Lecture 3

Market Data

Introduction to Trading Strategies

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Administrative

- Group submissions
  - Extended to this Friday

- Assignment 1:
  - Will be published this weekend due in ~2 weeks
  - Will probably include:
    - Question on market structure
    - Question on Order Book logic and order types
    - Market data (bar and tick) acquisition and processing
    - Etc.

Lecture 3 Agenda

- Orderbooks
- Orderbooks: Dark Pools continued
- Market Data
  - Types and uses
  - Sources
- Introduction to Execution Strategies

Dark Pools - Continued

Bid Qty	Bid Px		Offer Px	Offer Qty
200,000	10		10.01	100,000
250,000	9.99		10.02	150,000
340,000	9.98		100.03	90,000
125,000	9.97		100.04	4,500,000

1. Orders are not displayed
2. Intended to help buy side seek out large blocks of liquidity, but the truth is more complicated than that.
3. Generally Only match at or inside the NBBO ("reference price")
4. Different Pools have different price improvement rules
5. Orders are generally not routed to other venues (you don't usually want them to).
6. Usually support market, limit and "midpoint pegged" orders.
7. Orders will trade whenever they are crossed inside NBBO.

Dark Pool: Symmetric Price Improvement

Bid Qty	Bid Px		Offer Px	Offer Qty
200,000	10		10.01	100,000
250,000	9.99		10.02	150,000
340,000	9.98		100.03	90,000
125,000	9.97		100.04	4,500,000

If Price Improvement is split between both orders:

1. Given the above order book,
2. A “midpoint pegged” buy order is sent ($10.0050).
3. The seller enters a marketable sell limit order at $10.00.
4. The orders will match at $10.0025, splitting the price improvement equally between both orders.

Dark Pool: Asymmetric Price Improvement

Bid Qty	Bid Px		Offer Px	Offer Qty
200,000	10		10.01	100,000
250,000	9.99		10.02	150,000
340,000	9.98		100.03	90,000
125,000	9.97		100.04	4,500,000

If Price Improvement is not split between both orders:

1. Given the above order book,
2. A “midpoint pegged” buy order is sent ($10.0050).
3. The seller enters a marketable sell limit order at $10.00.
4. Depending on the order book rules, the orders may match at $10.00, or 10.005, giving all the price improvement to the buyer or seller, respectively.

Initial Order Book Analysis

Order Book Analysis and Metrics

Bid Qty	Bid Px		Offer Px	Offer Qty
2100	100		100.01	1000
1000	99.99		100.02	1500
999	99.98		100.03	900
500	99.97		100.04	450

What kind of analysis can we do on an order book?

Some Simple Metrics

Order Book Metrics

- Bid / Offer size imbalance
- Location in spread
- Quote turnover
- Book depth
- Volatility of spread

Execution Metrics using Order Book Data (for the market or your algo)

- % of spread
- Price Improvement

Bid / Offer Size Imbalance

Order imbalance = *ask size – bid size*

- Quantity or Notional
- Possible uses
  - Trade signal when absolute imbalance >= some threshold value
  - Monitor change in absolute imbalance
  - Monitor change in directional imbalance
  - May be coupled with monitoring execution side (bid or offer

Location in spread

$$\text{Location in spread} = \frac{ask\ px - trade\ px}{(ask\ px - bid\ px)}$$

- Location < 0.5 → bid side (location = 0 → offer)
- Location > 0.5 → offer side (location = 1 → bid)

- Calculate the volume weighted average of the location over some rolling window to measure if there is consistent directionality over time
- May be coupled with time series evaluation of price evolution

% of Spread / Price Improvement

We like to analyze execution placement for a variety of reasons, including routing efficiency and performance. One way to do this is to calculate execution price as a % of the spread (i.e. how aggressive or passive were your trades in reality?):

Similar to location in spread, but we know what side the order was :

Calculate side-adjusted % of spread:

if sell:

$$\% \text{ of spread} = \frac{trade\ px - bid\ px}{(ask\ px - bid\ px)}$$

if buy

$$\% \text{ of spread} = \frac{ask\ px - trade\ px}{(ask\ px - bid\ px)}$$

Take volume weighted average over a rolling window

May need to merge trade and quote data

Quote Turnover

How often does the quote update relative to the volume traded

For some time interval, calculate:

Total quote change in shares / total shares traded

Book Depth

Another measure of activity (and potentially imbalance) is book depth:
What is the price difference from the top of the book to some total shares n?

Bid Depth: top of book px – bottom of book px

Ask Depth: top of book px – bottom of book px

May scale by quantity

Cons: Requires you to maintain complete order book, not just top of book.

Time Series Market Data
An Introduction

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Types of Market Data

1. Bar Data – aggregated market data taken at intervals

2. Tick data – lowest level of market data
   - Quotes – Orders sent to order books
   - Trades – Records of trades that occur when orders cross in the order book

Bar Data

- Aggregated over some time interval
- Large Scale: daily, weekly, monthly, quarterly, etc.
- Intraday: 1-minute, 5-minute, 10-minute, etc.
- Important data points:
  - Open
  - High
  - Low
  - Close
  - Volume
- Useful for analyzing trends, evaluating alpha processes of a certain scale.
- Limited utility for true microstructure work
- You can always create lower resolution (i.e. larger bars) from more granular data, but not vice versa.

Tick Data

- Tick data literal change-by-change data, in which each data point reflects a single (visible) update to an order book:
  - Bid size change
  - Bid price change
  - Offer size change
  - Offer price change
  - Transaction

- Data format and file layouts will vary by product and vendor, but these are general principles.

- Market data may also include different kinds of messages (market status, etc.)

Market Data Sources

Recommended for class

- https://www.alphavantage.co: free bar data – register for an account and use the API (see example code)

- https://wrds-web.wharton.upenn.edu/wrds/ -> TAQ (Trade and Quote)

Note your data source in assignment and project credits

Market Data - Examples

Demo: getting data from AlphaVantage.co

See `getstock.py`

Demo: getting data from Wharton WRDS

See `Working with Bar Data.ipynb`

See `Working with Tick Data Part 1.ipynb`

Building an Order Book from Tick Data

1. Establish variables / data structures for tracking
   - Top of book or full book depth?
   - Bid Px
   - Bid Qty
   - Offer Px
   - Offer Qty
   - Last Trade Px
   - Last Trade Qty
2. Collect quote and tick data
   - Ensure data is time aligned
   - If you have to merge tick and quote data from separate files / sources, leverage the indexing functions of pandas DataFrame to make merges easier.
3. Main Event Loop
   a. Read a line
   b. Update mkt data accordingly
   c. Process the ”tick event” – calculating other metrics as needed
4. Calculate summary stats and output

<u>This is step 1 in any alpha or execution strategy simulation.</u>

A Simple Orderbook Simulation in Python

L2.1 Order Book Simulator-v1.ipynb

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Execution Strategies: A Taxonomy Types

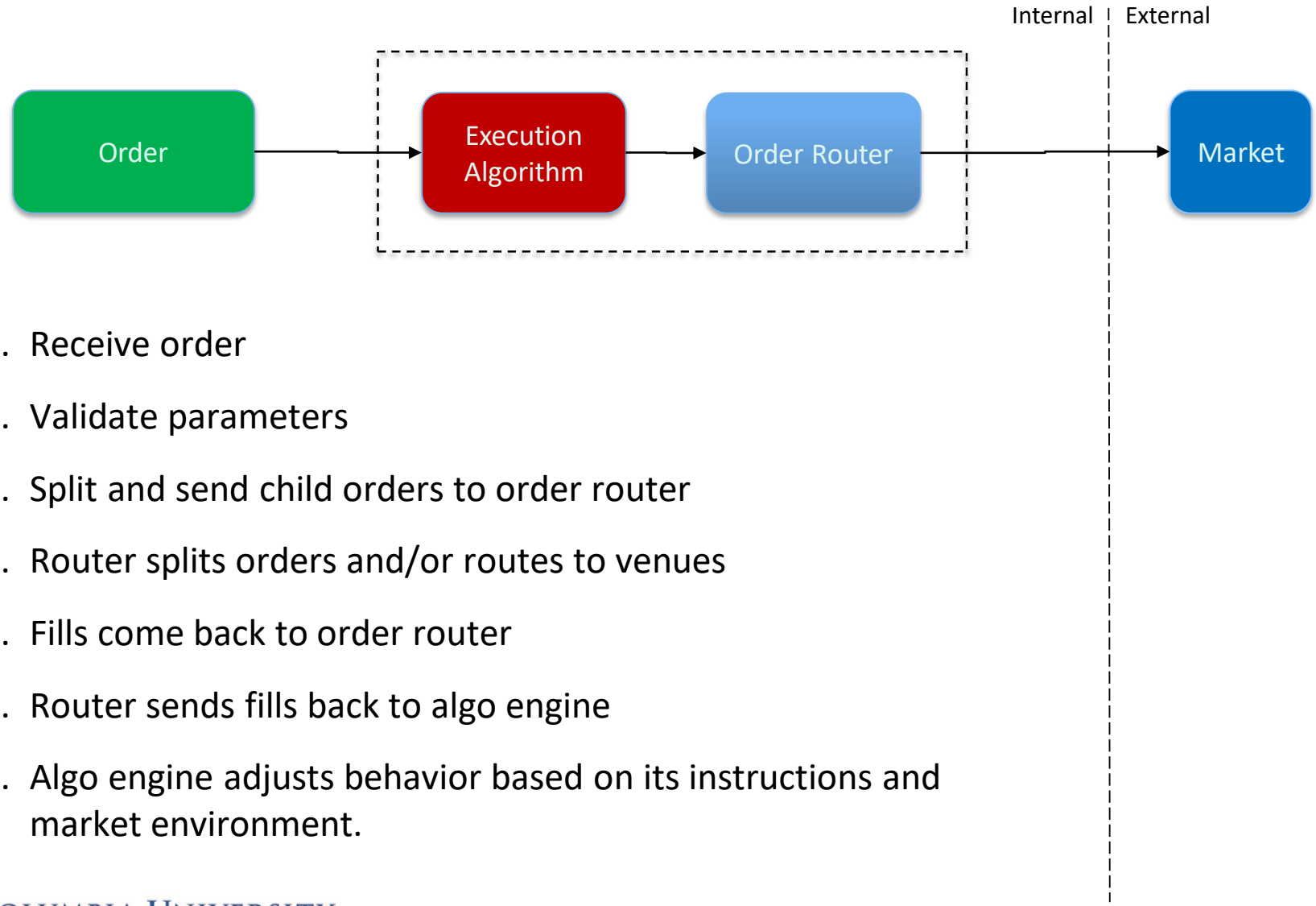
Trading Strategy Taxonomy

- Execution: Given instructions (buy/sell, instrument, price, quantity, style/benchmark), achieve an execution objective.
  - Execution "Algorithm" – when to trade, achieve a benchmark
  - Smart Order Router – where to trade, choose between trading venues
- Market-making: concurrently place orders to buy and sell a security
  - Capture spread
  - Earn rebates
  - For our purposes Indications of Interest (IOIs) also fall into this category
- Hedging: Given some risk metric trade to reduce said risk
  - For example the delta of a portfolio of equities
  - Buy delta when portfolio is short delta, sell delta when portfolio is long
- Multi-instrument strategies (pairs, portfolios)
  - Trade multiple securities concurrently
  - Maintain a relationship, e.g. $ neutrality (similar to hedging)
  - Statistical arbitrage, e.g. buy/sell securities when the relationship diverges from the expected
- "Alpha" strategies: some capability to forecast asset prices, either directly or in relationship to others. In our taxonomy these are purely "pricing" and still require execution.

Execution Workflow

1. Trader receives order and instructions from PM
2. Select algorithm
3. Select parameters
4. Submit order
5. Algo works order, sends child orders to market
6. Receive executions back
7. Monitor performance in real time
8. Monitor full post trade performance and historical performance.

Trading Data Ecosystem



1. Receive order
2. Validate parameters
3. Split and send child orders to order router
4. Router splits orders and/or routes to venues
5. Fills come back to order router
6. Router sends fills back to algo engine
7. Algo engine adjusts behavior based on its instructions and market environment.

Execution Objectives Restated

- Minimize impact: avoid signaling your intent to the market
- Achieve a benchmark (PM determined)
  - ✓ VWAP, TWAP, Open, Close, PWP, etc.
  - ✓ Benchmark used to distinguish alpha from execution
- Manage explicit execution costs
  - ✓ Minimize fees
  - ✓ Maximize rebates
- Portfolio trade
  - ✓ Maintain neutrality or specific skew (cash, sector)
  - ✓ Raise cash

Execution Strategy Types

- Direct Market Access (DMA)
- Order Routers: “Smart Order Routers”, Dark liquidity aggregators
- Schedule-Based: VWAP, TWAP
- Volume-Following: Percent of Volume
- Implementation Shortfall: minimize difference from arrival price by balancing market impact and opportunity cost
- Event Driven / Liquidity Seeking: Sensitive to spread, order book size, may use quantitative models
- Pairs
- Higher-Order Activities
  - ✓ Portfolio
  - ✓ Hedging, Market Making
  - ✓ Statistical Arbitrage

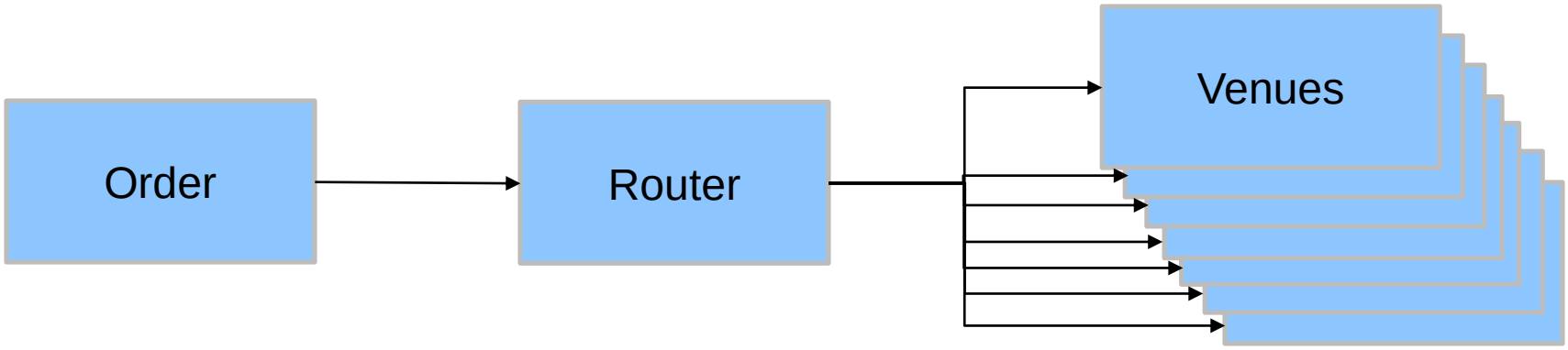
Direct Market Access (DMA)

Objectives

- Send a single order to a single market venue
- Simple, Fast Order Placement

- Pros
  - Fast access to market
  - Simple
- Cons
  - Trader must select venue
  - Practical limits to order size
  - RegNMS (in US)

Order Router



"Smart Order Routers"

Dark Liquidity Aggregators

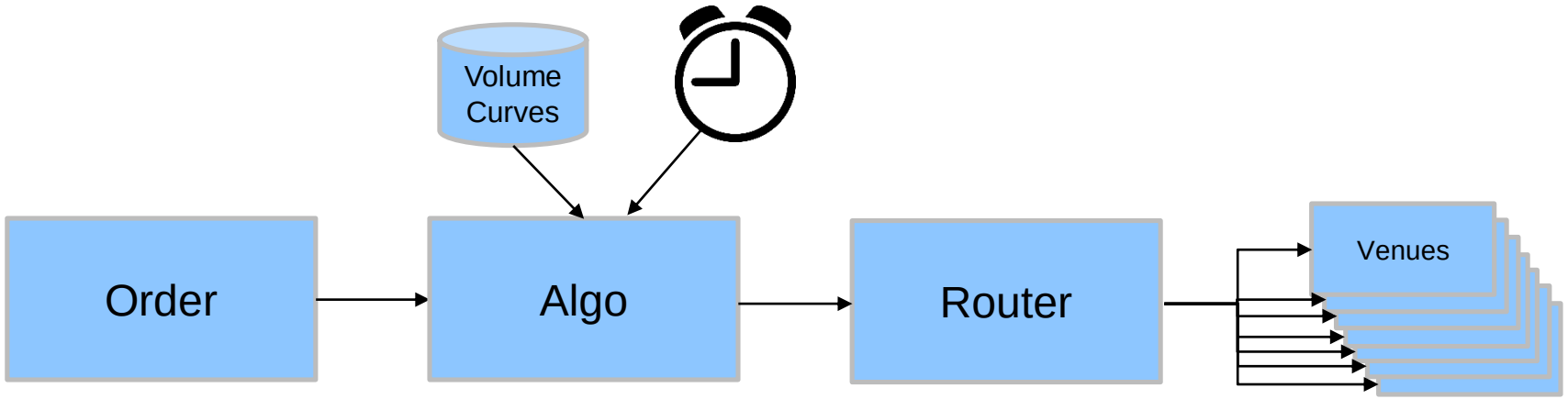
Objectives

- Send a single order to multiple market venues
- Fast "Optimal" Order Placement

Splits order into multiple child orders

- Pros
  - Venue Access Included
  - Handles Issues like RegNMS
- Cons
  - Trader must select venue
  - Practical limits to order size

Schedule Based

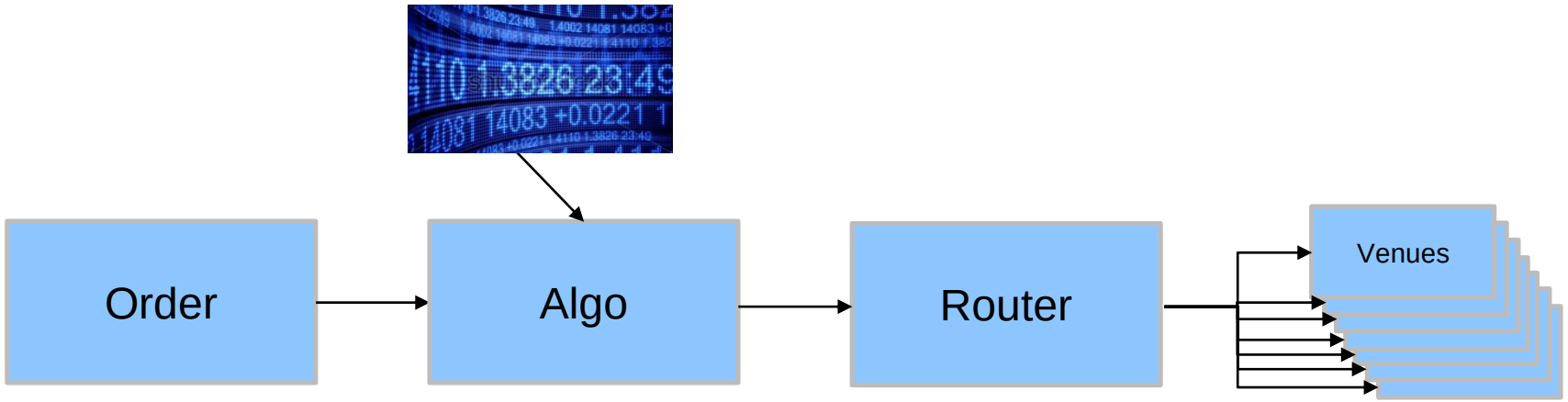


Objective: Match or beat a time-based benchmark, e.g. TWAP or VWAP

- Separate algo and router
- Typical Parameters
  - Start, End time
  - Limit price
  - Volume cap

- Splits order over time based on a schedule
- Pros
  - Handles much larger orders
  - Reduced impact
- Cons
  - Opportunity cost
  - Not reactive

Volume Following

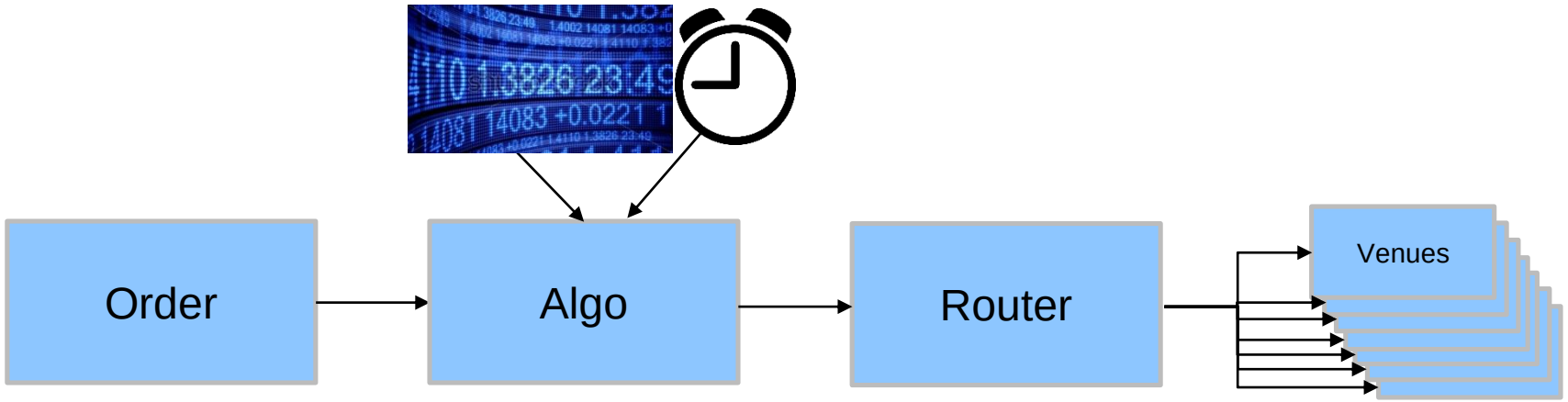


Objective: participate in market flow at some predetermined rate

- Typical Parameters
  - Min % of Volume
  - Max % of Volume
  - Limit Price

- Pros
  - Handle large orders
  - Responsive to volume changes
  - "No one ever got fired for following VWAP"
- Cons
  - Somewhat responsive but still dumb
  - Opportunity cost

Implementation Shortfall



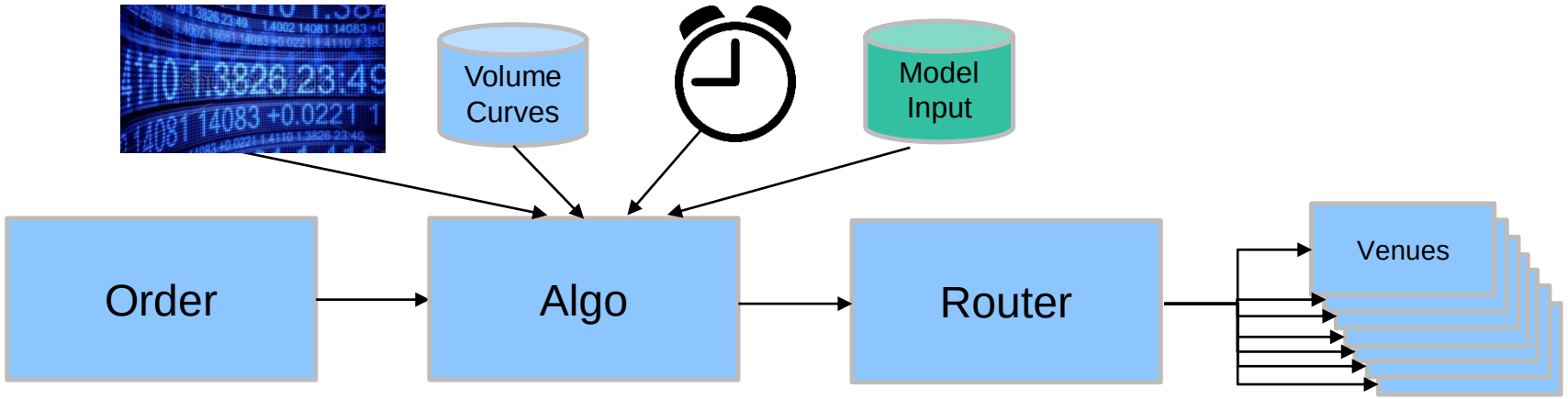
Objective: balance market impact and opportunity cost, benchmark = arrival price

- Typical Parameters
  - "Urgency" – a static set of rules that range from passive to aggressive by modulating participation rate and other parameters
  - Limit Price
  - Max % Volume

- Pros
  - Handles Larger orders
  - Fine tuned performance

- Cons
  - Rules based behavior smarter but still not responsive to market conditions
  - Inflexibility often requires customization

Event Driven / Dynamic

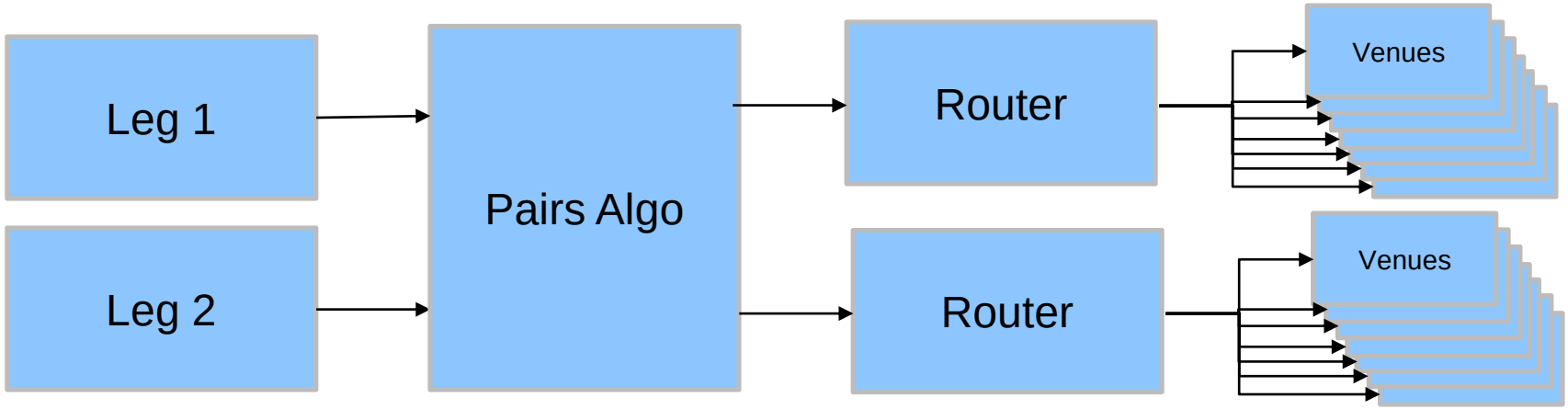


Objective: Optimize execution for <u>current market conditions</u>

- Typical Parameters
  - Urgency – define a range of behaviors based on certain parameters
  - Max % of Volume
  - Spread sensitivity
  - Liquidity Sensitivity
  - Limit Price

- Pros
  - Responsive to volatile conditions
  - Good for illiquid names
  - Arrival price benchmarking
  - Large orders
- Cons
  - Behaviors less obvious
  - Decomposing performance can be more difficult

Pairs



Objective: trade two securities in parallel, maintaining some relationship

Example:

Buy $50K GOOG

Sell $50K AMZN

Uses

- Transition between Securities
- Arbitrage

Example Relationship / Strategies

- $ Neutrality: trade equal amounts over time until complete
- Spread: only trade when the spread between them

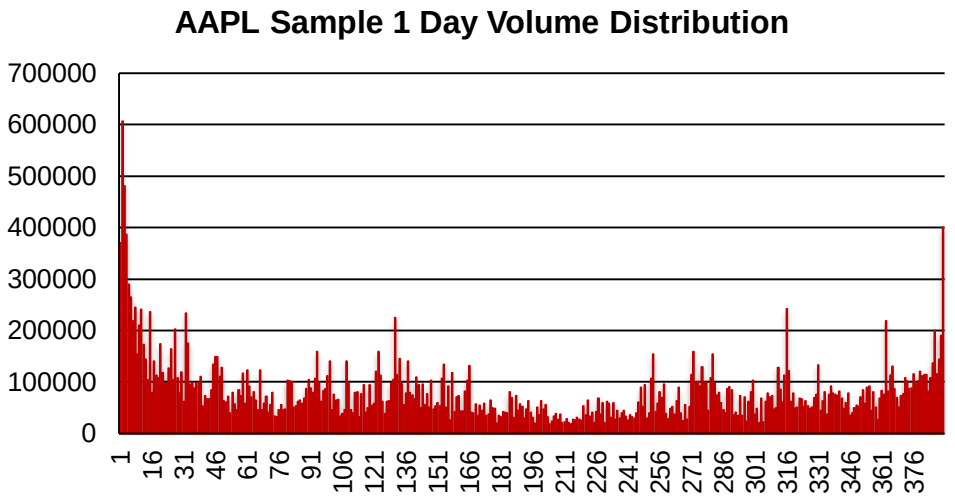
VWAP: Volume Distribution Profile

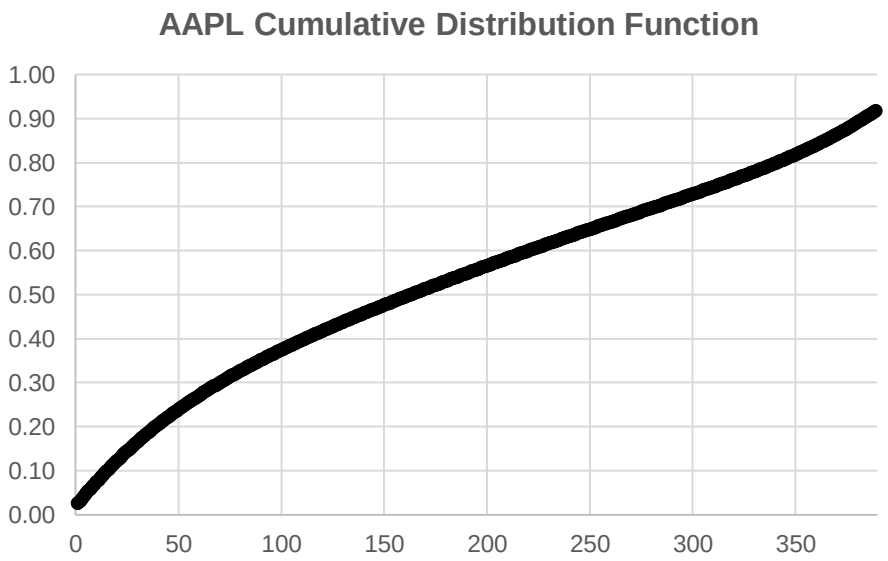
1. Why? You can't control price, but you can predict **volume.**

2. The distribution of volume over the course of a trading day in the US is quite stable.

3. It can be used to estimate the proportion of volume that will trade *on average* in a given segment of the day.

4. Using the profile as a guide, you can apply other techniques to improve order placement and volume prediction





COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

A Basic VWAP Algo

Target the Volume Weighted Average Price p_{VWAP} :

$$p_{VWAP} = \frac{\sum(quantity_{@price} \times price)}{total\ quantity}$$

At any point in time, predict what % of cumulative volume will trade and attempt to match:

1. Calibrate a Cumulative Volume Function

2. Target Volume = volume(q, t) where q = total order quantity and t = time segment

3. Target order size at time *t* = Target volume at time *t+n* – Executed volume

4. Example Primitive Order Placement Strategy
   - Define Shortfall => Target volume at time *t* – Completed Volume
   - Threshold value is the tolerance for being off target volume distribution
   - Passive if Shortfall <= threshold value
   - Aggressive if Shortfall > threshold value

Building a Volume Profile - Data

AAPL US Equity
BarTp=T

Date	OPEN	HIGH	LOW	LAST_PRICE	NUMBER_TICKS	VOLUME	VALUE
9/15/16 9:30	113.86	114.25	113.49	114.09	6,141	2,531,717	288,265,408
9/15/16 9:31	114.099	114.35	114	114.1099	5,341	1,390,148	158,717,056
9/15/16 9:32	114.1	114.28	113.72	113.7519	3,637	948,012	108,137,632
9/15/16 9:33	113.8	114.01	113.74	113.91	2,827	640,697	72,981,640
9/15/16 9:34	113.9001	114.2	113.84	114.17	2,418	580,885	66,263,004
9/15/16 9:35	114.17	114.4	114.17	114.252	3,742	858,753	98,137,392
9/15/16 9:36	114.26	114.47	114.21	114.435	2,945	744,893	85,186,472
9/15/16 9:37	114.445	114.65	114.43	114.554	3,938	939,103	107,570,568
9/15/16 9:38	114.569	114.73	114.51	114.61	3,243	773,059	88,624,216
9/15/16 9:39	114.62	114.62	114.3	114.3799	2,071	529,009	60,537,704
9/15/16 9:40	114.37	114.51	114.33	114.36	1,833	450,583	51,565,112
9/15/16 9:41	114.35	114.4	114.1799	114.35	2,638	668,989	76,453,024
9/15/16 9:42	114.36	114.43	114.34	114.34	1,544	356,846	40,817,200
9/15/16 9:43	114.34	114.36	114.19	114.35	2,083	427,851	48,893,000
9/15/16 9:44	114.345	114.36	114.13	114.26	2,261	434,212	49,595,816
9/15/16 9:45	114.26	114.26	114	114.01	2,174	536,700	61,239,476

Date	bin	VOLUME	CUM_VOL	PCT
9/15/16 9:30	1	2,531,717	2,531,717	0.03
9/15/16 9:31	2	1,390,148	3,921,865	0.05
9/15/16 9:32	3	948,012	4,869,877	0.06
9/15/16 9:33	4	640,697	5,510,574	0.07
9/15/16 9:34	5	580,885	6,091,459	0.07
9/15/16 9:35	6	858,753	6,950,212	0.08
9/15/16 9:36	7	744,893	7,695,105	0.09
9/15/16 9:37	8	939,103	8,634,208	0.1
9/15/16 9:38	9	773,059	9,407,267	0.11
9/15/16 9:39	10	529,009	9,936,276	0.12
9/15/16 9:40	11	450,583	10,386,859	0.13
9/15/16 9:41	12	668,989	11,055,848	0.13
9/15/16 9:42	13	356,846	11,412,694	0.14
9/15/16 9:43	14	427,851	11,840,545	0.14
9/15/16 9:44	15	434,212	12,274,757	0.15

1. Decide on data requirements
   - Bar size (1 minute in this example)
   - Number of days of data
2. Get some data
   - Minute timestamp
   - Volume per minute
3. Prepare the data
   - Convert timestamp to bin (1-390 for US Stock Market)
   - Calculate Cumulative Volume
   - Normalize Cumulative Volume -> Total volume = 1%

Using the Volume Profile - Model

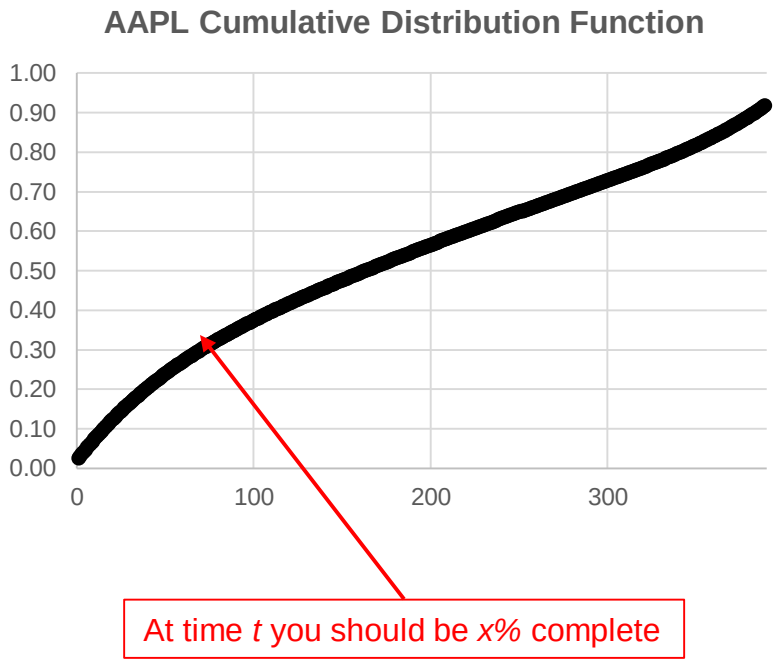
Now let's use it:

Order Placement

1. Target Volume = volume_at_time(q, t) where q = total order quantity and t = time segment

2. Target order size at time *t* = Target volume at time *t+n* – Executed volume

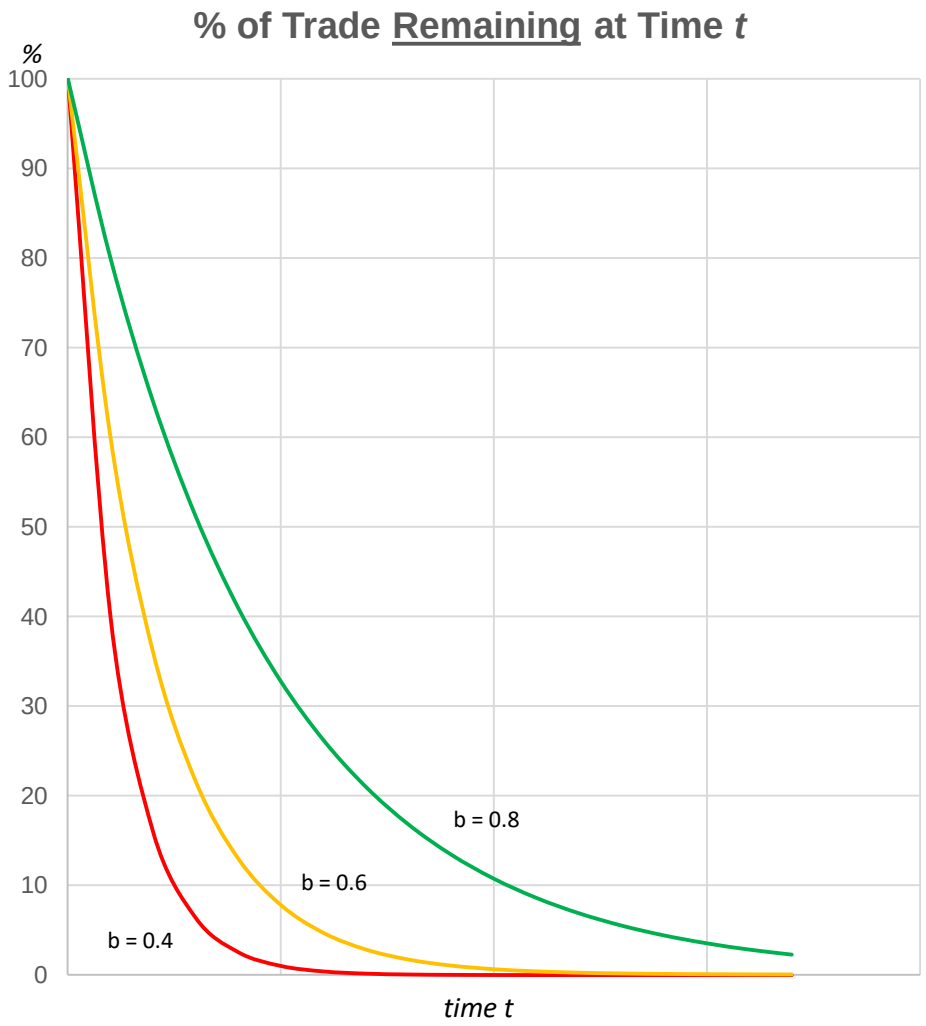
3. Example Primitive Order Placement Strategy
   - Define Shortfall => Target volume at time *t* – Completed Volume
   - Threshold value is the tolerance for being off target volume distribution
   - Passive if Shortfall <= threshold value
   - Aggressive if Shortfall > threshold value
4. Depending on size of shortfall, leverage non-displayed orders, etc.



Implementation Shortfall

Opportunity cost / impact trade off

- Common parameters
  - Volume caps
  - Aggressiveness / patience (tuning the tradeoff)
- Models vs. heuristics
- Cleanup
  - Order has to finish eventually!
  - Size Thresholds
  - Minimum % volume



where

$a=100, 0 <= b <= 1, t = time$

Next Week

- Developing our Execution Algo Simulator
- A First Execution Algo Example