence

NoSQL tools like MongoDB and Redis support flexible scalable, real-time portfolio analysis

Takeaways for Investors:

Traditional diversification doesn’t guarantee reduced risk

Graph analytics can optimize diversification

Real-time insights enable faster, better investment decisions

Thanks for listening!

Q&A