Introduction to Quantitative Trading

BOPU Technologies

Overview

- Concepts of Quantitative Trading
- Financial Markets
- Quantitative Strategies
- Trading Systems

  - Lecture Objective:
    - Not to learn a trading algorithm that can make money immediately;
    - Understand the very basics of quantitative trading from a computer science perspective

What is quantitative trading?

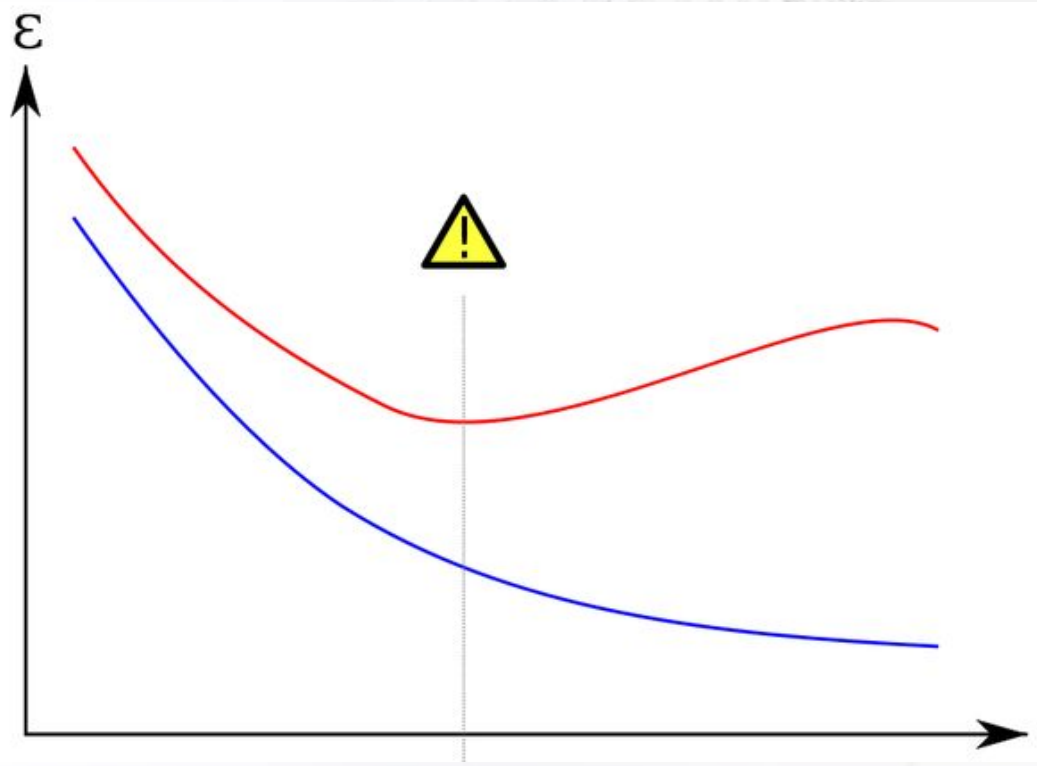
- Make trading decisions with computer algorithms
- Minimize human intervention
- Trader's roles:
  - Monitor the trading program's execution
  - Stop the program in abnormal market conditions
  - Adjust parameters of the trading algorithms

What is quantitative trading?

- Not equivalent to the so called "technical analysis"
- Trading decisions are backed up by
  - Solid math models
  - Systematic backtests
- Trade executions are usually automated

Backtesting

- Use historical data to test trading ideas
- The problem of overfitting
  - training data and validation data
  - testing data
  - forward testing



Evaluation of Trading Performances

- Test for statistical significance of strategy returns
  - t-test
  - Monte Carlo Simulation
- Benchmarks:
  - Annualized returns
  - Maximum drawdown
  - Time to recover to high watermark
  - Sharpe Ratio, etc

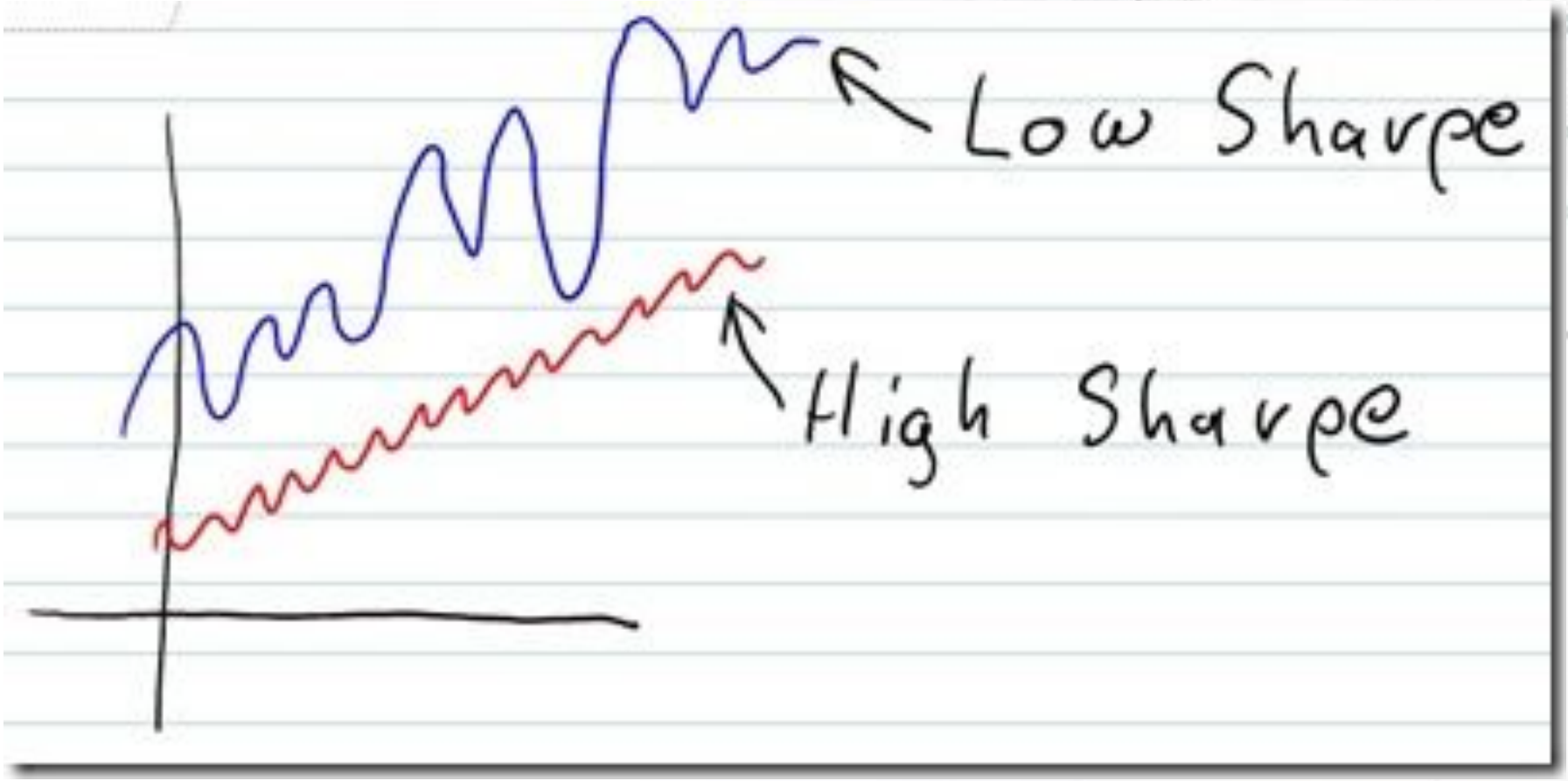
Sharpe Ratio

- Sharpe Ratio (frequently used, but has its own problem)
  - R : return
  - Rf : risk-free rate (usually 3-month treasury bill in US)

$$S = \frac{E[R - R_f]}{\sqrt{\operatorname{var}[R]}}.$$

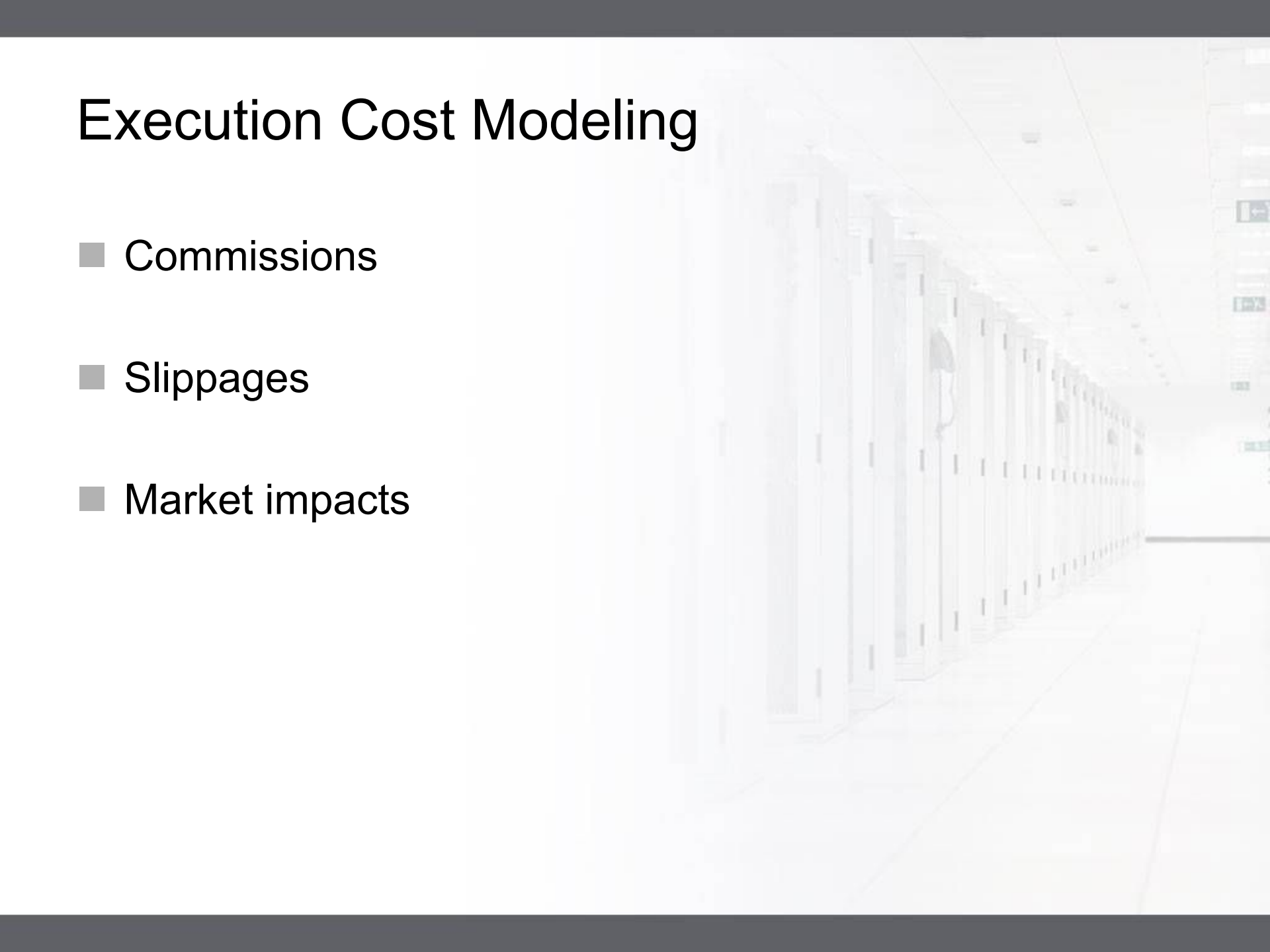
- Others: Sortino Ratio, Kurtosis, Skewness, etc

Sharpe Ratio



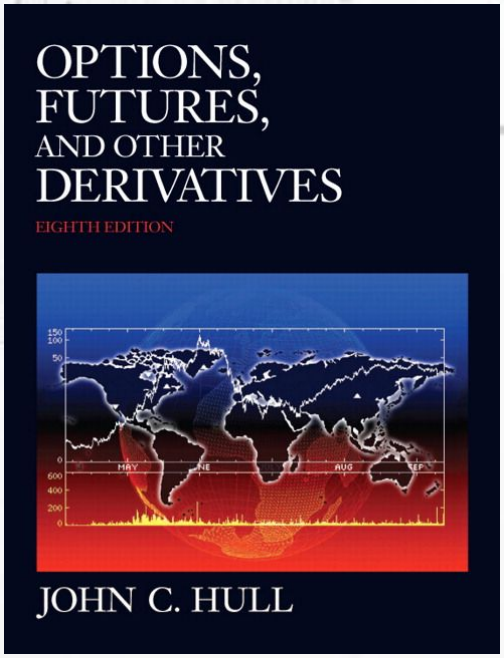
Execution Cost Modeling

- Commissions
- Slippages
- Market impacts



Financial Markets

- Stocks and ETFs
- Fixed Income
  - Treasuries, Bonds,
- Currency
- Futures
- Options
- Other more complicated derivatives, see the quant bible by John Hull:



Traditional Auctions

- Designated market makers (exchange specialist, liquidity providers)
- Provide tradable bid and ask prices along with sizes
- Market makers earn bid-ask spreads

Electronic Trading

- Market participants places orders to an electronic order book
- Order Types
  - Market Orders
  - Limit Orders
  - Stop Orders
  - Many others
- Limit Order Book
  - Limit orders only
  - Usually first come first serve
  - When an order comes, try to match the best opponent price

买卖队列		总委托量	每单均量											
卖五	13.05	344	◆19.1	10	10	6	410	10	10	2	2	10	4	2
卖四	13.04	24	12.0	22	2 [总单数2]									
卖三	13.03	31	15.5	30	1 [总单数2]									
卖二	13.01	14	7.0	4	10 [总单数2]									
				10	12 [总单数13]									
卖一	13.00	235	18.1	144	1	2	5	10	10	7	4	5	5	20
买一	12.99	3282	◆182.3	50	25	1553	10	1501	20	9	10	11	1	8
				10	40	20	2	1	3	8 [总单数18]				
买二	12.98	2746	◆211.2	881	882	70	800	39	10	20	5	5	1	1
				30	2 [总单数13]									
买三	12.97	4	2.0	1	3 [总单数2]									
买四	12.92	15	7.5	5	10 [总单数2]									
买五	12.91	10	10.0	10 [总单数1]										
买六	12.90	3435	◆490.7	8	1400	2	10	2	1	2012 [总单数7]				
买七	12.89	367	◆367.0	367 [总单数1]										
买八	12.88	59	11.8	1	16	16	10	16 [总单数5]						
买九	12.87	1	1.0	1 [总单数1]										
买十	12.86	1	1.0	1 [总单数1]										
十一	12.85	200	66.7	124	26	50 [总单数3]								
十二	12.83	4	2.0	1	3 [总单数2]									
十三	12.82	144	16.0	50	8	40	8	5	4	4	6	19 [总单数9]		
十四	12.81	941	◆55.4	150	4	500	30	20	10	10	170	5	3	10

Quantitative Strategies

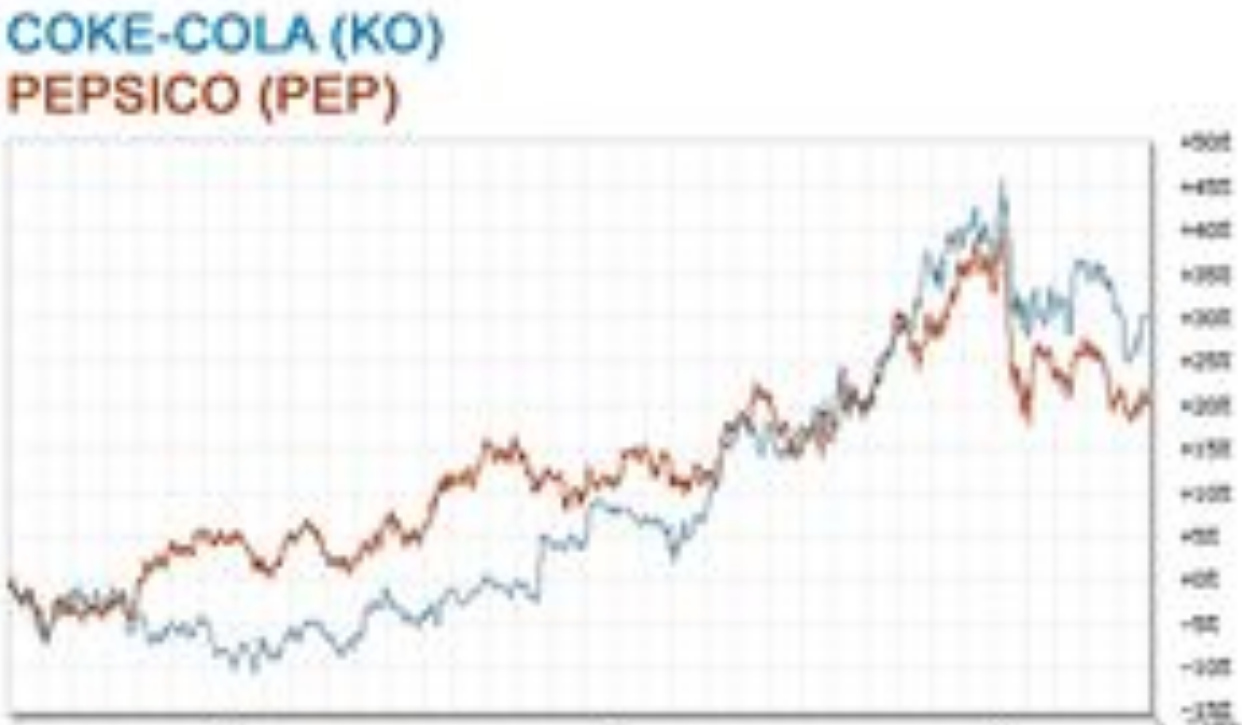
- Trend following (Momentum)
  - Past winners will be winners again.
  - Past losers will be losers again.
- Mean reverting
  - Past winnners will be losers.
  - Past losers will be winners.
- Alpha Model
  - Use quantitative approach (e.g., multi-factor models) to find stocks that can beat the market
  - Hedge risk by shorting index futures
- Market Neutral
  - Dollar-neutral
  - Beta-neutral

Quantitative Strategies

- Arbitrage
  - The law of one price
  - Currency Triangular Arbitrage
    - EUR/USD 1.5
    - USD/JPY 100
    - EUR/JPY 120
  - Index Arbitrage
  - ETF Arbitrage

Quantiative Strategies

- Statistical Arbitrage
  - Bet that a basket of long/short portfolio is mean reverting.
  - Pair trading - simplified statistical arbitrage
    - Example:



Quantitative Strategies

- Market Making
  - Buy low sell high!
  - Passively place limit orders to the order book
  - Earn bid/ask spread
  - Earn exchange rebates for the service of liquidity providing

High Frequency Trading

- Employ high-frequency data analysis to exploit market mispricings (e.g., arbitrage)
- Electronic market making

- Characteristics
  - Typically generate a large number of turnovers in a single day
  - Holding periods: from several seconds to several minutes; sometimes even within few milliseconds
  - Many times compete for orders
  - Require low-latency trading systems
  - Exchange co-location

Trading Systems

- Asynchronous event processing
- Typical events
  - Market data arrive
  - Order status update
  - Trade notifications
- Typical actions
  - Place orders
  - Cancel orders
  - Modify orders
  - Maintain internal states and data structures

System Components

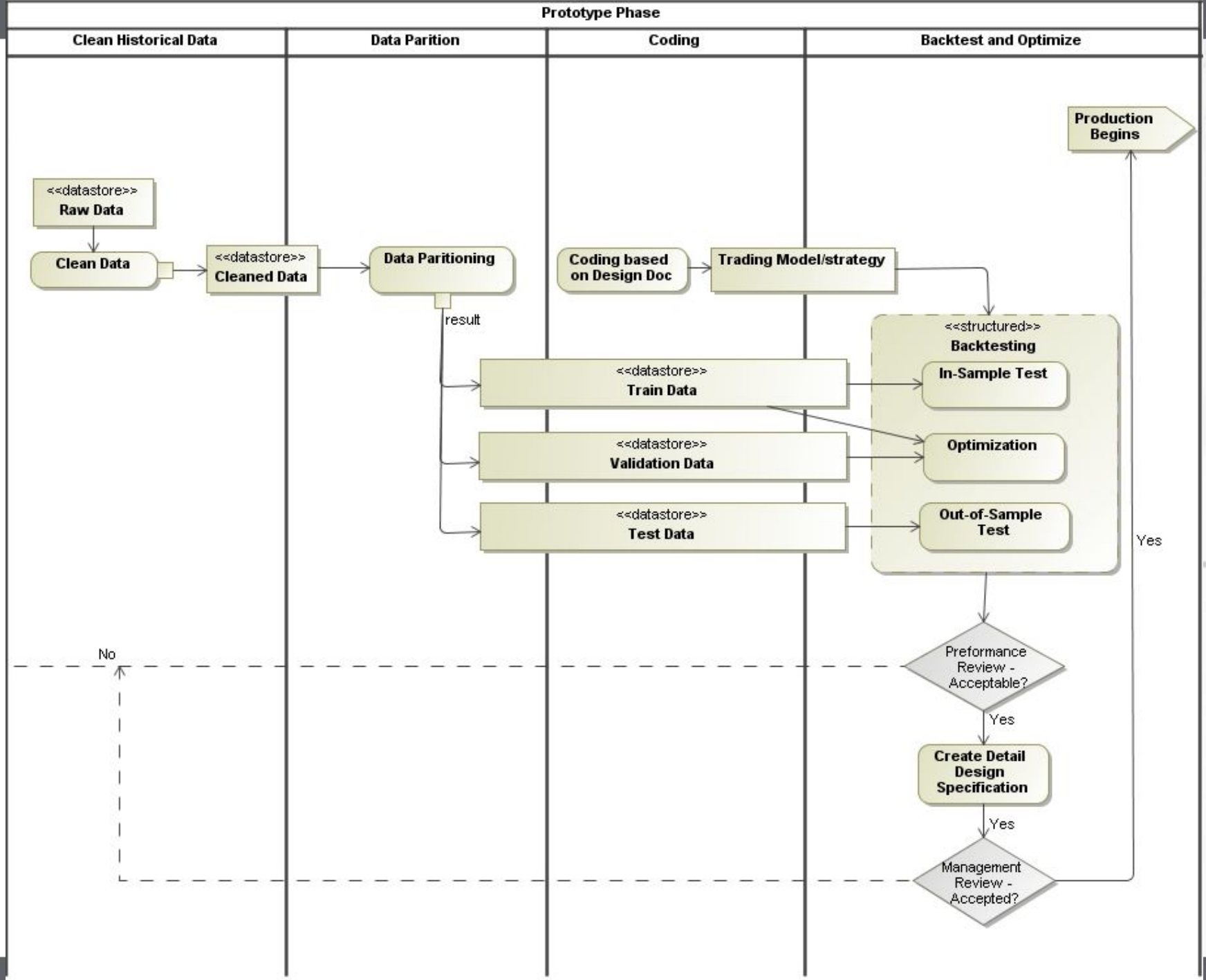
- Data Feed
  - Data cleaning (e.g., remove outliers)
  - Some calculations (create Open,High,Low,Close bars)
  - Historical data store and backfill
- Execution Engine
  - Order splitting
  - Order routing
- Risk Management System
  - Reject orders that may lead to significant risks
  - Reject orders that may violate compliances

Concurrency and Latency

- Multithreaded Proramming
- Garbage collection sometimes unfavorable
  - Latency spike
  - C++ is the typical choice
- I/O bottleneck
  - Optimized OS kernel
  - Specialized hardware (System on Chip)
  - FPGA for some routine modules (e.g., data feed)
- Networking
  - Dedicated lines between cities
  - Microwave

Databases

- Large amount of data
  - 3GB per day for Shenzhen Stock Exchange
  - Need data compression for storage and delivery
- Specialized database for high frequency data, e.g., kdb
- Key-value database can be useful


Prototype Phase
Clean Historical Data
Data Parition
Coding
Backtest and Optimize
Production Begins
<<datastore>>
Raw Data
Clean Data
<<datastore>>
Cleaned Data
Data Paritioning
result
Coding based on Design Doc
Trading Model/strategy
<<structured>>
Backtesting
<<datastore>>
Train Data
In-Sample Test
<<datastore>>
Validation Data
Optimization
<<datastore>>
Test Data
Out-of-Sample Test
Yes
No
Preformance Review - Acceptable?
Yes
Create Detail Design Specification
Yes
Management Review - Accepted?

Trading


THANK YOU!