อ่านกระดาน
Reading the Book

Order Flow & Microstructure สำหรับรายย่อย — ฉบับรวมเล่ม 17 ตอน

เลนส์ execution ของซีรีส์ · เทรด "ตรงไหน / เมื่อไหร่ / ต้นทุนจริงเท่าไหร่"

5 องก์ + ภาคเครื่องมือ · ตัวอย่างคริปโต (Binance/Bybit L2 ฟรี)

สารบัญ

1. Part 0 — กระดานคืออะไร
2. Part 1 — กายวิภาคของคิว
3. Part 2 — 3 เหตุการณ์ที่ขยับกระดาน
4. Part 3 — Bid-Ask Spread แยกส่วน
5. Part 4 — กระดานเชื่อถือได้แค่ไหน
6. ภาคทบทวน — สถิติพื้นฐาน
7. Part 5 — Order Book Imbalance (OBI)
8. Part 6 — ทำไม Market Order สำคัญ
9. Part 7 — OFI & Price Impact
10. Part 8 — VPIN ความเป็นพิษของ flow
11. Part 9 — Market Making พื้นฐาน
12. Part 10 — Optimal Market Making
13. Part 11 — ลงมือจริงในคริปโต
14. Part 12 — Grid Trading
15. Part 13 — Statistical Arbitrage ด้วย OFI
16. Part 14 — Cross-Exchange Arbitrage
17. Part โบนัส — เครื่องจักรอ่านกระดาน

อ่านกระดาน
Reading the Book

Part 0: กระดานคืออะไร — "ราคา" จริง ๆ มาจากไหน

Order Flow & Microstructure สำหรับรายย่อย · เลนส์ execution ของซีรีส์
อ่านจบแล้วเข้าใจ bid / ask / spread / mid และอ่านกระดานคริปโตเป็น

Part 0 — กระดานคืออะไร

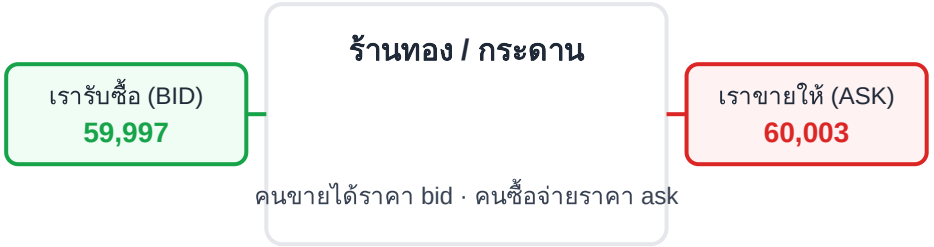
🎯 **อ่านจบ Part นี้คุณจะ**

- เข้าใจว่า "ราคา" บนจอจริง ๆ มาจากการ **จับคู่คำสั่งซื้อขาย** ในกระดาน ไม่ใช่ตัวเลขลอย ๆ
- แยก bid / ask / spread / mid ออกจากกัน และอ่านค่าจากกระดานจริงได้
- รู้ว่า "ไม่มีราคาเดียว" — มีราคาที่ซื้อได้ (ask) กับราคาที่ขายได้ (bid) เสมอ
- แยก order-driven market (คริปโต/หุ้น) กับ quote-driven market (มี dealer ขานราคา)

เริ่มจากร้านทอง

ลองนึกถึงร้านทองในตลาด — หน้าร้านเขียน **2 ราคาเสมอ**: ราคาที่ร้าน "รับซื้อ" จากคุณ (ต่ำกว่า) กับราคาที่ร้าน "ขายให้" คุณ (สูงกว่า) ส่วนต่างนั้นคือกำไรของร้าน

กระดานเทรด (order book) ก็คือร้านทองที่มีคนนับพันมายืนเสนอราคาพร้อมกัน — และคุณเลือกได้ว่าจะเป็น "คนตั้งป้ายราคา" หรือ "คนเดินเข้าไปรับราคา"



ส่วนต่าง (spread) = 6 = "ค่าผ่านทาง" ของการเทรดทันที

กระดานคือบัญชีคำสั่งที่ยังไม่ถูกจับคู่

ทุกคนส่ง "ความตั้งใจจะซื้อ/ขาย" เข้ามา ระบบเรียงไว้เป็นกระดาน:

- **ฝั่งซื้อ (bids)** — คนอยากซื้อ เรียงจากราคาแพงสุดเข้าหากึ่งกลาง (สีเขียว)
- **ฝั่งขาย (asks / offers)** — คนอยากขาย เรียงจากราคาถูกสุดเข้าหากึ่งกลาง (สีแดง)

ภาพ 0.1 · กระดาน LOB (เทมเพลตแม่ของทั้งเล่ม) — เป็นภาพ *schematic* เพื่อสื่อแนวคิด ความสูงของแท่งไม่ใช่โปรไฟล์ depth จริง

ศัพท์ที่ต้องจำ

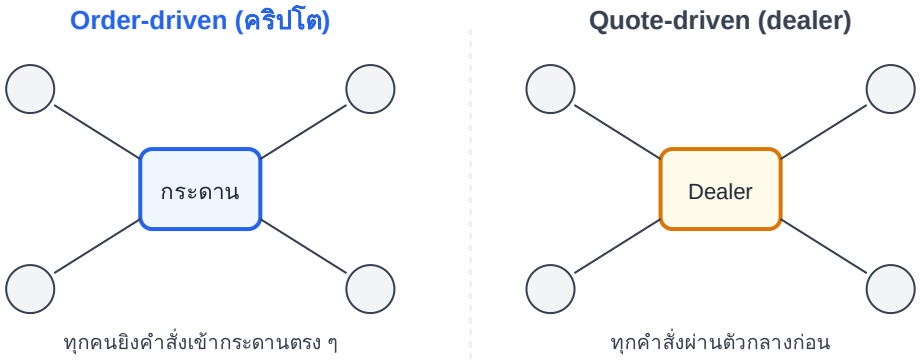
ศัพท์	ความหมาย
Best bid	ราคาซื้อที่แพงสุด (เขียวที่ใกล้กึ่งกลางสุด) = 59,997
Best ask	ราคาขายที่ถูกสุด (แดงที่ใกล้กึ่งกลางสุด) = 60,003
Spread	ช่องว่างระหว่างสองฝั่ง = "ค่าผ่านทาง" เข้า-ออกทันที = 6
Mid	จุดกึ่งกลาง = ราคาอ้างอิงที่ยุติธรรมสุด (แต่เทรดที่ราคานี้จริง ๆ ไม่ได้)

```
Spread = BestAsk − BestBid = 60,003 − 59,997 = 6 USDT
Mid Price = (BestAsk + BestBid)/2 = (60,003 + 59,997)/2 = 60,000
Spread (bps) = Spread / Mid × 10,000 = 6 / 60,000 × 10,000 = 1.0 bps
```

> ⚠️ **กับดักมือใหม่ — "ราคาเดียว" ที่ไม่มีจริง**
>
> มือใหม่มักคิดว่า "ราคา BTC ตอนนี้ = 60,000" เป็นตัวเลขเดียว — **ผิด!** ถ้าจะ "ซื้อทันที" คุณจ่าย best ask (60,003) ถ้าจะ "ขายทันที" คุณได้ best bid (59,997) ตัวเลข 60,000 ที่เว็บแสดงมักเป็น last trade หรือ mid ซึ่งคุณเทรดที่ราคานั้นไม่ได้ — **การเอา last price ไปคิดกำไรแล้วลืม spread ทำให้ตอนปิดจริงขาดทุนกว่าที่คิด**

Order-driven vs Quote-driven

คริปโตเป็น order-driven แท้ ๆ — ไม่มี dealer กลาง ทุกคน (รวมคุณ) ช่วยกันสร้างกระดาน ส่วนตลาดแบบเก่าบางที่เป็น quote-driven มี dealer คอยขานราคาให้



📑 งานวิจัยรองรับ

Glosten & Milgrom (1985) วางรากฐานว่า "spread มีอยู่เพราะผู้ตั้งราคาไม่รู้ว่าคนที่มาเทรดมีข้อมูลลับหรือไม่" — ราคาสองฝั่งจึงเป็นกลไกป้องกันตัวของตลาด ไม่ใช่แค่ค่าธรรมเนียม (จะลงลึกใน Part 3 เรื่ององค์ประกอบของ spread)

🔗 เชื่อมโยงเล่มอื่น

Mid price ที่เรียนตรงนี้คือ "underlying price" เดียวกับที่ใช้คำนวณ payoff ในเล่ม **Payoff Mastery (pm)** — เวลาคิดกำไรขาดทุนจริง อย่าลืมว่าราคาเข้า-ออกไม่ใช่ mid แต่เป็น ask/bid (ต้นทุน spread กินกำไรจริง)

✍️ แบบฝึก (ทำได้ด้วยข้อมูลคริปโตฟรี)

เปิด Binance หน้า BTC/USDT order book ดู best bid และ best ask จดค่า แล้วคำนวณ **spread (USDT)** และ **spread (bps)** ด้วยมือ จากนั้นทำซ้ำกับเหรียญเล็ก (อันดับ 100+) แล้วเทียบว่า spread (bps) ต่างกันแค่ไหน — คุณจะเห็นทันทีว่าเหรียญเล็ก "ค่าผ่านทาง" แพงกว่ามาก

ต่อ Part 1: กายวิภาคของคิว — ทำไม "อยู่ต้นคิว" ถึงมีค่าเป็นเงิน →

อ่านกระดาน
Reading the Book

Part 1: กายวิภาคของคิว — ทำไม "อยู่ต้นคิว" ถึงมีค่าเป็นเงิน

Order Flow & Microstructure สำหรับรายย่อย · เลนส์ execution ของซีรีส์

เจาะลึกราคาเดียวว่ามีคิวซ่อนอยู่ — price-time priority, tick, depth และมูลค่าของการได้ยืนต้นคิว

Part 1 — กายวิภาคของคิว

🎯 **อ่านจบ Part นี้คุณจะ**

- เห็นว่า "ราคาหนึ่งราคา" บนกระดานจริง ๆ เป็น **คิวของคำสั่งหลายคน** ไม่ใช่ตัวเลขก้อนเดียว
- เข้าใจกฎจัดลำดับ **price-time priority (FIFO)** — ราคาดีกว่าได้ก่อน ราคาเท่ากันใครมาก่อนได้ก่อน
- อ่าน **tick size** และ **depth** (ความลึกหลายชั้น) ออกจากกระดานได้
- เข้าใจ **queue value** — ทำไมการ "อยู่ต้นคิว" จึงแปลว่าได้ fill ก่อนและมีมูลค่าเป็นเงิน

เริ่มจากคิวซื้อ iPhone รุ่นใหม่

ตอน iPhone รุ่นใหม่เปิดขาย คนไปต่อแถวหน้าร้านตั้งแต่ตี 4 — **ทุกคนยอมจ่ายราคาป้ายเท่ากัน** แต่ "ใครได้เครื่องก่อน" ขึ้นกับตำแหน่งในแถว คนที่มาก่อนยืนต้นแถว ได้ของชัวร์ คนท้ายแถวอาจอดถ้าของหมด

กระดานเทรดก็มีแถวแบบนี้ในทุกราคา — ที่ราคา 59,997 อาจมีคนตั้งซื้อรออยู่ 50 คน เรียงตามเวลาที่ส่งคำสั่งเข้ามา เมื่อมีคนขายทันที (market sell) เข้ามา "ตัดของ" คนต้นแถวได้ขายของก่อนเสมอ

ในหนึ่งราคา มี "คิว" ซ่อนอยู่

กระดานที่เราเห็นแบบสรุปจะบอกแค่ "ที่ราคา 59,997 มี 12 BTC รออยู่" — แต่ 12 BTC นั้นจริง ๆ คือคำสั่งของหลายคนเรียงต่อกัน เช่น 3 + 5 + 1 + 2 + 1 ตามลำดับเวลา ระบบจับคู่จะ "กิน" จากหัวคิวเสมอ

```
ราคา 59,997 (best bid): [3 BTC][5 BTC][1 BTC][2 BTC][1 BTC] รวม = 12 BTC
                        หัวคิว ──────────────────────────▶ ท้ายคิว
```

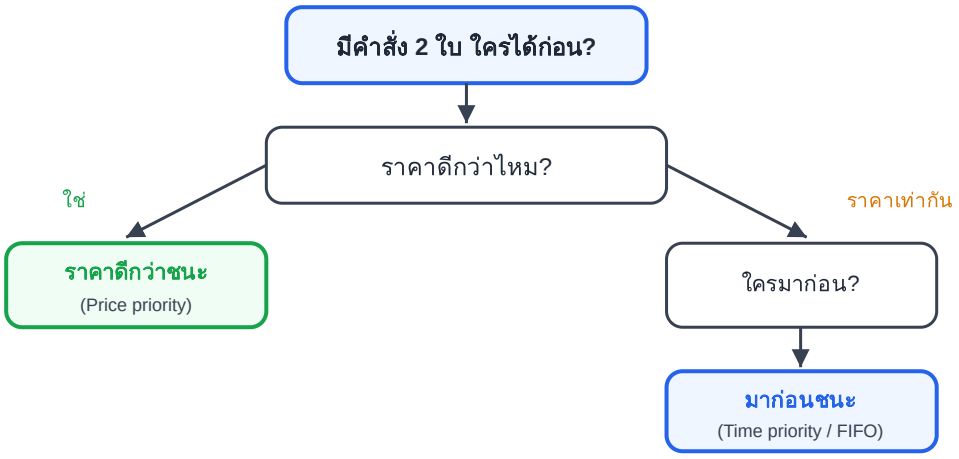
```
market sell 4 BTC เข้ามา → กิน [3] หมด + กิน 1 จาก [5] → เหลือ [4][1][2][1] =
8 BTC
```

ภาพ 1.1 · คิวภายในหนึ่งราคา (zoom-in)

กฎจัดลำดับ: Price-Time Priority (FIFO)

ระบบจับคู่ของกระดาน (matching engine) ใช้กฎ 2 ชั้นในการเลือกว่าจะให้ใครได้ fill ก่อน:

1. **Price priority (ราคาก่อน):** ฝั่งซื้อ ราคาแพงกว่าได้ก่อน · ฝั่งขาย ราคาถูกกว่าได้ก่อน — เพราะเป็นราคาที่ "ดึงดูด" คู่ค้ามากที่สุด
2. **Time priority (เวลาเป็นตัวตัดสินเมื่อราคาเท่ากัน):** ที่ราคาเดียวกัน ใครส่งคำสั่งมาก่อนอยู่ต้นคิวก่อน = FIFO (First In, First Out)



ภาพ 1.2 · ผังตัดสินใจ price-time priority

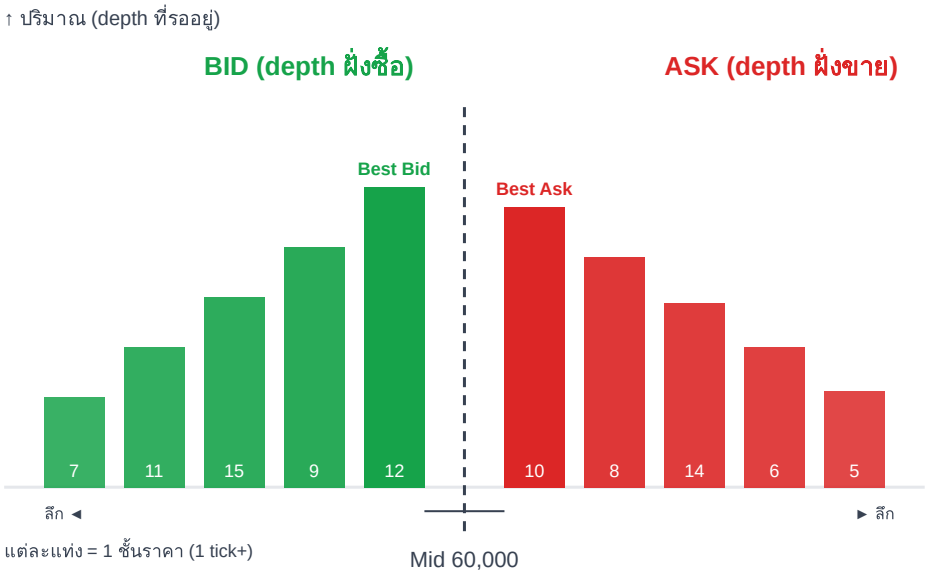
> 📌 **ข้อสังเกต — ไม่ใช่ทุกตลาดใช้ FIFO ล้วน**
>
> คริปโตสปอตส่วนใหญ่ (Binance/Bybit) ใช้ **price-time priority (FIFO)** ตรง ๆ แต่ตลาดอนุพันธ์บางแห่งใช้ **pro-rata** (แบ่ง fill ตามสัดส่วนขนาด ไม่ใช่ตามเวลา) ในเล่มนี้เรายึด FIFO เป็นหลักเพราะเป็นกติกาของกระดานคริปโตที่รายย่อยเจอจริง

Tick size และ Depth

Tick size คือ "ขั้นต่ำสุด" ที่ราคาขยับได้ — เช่น BTC/USDT อาจมี tick = 0.1 USDT แปลว่าราคาไปได้ทีละ 60,000.0 → 60,000.1 → 60,000.2 ไม่มีราคาแทรกกลาง tick เล็ก = ราคาละเอียด คิวกระจายหลายชั้น · tick ใหญ่ = ราคาหยาบ คิวอัดแน่นในไม่กี่ชั้น (สำคัญมากใน Part 5 เรื่อง large-tick)

Depth คือปริมาณที่รออยู่ "ลึกลงไป" จาก best หลายชั้นราคา ไม่ใช่แค่ชั้นบนสุด — กระดานที่ลึกหมายถึงรับแรงซื้อขายก้อนใหญ่ได้โดยราคาไม่กระเด็น

```
Queue(P) = Σ ปริมาณของทุกคำสั่งที่ตั้งอยู่ที่ราคา P
Depth(N) = Σ ปริมาณรวมของ N ชั้นราคาแรกนับจาก best (ทั้งฝั่ง)

ตัวอย่าง bid 5 ชั้น (ตามภาพ 1.3): 7 + 11 + 15 + 9 + 12 → Depth(5) ฝั่งซื้อ = 54 BTC
```



ภาพ 1.3 · Depth profile 5 ชั้น (ต่อยอดเทมเพลตแม่กระดาน LOB)

Queue Value — ทำไมต้นคิวจึงมีค่าเป็นเงิน

คำสั่ง limit ที่อยู่ต้นคิวได้เปรียบ 2 อย่าง: (1) **ได้ fill ก่อน** เมื่อมี taker เข้ามา และ (2) **โอกาสถูก fill สูงกว่า** เพราะ taker อาจไม่ลึกพอจะถึงคิวหลัง ความน่าจะเป็นที่ "เราจะได้ fill" ขึ้นกับว่ามี volume ฝั่งตรงข้ามมากกว่าปริมาณที่อยู่ *หน้าเรา* หรือไม่

```
Q_ahead = ปริมาณทั้งหมดที่อยู่ "หน้าเรา" ในคิวเดียวกัน
เราได้ fill ก็ต่อเมื่อ: ปริมาณ taker ที่เข้ามากินคิวนี้ > Q_ahead

ยิ่ง Q_ahead น้อย (อยู่ต้นคิว) → P(fill) สูง · ต้องรอ adverse move น้อยลง
```

นี่คือ "ราคาที่จ่ายไม่เห็น" ของการเป็น maker: ถ้ามาช้า Q_ahead เยอะ คุณอาจต้องรอนาน และระหว่างรอราคาอาจวิ่งหนี ทำให้คำสั่งไม่ได้ fill เลย — เรื่องนี้คือหัวใจของ market making ใน Part 9–11

⚠️ กับดักมือใหม่ — "ตั้ง limit แล้วชัวร์"

มือใหม่ชอบคิดว่า "ตั้ง limit ที่ราคานี้ เดี๋ยวก็โดน fill แน่ ๆ" — **ผิด!** limit order ไม่การันตี fill มันการันตีแค่ "ราคา" ที่จะได้ *ถ้า* ได้ fill คุณยังต้องรอจน Q_ahead หน้าคุณถูกกินหมดก่อน ถ้าราคาวิ่งหนีไปก่อน คำสั่งคุณค้างเติ่ง นี่คือ **queue risk** — ความเสี่ยงที่มือใหม่มองข้ามตลอด แล้วงงว่าทำไม "ตั้งราคาดีแล้วไม่ติด"

📑 งานวิจัยรองรับ

งานสาย queue dynamics เช่น **Lu & Abergel (2018)** และ **Queue-Reactive Hawkes (arXiv 1901.08938)** ชี้ว่าพฤติกรรมของคิวไม่ได้สุ่มแบบไร้ความจำ — อัตราการเติม/ถอน/กินคิวขึ้นกับ "สภาพคิวปัจจุบัน" เอง (queue-reactive) และ **Importance of Order Sizes (arXiv 2405.18594)** เน้นว่า "ขนาด" ของแต่ละคำสั่งในคิวมีผลต่อการพยากรณ์ ไม่ใช่แค่ปริมาณรวม — ตอกย้ำว่า Q_ahead เป็นตัวแปรจริงที่วัดได้

🔗 เชื่อมโยงเล่มอื่น

Queue value และ Q_ahead ที่เรียนตรงนี้คือรากฐานของ **market making (Part 9–11)** — รายได้ของ MM คือ "ครึ่ง spread" ที่ได้จากการอยู่ในคิวแล้วถูก fill ส่วน queue risk เชื่อมตรงไปยังการคิด P&L และ inventory ในเล่ม **Payoff Mastery (pm)** เพราะคำสั่งที่ไม่ได้ fill = position ที่ไม่เกิด = แผนเทรดพัง

✍️ แบบฝึก (ทำได้ด้วยข้อมูลคริปโตฟรี)

สมัคร WebSocket stream ของ Binance ช่องทาง **`btcusdt@depth`** (diff depth) แล้วบันทึกปริมาณที่ best bid เป็นเวลา 10 วินาที สังเกตว่าตัวเลขขยับขึ้นลงตลอดจาก limit เข้า/cancel/market กิน ลองเดาว่าถ้าคุณวาง limit เพิ่ม 0.01 BTC ตอนนี้ Q_ahead ของคุณจะเป็นเท่าไร และต้องมี market sell เข้ามากี่ BTC คุณถึงจะได้ fill

---

ต่อ Part 2: 3 เหตุการณ์ที่ขยับกระดาน — Limit / Cancel / Market และ walk the book →

อ่านกระดาน
Reading the Book

Part 2: 3 เหตุการณ์ที่ขยับกระดาน — Limit / Cancel / Market

Order Flow & Microstructure สำหรับรายย่อย · เลนส์ execution ของซีรีส์

เติม-เอาออก-หยิบของ: maker vs taker, walk the book, slippage และเมื่อไหร่ควรใช้คำสั่งแบบไหน

Part 2 — 3 เหตุการณ์ที่ขยับกระดาน

🎯 **อ่านจบ Part นี้คุณจะ**

- รู้ว่ามี **3 เหตุการณ์เดียวเท่านั้น**ที่ทำให้กระดานเปลี่ยน: Limit (เติม) / Cancel (เอาออก) / Market (หยิบ)
- แยก **maker (ให้สภาพคล่อง)** ออกจาก **taker (กินสภาพคล่อง)** และรู้ว่าใครจ่าย spread
- เข้าใจ **walk the book** — ทำไมออเดอร์ก้อนใหญ่จึง "ไต่" กินหลายชั้นจนได้ราคาแย่ลง (slippage)
- เลือกได้ว่า **เมื่อไหร่ควรใช้ limit เมื่อไหร่ควรใช้ market** (มีตารางสรุปครบ)

เริ่มจากชั้นวางของในซูเปอร์

นึกถึงชั้นวางสินค้าในซูเปอร์มาร์เก็ต มีแค่ 3 อย่างที่ทำให้ของบนชั้นเปลี่ยน:

- **พนักงานเติมของ** = **Limit order** — เอาของมาวางเพิ่ม (ให้สภาพคล่อง / provide)
- **พนักงานที่เคยเอาของมาวางบนชั้น เปลี่ยนใจยกกลับเข้าคลัง → ของบนชั้นหายไป** = **Cancel** — ถอน limit ของตัวเองออก (ลดสภาพคล่อง)
- **ลูกค้าหยิบของใส่ตะกร้าจ่ายเงิน** = **Market order** — กินของออกจากชั้นทันที (consume)

ทุกการเคลื่อนไหวบนกระดาน — ทุก ๆ ตัวเลขที่กระพริบ — สรุปได้เป็นหนึ่งใน 3 เหตุการณ์นี้เสมอ

มีแค่ 3 อย่างที่ขยับกระดาน

Limit/Cancel = ขยับ "ของที่รออยู่" · Market = ทำให้เกิดการเทรดจริง

เหตุการณ์ที่ 1–3 ทำงานยังไงบนกระดาน

มาดูผลของแต่ละเหตุการณ์ที่ best ask = 60,003 (มี 5 BTC รออยู่):

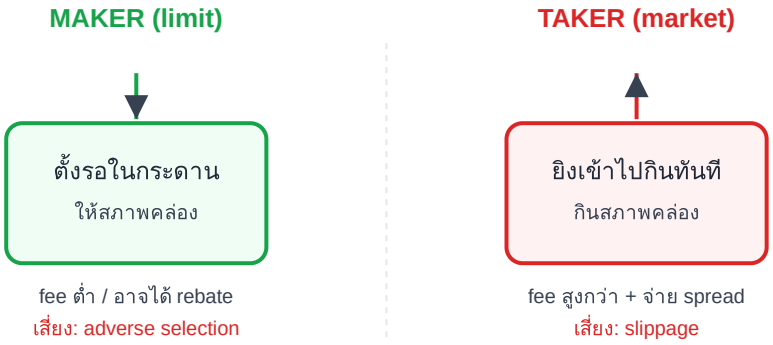
- **Limit (เติม):** มีคนตั้งขายเพิ่มที่ 60,003 อีก 3 BTC → ฝั่ง ask หนาขึ้น (5→8) ราคายังไม่ขยับ
- **Cancel (เอาออก):** เจ้าของ 5 BTC ถอนคำสั่งคืน → ชั้น 60,003 ว่าง best ask เลื่อนไปชั้นถัดไป (เช่น 60,005) spread กว้างขึ้น

- **Market (หยิบ):** มี market buy 5 BTC → กิน 60,003 หมด เกิดการเทรดจริง best ask เลื่อนขึ้น

ภาพ 2.1 · กระดานก่อน/หลัง 3 เหตุการณ์ (ต่อยอดเทมเพลตแม่)

Maker vs Taker — ใครให้ ใครกิน

หัวใจคือ "ใครทำให้เกิดการเทรด": คนที่ **ตั้งรอ**ในกระดาน (limit) คือ **maker** เพราะเป็นคนสร้างสภาพคล่องไว้ให้ ส่วนคนที่ **ยิงเข้าไปกิน**คำสั่งที่รออยู่ (market) คือ **taker** taker เป็นฝ่ายจ่าย spread และมักโดนคิด fee สูงกว่า



ภาพ 2.2 · Maker vs Taker

🔑 **กล่องสรุปหัวใจ — Maker / Taker / Market Order: ใช้อันไหนเมื่อไหร่**

มิติ	Limit (Maker)	Market Order (Taker)
ทำอะไร	วางคำสั่งรอที่ราคาที่เลือก	สั่งซื้อ/ขายทันทีที่ราคาดีสุดที่มี
บทบาทสภาพคล่อง	**ให้ (provide)** เติมของเข้ากระดาน	**กิน (consume)** หยิบของออก
ค่าธรรมเนียม	maker fee ต่ำ / บางที่ได้ **rebate**	taker fee สูงกว่า + จ่าย **spread**
ความเร็ว/ความแน่นอน	อาจไม่ได้ fill (queue risk)	fill ทันที (แต่เสี่ยง slippage)
ความเสี่ยงหลัก	**adverse selection** (โดนคนรู้ข้อมูลกิน)	**slippage / walk the book**
เหมาะเมื่อ	ไม่รีบ, อยากได้ราคาดี, เป็น MM/grid	รีบ, ต้องการความแน่นอน, ปิดด่วน/hedge
"มีข้อมูล" ไหม	สัญญาณอ่อน (ตั้งรอ ยกเลิกฟรี)	**informative** (ยอมจ่ายเพื่อได้ทันที → Part 6)

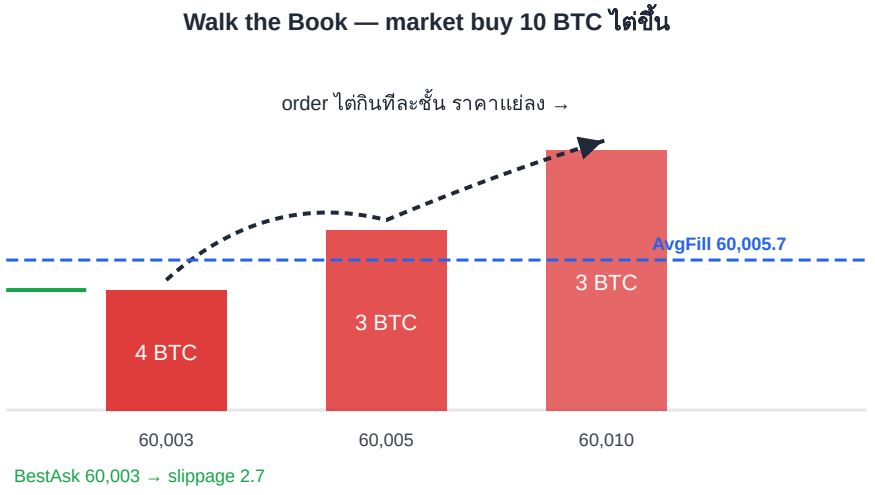
หมายเหตุ: Cancel เป็นเหตุการณ์ที่สาม (ถอน limit ออก = ลดสภาพคล่อง ไม่มีเงินเปลี่ยนมือ) — ไม่ใช่ maker/taker แต่เป็นส่วนของ event flow ที่ขยับกระดาน · **post-only** (Part 11) = limit order ที่ยกเลิกอัตโนมัติถ้าจะกลายเป็น taker → การันตีเป็น maker เสมอ

Walk the Book และ Slippage

เมื่อ market order ใหญ่กว่าปริมาณที่ best level จะรับได้ มันจะ "ไต่" (walk) กินชั้นถัดไป ๆ ที่ราคาแย่ลงเรื่อย ๆ — ราคาเฉลี่ยที่ได้จริงจึงแย่กว่า best ask ส่วนต่างนั้นคือ **slippage**

สมมติคุณ market buy 10 BTC แต่ ask แต่ละชั้นมีไม่พอ:

```
ชั้น ask: 60,003 × 4 BTC | 60,005 × 3 BTC | 60,010 × 3 BTC

AvgFill = Σ(p·q)/Σq
       = (60,003×4 + 60,005×3 + 60,010×3) / 10
       = 600,057 / 10 = 60,005.7 USDT

Slippage = AvgFill − BestAsk = 60,005.7 − 60,003 = 2.7 USDT/BTC
```



ภาพ 2.3 · Walk the book & slippage

⚠️ กับดักมือใหม่ — market order บนเหรียญบาง

บนเหรียญใหญ่อย่าง BTC กระดานลึก market buy เล็ก ๆ แทบไม่มี slippage — มือใหม่จึงชินมือกด market ตลอด แต่พอไปเทรด **เหรียญเล็กที่กระดานบาง** market order ก้อนเดียวกัน "ไต่" กินหลายชั้น **ดันราคาตัวเอง**ขึ้นไป 1–5% ในวินาทีเดียว แล้วตอนขายก็เจอ slippage กลับทางอีกรอบ — กลายเป็นว่าจ่ายค่า spread + slippage หนักกว่ากำไรที่หวัง ก่อนกด market บนเหรียญเล็กให้ดู depth ก่อนเสมอ หรือใช้ limit แทน

📑 งานวิจัยรองรับ

การจำแนกว่าเทรดแต่ละครั้งเป็น "ฝั่งกินซื้อ" หรือ "ฝั่งกินขาย" (aggressor side) มีรากจาก **Lee & Ready (1991)** ซึ่งเป็นมาตรฐานการติดป้าย trade ส่วนความสัมพันธ์ระหว่าง market order (taker flow) กับการเคลื่อนของราคา จะถูกขยายเป็นเชิงปริมาณใน Part 6 (Delta/CVD) และ Part 7 (price impact, Kyle's λ) — market order ไม่ใช่แค่ "วิธีซื้อ" แต่เป็น **สัญญาณ**ว่าใครยอมจ่ายเพื่อความเร็ว

🔗 เชื่อมโยงเล่มอื่น

Slippage และ effective price ที่เรียนตรงนี้คือ "ต้นทุนจริงของการ execute" ที่เล่ม **arb** ต้องหักออกก่อนสรุปว่ามีกำไร arbitrage จริงไหม (สะพานสู่ Part 14) ส่วนการเลือก maker/taker เชื่อมตรงไป **pm** เพราะค่าธรรมเนียมและ spread กินเข้าไปใน payoff ของทุกไม้

✍️ แบบฝึก (ทำได้ด้วยข้อมูลคริปโตฟรี)

ดึง order book L2 ของ BTC/USDT จาก REST `/api/v3/depth` (Binance) มา 20 ชั้น แล้วคำนวณด้วยมือว่า ถ้าคุณ **market buy มูลค่า 5,000 USD** จะกินกี่ชั้น ราคาเฉลี่ย (AvgFill) เท่าไร และ **slippage** กี่ USDT/bps เทียบกับ best ask จากนั้นทำซ้ำกับเหรียญอันดับ 100+ ที่กระดานบาง แล้วดูว่า slippage ต่างกันกี่เท่า

ต่อ Part 3: Bid-Ask Spread แยกส่วน — ช่องว่างสองฝั่งมาจากต้นทุนอะไรบ้าง →

อ่านกระดาน
Reading the Book

Part 3: Bid-Ask Spread แยกส่วน — ช่องว่างสองฝั่งมาจากต้นทุนอะไร

Order Flow & Microstructure สำหรับรายย่อย · เลนส์ execution ของซีรีส์

3 ก้อนต้นทุนของ spread (processing + inventory + adverse selection) และปัจจัยที่ทำให้กว้าง/แคบ

Part 3 — Bid-Ask Spread แยกส่วน

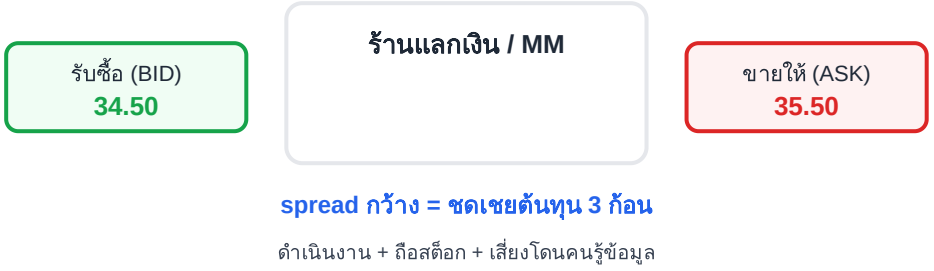
🎯 **อ่านจบ Part นี้คุณจะ**

- เห็นว่า spread ไม่ใช่ "ค่าธรรมเนียมลอย ๆ" แต่ประกอบจาก **3 ก้อนต้นทุน**ที่อธิบายได้
- แยก order-processing cost / inventory cost / **adverse-selection cost** ออกจากกัน
- รู้ปัจจัยที่ทำให้ spread **กว้างหรือแคบ**: tick, volatility, volume, การแข่งขัน MM, เวลา/ข่าว
- คำนวณ **effective spread** เพื่อวัดต้นทุนจริงที่เทรดเดอร์จ่าย ไม่ใช่แค่ spread ที่เห็นบนจอ

เริ่มจากร้านแลกเงินสนามบิน

ร้านแลกเงินในสนามบินตั้ง spread กว้างมาก — รับซื้อ USD ถูก ขายแพง ส่วนต่างเยอะกว่าร้านในเมืองหลายเท่า ทำไม? เพราะร้านมี **ต้นทุน 3 อย่าง**ซ่อนอยู่: ค่าดำเนินงาน (เช่าที่แพง พนักงาน) + ความเสี่ยงถือเงินสกุลที่ค่าผันผวน + ความเสี่ยงที่ "ลูกค้าบางคนรู้ว่าค่าเงินกำลังจะวิ่ง" แล้วมาแลกก่อน

Market maker ในกระดานคริปโตก็คือร้านแลกเงินนี้ — spread ที่เขาตั้งคือผลรวมของต้นทุน 3 ก้อนเดียวกัน ยิ่งสถานการณ์เสี่ยง spread ยิ่งกว้างเพื่อชดเชย



3 ก้อนต้นทุนของ Spread

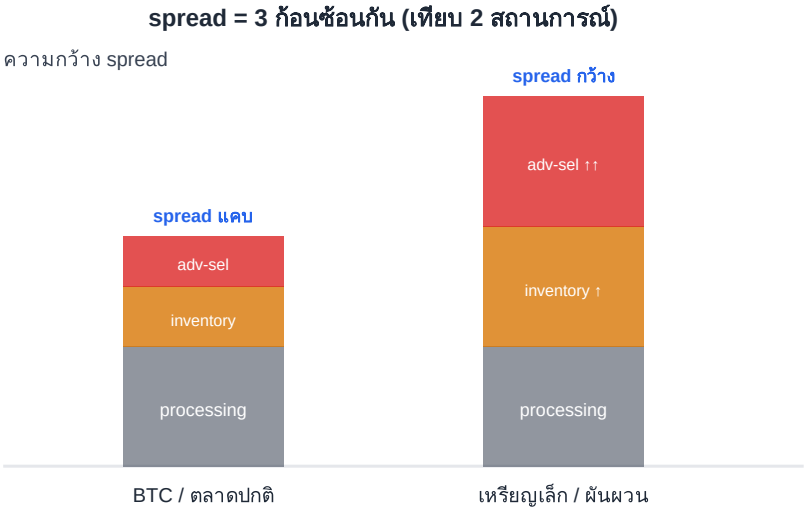
ทฤษฎี microstructure แยก spread ออกเป็น 3 องค์ประกอบ:

1. **Order-processing cost (ค่าดำเนินงาน):** ต้นทุนพื้นฐานของการทำธุรกิจ MM — โครงสร้างพื้นฐาน, ค่าธรรมเนียม, เทคโนโลยี เป็นก้อนคงที่ที่สุด
2. **Inventory cost (ความเสี่ยงสต็อก):** เมื่อ MM ถูก fill สะสม position ค้างไว้ เขาแบกความเสี่ยงราคา ถ้าผันผวนสูง (volatility) เขาต้องกาง spread กว้างขึ้นเพื่อชดเชย
3. **Adverse-selection cost (ความเสี่ยงโดนคนรู้ข้อมูลกิน):** ก้อนสำคัญที่สุดในเชิงทฤษฎี — MM ไม่รู้ว่าคนที่มากินคำสั่งมีข้อมูลลับหรือไม่ ถ้าโดน "informed trader" กินบ่อย ๆ MM ขาดทุนเป็นระบบ จึงต้องกาง

spread เผื่อ (รากจาก Glosten-Milgrom 1985)

```
Spread = Processing + Inventory + AdverseSelection

Effective Spread = 2 × | P_trade − Mid | ← ต้นทุนจริงต่อการเทรด 1 ครั้ง

ตัวอย่าง: market buy ได้ราคา 60,003 ขณะ mid = 60,000
  EffSpread = 2 × |60,003 − 60,000| = 6 USDT
```

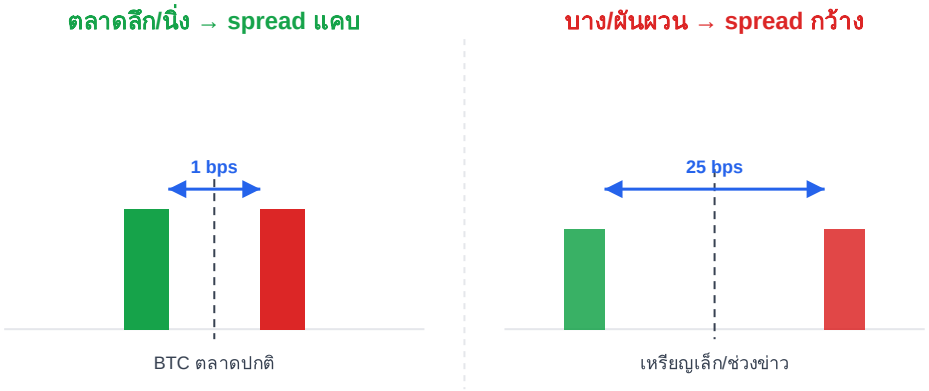
ย้ำให้ชัด: **quoted spread** คือช่องว่าง best ask − best bid ที่ *เห็นบนจอ* (ค่าที่ "ป้ายบอก") ส่วน **effective spread** คือต้นทุน *ที่จ่ายจริง* วัดจากราคา fill เทียบ mid — รวม slippage เมื่อออเดอร์ walk the book ไปกินหลายชั้น จึงมักกว้างกว่า quoted spread โดยเฉพาะออเดอร์ก้อนใหญ่หรือกระดานบาง



ภาพ 3.1 · Stacked bar — 3 ก้อนต้นทุนของ spread

ปัจจัยที่ทำให้ spread กว้างหรือแคบ

ปัจจัย	spread กว้างขึ้นเมื่อ	spread แคบลงเมื่อ
Tick size	tick ใหญ่ (ราคาขยับหยาบ spread ต่ำสุด = 1 tick)	tick เล็ก (ราคาละเอียด แข่งกันได้)
Volatility	ผันผวนสูง (inventory risk พุ่ง)	ตลาดนิ่ง
Volume / สภาพคล่อง	ปริมาณซื้อขายน้อย (เหรียญเล็ก)	ปริมาณมาก (BTC/ETH)
การแข่งขัน MM	มี MM น้อยราย ผูกขาด	MM แข่งกันบีบ spread
เวลา / ข่าว	ใกล้ข่าว, ตลาดเงียบนอกเวลา, ความไม่แน่นอนสูง	ช่วงสภาพคล่องสูง ไม่มีข่าวค้าง



ภาพ 3.2 · Spread กว้าง vs แคบ 2 สถานการณ์ (ต่อยอดเทมเพลตแม่)

Informed vs Noise Trader — ที่มาของ adverse selection

ทำไมก้อน adverse selection ถึงสำคัญที่สุด? เพราะ MM ต้องตั้งราคาทั้งที่ **ไม่รู้ว่าคนที่จะมากินเป็นใคร**:

- **Noise trader:** เทรดด้วยเหตุผลสุ่ม (ปรับพอร์ต, sentiment) — MM กิน spread จากคนกลุ่มนี้ได้กำไร
- **Informed trader:** รู้ว่าราคากำลังจะวิ่ง มากินฝั่งเดียว — MM ที่ขายให้เขาจะ "ขาดทุนทันที" เพราะราคาวิ่งตามหลัง

MM กาง spread เพื่อให้ "กำไรจาก noise" ชดเชย "ขาดทุนจาก informed" ได้พอดี — นี่คือหัวใจของ Glosten-Milgrom

ภาพ 3.3 · Informed vs noise trader

⚠️ กับดักมือใหม่ — เทรดเหรียญเล็กช่วงตลาดเงียบ

มือใหม่เห็นเหรียญเล็กราคาวิ่งแรงตอนดึก (ตลาด US ปิด) แล้วกระโดดเข้า — **ช่วงเงียบ MM ถอนตัว spread พองเป็น 10–50 bps** คุณจ่ายค่า spread มหาศาลแค่ "เข้า-ออก" ก่อนจะคิดเรื่องทิศทางด้วยซ้ำ ยิ่งถ้าเทรดบ่อย ต้นทุน spread สะสมกินกำไรหมด — **spread บนจอคือ "ภาษีเข้า-ออก" ที่ต้องคิดก่อนทุกไม้** ไม่ใช่ดูแค่ราคาเป้าหมาย

📑 งานวิจัยรองรับ

Glosten & Milgrom (1985) พิสูจน์ว่า spread เกิดจาก adverse selection แม้ MM ไม่มีต้นทุนดำเนินงานเลยก็ตาม ส่วน **Stoll** และงานสาย spread decomposition ต่อมาแยกวัด 3 องค์ประกอบเชิงประจักษ์ได้ — ในคริปโต โครงสร้าง fee/rebate ของ Binance/Bybit ก็มีผลโดยตรงต่อ spread ที่ MM ยอมตั้ง

🔎 ต้องเช็คก่อนใช้จริง

โครงสร้าง **fee/rebate** ของ Binance/Bybit เปลี่ยนตามโปรโมชันและ VIP tier เสมอ ตัวเลขในเล่มนี้เป็นตัวอย่างเชิงแนวคิด — ก่อนคิด spread/ต้นทุนจริง ให้เปิดหน้า fee schedule ปัจจุบันของ exchange ที่ใช้

🔗 เชื่อมโยงเล่มอื่น

ก้อน **adverse selection** คือสะพานตรงสู่ Part 6 (market order = informative) และ Part 9–11 (market making — MM ต้องบริหารก้อนนี้ไม่ให้ขาดทุน) ส่วน effective spread คือ "ต้นทุนเข้า-ออกจริง" ที่เล่ม **pm** และ **arb** ต้องหักก่อนสรุปกำไรทุกครั้ง

✍️ แบบฝึก (ทำได้ด้วยข้อมูลคริปโตฟรี)

เก็บ best bid/ask ของ BTC/USDT ทุก ๆ นาทีเป็นเวลา 24 ชั่วโมง (ผ่าน REST `/api/v3/ticker/bookTicker`) แล้ว plot **spread (bps) ราย 24 ชม.** เทียบกับเวลาข่าวเศรษฐกิจสหรัฐ (เช่น CPI, FOMC) คุณจะเห็น spread "พอง" ขึ้นชัดในช่วงก่อน/หลังข่าว และช่วงตลาดเงียบ — ลองทำซ้ำกับเหรียญเล็กเทียบกัน

ต่อ Part 4: กระดานเชื่อถือได้แค่ไหน/นานแค่ไหน — คลื่นจริง vs ฟองลวง →

อ่านกระดาน
Reading the Book

Part 4: กระดานเชื่อถือได้แค่ไหน / นานแค่ไหน

Order Flow & Microstructure สำหรับรายย่อย · เลนส์ execution ของซีรีส์

คลื่นจริง vs ฟองลวง: signal horizon, depth-at-touch vs deep, spoofing/fleeting liquidity, resiliency

Part 4 — กระดานเชื่อถือได้แค่ไหน / นานแค่ไหน

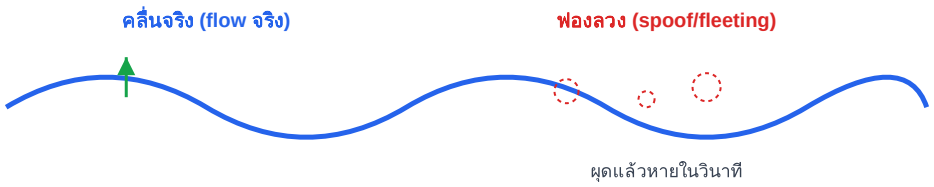
🎯 **อ่านจบ Part นี้คุณจะ**

- รู้ว่าสัญญาณจากกระดานมี **อายุสั้น (วินาที)** — signal horizon ไม่ใช่ทิศทางรายชั่วโมง
- แยก **depth-at-touch (สภาพคล่องที่ best)** ออกจาก deep liquidity (ลึกลงไป) และรู้ว่าอันไหนเชื่อได้กว่า
- จับสัญญาณ **spoofing / fleeting liquidity** — ออเดอร์ลวงที่ตั้งไว้หลอกแล้วถอน
- เข้าใจ **resiliency** — กระดานเติมตัวเองกลับเร็วแค่ไหนหลังถูกออเดอร์ใหญ่กิน

เริ่มจากผิวน้ำทะเล

มองผิวน้ำทะเลจากชายฝั่ง คุณเห็นทั้ง **คลื่นจริง** (พลังงานเคลื่อนเข้าหาฝั่งจริง ๆ) ปนกับ **ฟองและระลอกลวง** (ผุดขึ้นแล้วหายในวินาที) — กระดานก็เหมือนกัน บางก้อนคือความตั้งใจซื้อขายจริง บางก้อนเป็นแค่ "ฟอง" ที่ตั้งไว้หลอกแล้วถอนทันทีที่ราคาใกล้แตะ

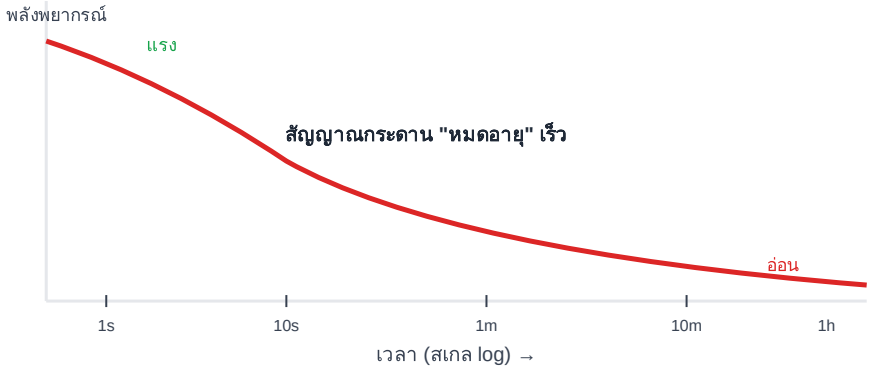
คนอ่านกระดานเก่งไม่ได้เชื่อทุกอย่างที่เห็น แต่รู้ว่า **อะไรคือคลื่นจริง อะไรคือฟอง และฟองนั้นอยู่ได้นานแค่ไหน**



อ่านกระดาน = แยกคลื่นจริงออกจากฟอง

มิติ 1: Signal Horizon — สัญญาณอายุสั้น

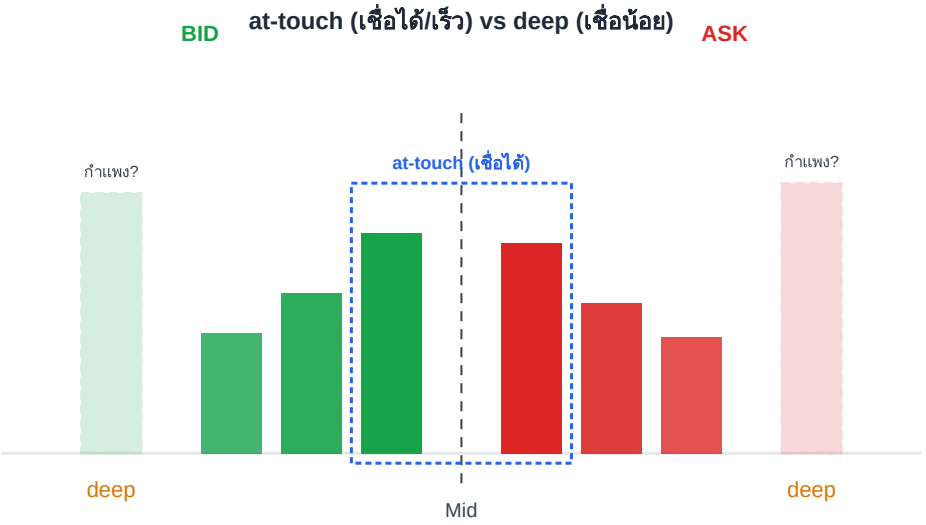
ข้อมูลจากกระดาน (เช่น imbalance) ทำนายราคาได้ดี แต่ **ในกรอบเวลาสั้นมาก** — ระดับวินาทีถึงหลักนาที ไม่ใช่ชั่วโมงหรือวัน ยิ่งทอดเวลาออกไป พลังพยากรณ์ยิ่งสลายตัวเร็ว (decay) เพราะกระดานถูกเขียนทับด้วยเหตุการณ์ใหม่ตลอด



ภาพ 4.1 · Signal horizon (แกนเวลา log) — พลังพยากรณ์สลายตามเวลา

มิติ 2: Depth-at-touch vs Deep

สภาพคล่องที่ **best level (at-touch)** คือของที่ "พร้อมโดนเทรดทันที" — เปลี่ยนแปลงเร็ว สะท้อนแรงจริงระยะสั้น ส่วนสภาพคล่อง **ลึกลงไป (deep)** มักเป็นออเดอร์ที่ตั้งห่าง ๆ ไว้ ซึ่ง **เชื่อถือได้น้อยกว่า** เพราะเจ้าของมีเวลาถอนก่อนราคาแตะ — "กำแพง" ใหญ่ที่อยู่ลึกจึงมักเป็นเป้าของ spoofing



ภาพ 4.2a · Depth-at-touch vs deep (ต่อยอดเทมเพลตแม่ LOB)

มิติ 3: Spoofing / Fleeting Liquidity

Spoofing คือการตั้งออเดอร์ก้อนใหญ่ไว้ (มักลึกห่าง best) เพื่อ "หลอก" ให้คนอื่นคิดว่ามีแรงรับ/แรงต้านจริง แล้ว **ถอน (cancel) ทันทีที่ราคาใกล้จะแตะ** — เป็นการสร้างภาพลวงโดยไม่ตั้งใจให้ถูก fill จริง ส่วน **fleeting liquidity** คือออเดอร์อายุสั้นมากที่ผุดและหายตลอดเวลา

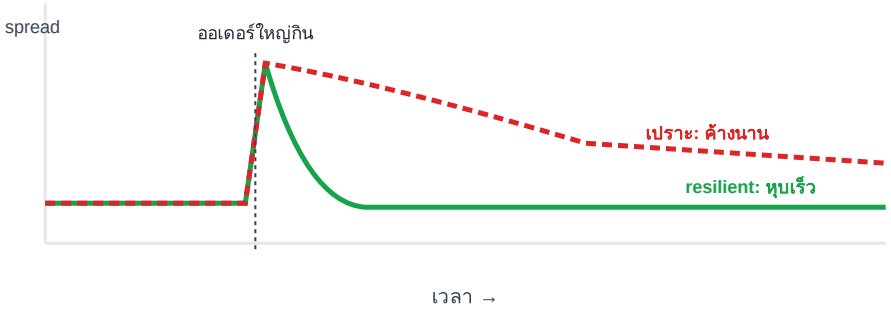
🚩 **ธงเตือน spoof — สังเกต 3 อย่างพร้อมกัน**

- **ก้อนใหญ่ผิดปกติ** เมื่อเทียบกับ depth ปกติของเหรียญนั้น
- **ตั้งห่าง best** (ไม่ยอมเข้าใกล้จุดที่จะโดน fill จริง)
- **cancel ทันทีเมื่อราคาเข้าใกล้** — ก้อนหายก่อนถูกแตะเสมอ

ภาพ 4.2b · Spoof 3 เฟรม (ต่อยอดเทมเพลตแม่ LOB)

มิติ 4: Resiliency — กระดานเติมตัวกลับเร็วแค่ไหน

Resiliency คือความสามารถของกระดานในการ "เติมสภาพคล่องกลับ" หลังถูกออเดอร์ใหญ่กินไป กระดานที่ resilient สูง: ถูกกินแล้ว MM เติม limit กลับมาภายในเสี้ยววินาที spread หุบกลับเร็ว = เชื่อถือได้ ส่วนกระดานที่ resilient ต่ำ: ถูกกินแล้วช่องว่างค้างนาน = เปราะ เสี่ยงราคากระชาก



ภาพ 4.3 · Resiliency timeline

```
OBI = (BidVol − AskVol) / (BidVol + AskVol) ← สเกล −1..+1, กลาง = 0 (พรีวิว — รายละเอียดเต็มใน Part 5)
```

```
ธงเตือน spoof: ก้อนใหญ่ + ตั้งห่าง best + cancel เมื่อราคาเข้าใกล้
→ OBI ที่คำนวณจากก้อน spoof เป็น "สัญญาณลวง" ใช้ทำนายไม่ได้
```

⚠️ กับดักมือใหม่ — เชื่อ "กำแพง" เป็นแนวรับจริง

มือใหม่เห็นก้อน bid ใหญ่ ๆ ที่ราคาหนึ่งแล้วคิดว่า "นี่แนวรับแข็ง ราคาลงไม่หลุดแน่" แล้วเข้าซื้อตาม — **แต่ถ้านั่นคือ spoof พอราคาวิ่งลงมาใกล้ กำแพงหายวับ** ราคาทะลุลงต่อ คุณติดดอย นักเทรดที่ตั้ง spoof ตั้งใจให้คุณคิดแบบนี้พอดี — **อย่าเชื่อก้อนใหญ่ที่อยู่ลึกและไม่เคยถูกแตะ ดูที่ depth-at-touch และพฤติกรรม cancel เป็นหลัก**

📑 งานวิจัยรองรับ

งานสาย microstructure ยืนยันว่า order-book imbalance ทำนายทิศ mid ระยะสั้นได้ (เช่น **Gould & Bonart 2015** สำหรับ queue imbalance) แต่ **พลังพยากรณ์เสื่อมตามเวลาเร็ว** และ **cancel/spoof สร้างสัญญาณลวงได้** — ตอกย้ำว่ากระดานเป็น indicator "อายุสั้น" ที่ต้องใช้คู่ risk management ไม่ใช่ระบบเบ็ดเสร็จ (รายละเอียด OBI เต็มใน Part 5)

🔗 เชื่อมโยงเล่มอื่น

มิติทั้ง 4 ที่นี่เป็น "เครื่องกรองความน่าเชื่อถือ" ก่อนเข้าสู่การอ่าน flow จริงใน Part 5 (OBI — ภาพนิ่งที่ spoof หลอกได้), Part 6 (market order = informative) และ Part 7 (OFI) ส่วนแนวคิด "สัญญาณอายุสั้น + ต้องคู่ risk management" เชื่อมตรงไปยัง mindset ในเล่ม **eye**

✍️ แบบฝึก (ทำได้ด้วยข้อมูลคริปโตฟรี)

เปิด WebSocket `btcusdt@depth` หรือเหรียญสภาพคล่องปานกลาง บันทึก order book ต่อเนื่อง แล้วลอง **จับออเดอร์ก้อนใหญ่ที่หายไปก่อนที่ราคาจะแตะ** — จดเวลาที่มันปรากฏ ระยะห่างจาก best และเวลาที่มันถูก cancel ถ้าก้อนนั้นหายทุกครั้งที่ราคาเข้าใกล้ คุณเพิ่งจับ spoof ได้ด้วยตาตัวเอง ลองนับดูว่าใน 1 ชั่วโมงเจอกี่ครั้ง

จบองก์ I · ต่อ Part 5: Order Book Imbalance (OBI) — ฝั่งไหนหนัก (ภาพนิ่ง) →

อ่านกระดาน
Reading the Book

ภาคทบทวน: สถิติพื้นฐานสำหรับ Order Flow

Order Flow & Microstructure สำหรับรายย่อย · ภาคเครื่องมือ

mean · median · mode · variance · SD · z-score · t-score · normal & t-distribution

แนะนำอ่านก่อน Part 5–11 ถ้าสถิติยังไม่แน่น

ภาคทบทวน — สถิติพื้นฐานที่ Order Flow ต้องใช้

🎯 **อ่านจบภาคนี้คุณจะ**

- สรุป "กองตัวเลข" จากกระดาน (ขนาดออเดอร์ / spread / OFI) ด้วย **ค่ากลาง** (mean, median, mode) ได้ถูกตัว
- วัด "ความผันผวน/ความกระจาย" ด้วย **variance & SD (σ)** — ตัวเดียวกับ σ ใน Avellaneda-Stoikov
- เทียบว่าค่าตอนนี้ "ผิดปกติแค่ไหน" ด้วย **z-score** — รากฐานของการ normalize OFI
- เข้าใจ **normal vs t-distribution** และรู้ว่าทำไม "คริปโตหางหนา" จึงสำคัญต่อการประเมินความเสี่ยง

ทำไม order flow trader ต้องรู้สถิติ

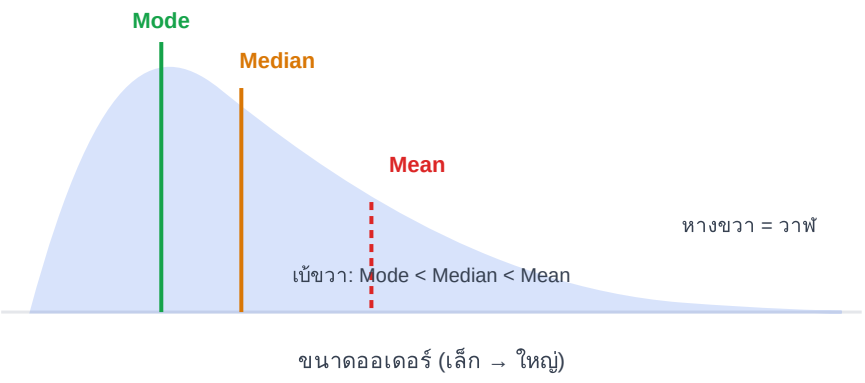
กระดานคือ "กระแสตัวเลข" มหาศาล — trade หลายพันไม้ต่อนาที, ขนาดออเดอร์, spread, OFI ที่แกว่งตลอด สถิติคือเครื่องมือ **3 อย่าง**: (1) **สรุป** กองตัวเลขเป็นค่าเดียว, (2) **วัดความกระจาย** ว่าเหวี่ยงแค่ไหน, (3) **เทียบ** ว่าค่าที่เห็นตอนนี้ "ปกติหรือผิดปกติ" ทุกอย่างในเล่มนี้ (OFI, VPIN, Kyle's λ, A-S) ตั้งอยู่บน 4 กลุ่มนี้

1. ค่ากลาง — Mean / Median / Mode

```
Mean (μ หรือ x̄) = (Σ xᵢ) / n ← ผลรวมหารจำนวน "ค่าเฉลี่ย"
  (Σ = "บวกค่าทุกตัวเข้าด้วยกัน", n = จำนวนข้อมูล)
Median = ค่าตรงกลางเมื่อเรียงน้อย→มาก (ทนต่อค่าสุดโต่ง)
Mode = ค่าที่พบ "บ่อยที่สุด"
```

ตัวอย่างขนาด market order 5 ไม้ (BTC): `0.10, 0.15, 0.10, 0.20, 5.00` (ไม้สุดท้ายคือวาฬ)

```
Mean = (0.10+0.15+0.10+0.20+5.00)/5 = 5.55/5 = 1.11 BTC ← ถูกวาฬดึงสูงเกินจริง
Median = เรียง [0.10, 0.10, 0.15, 0.20, 5.00] → ตัวกลาง = 0.15 BTC ← สะท้อน
"ไม้ทั่วไป" ดีกว่า
Mode = 0.10 (พบ 2 ครั้ง) ← ขนาดยอดนิยม (มักเป็นเลขกลม)
```



ภาพ S.1 · ข้อมูลเบ้ขวา (right-skewed) — วาฬไม่กี่ไม้ดึง Mean ออกไปขวา

> 📌 **ใช้ใน order flow ยังไง**
>
> ขนาดออเดอร์/ปริมาณในคริปโต **เบ้ขวาเสมอ** (วาฬทำให้มีหางขวายาว) → ใช้ **median** อธิบาย "ไม้ทั่วไป" ดีกว่า mean ส่วน **mode** ช่วยจับ "เลขกลมยอดนิยม" (0.1, 0.5, 1 BTC) ที่มักเป็นจุดวางออเดอร์ของคน — ส่วน **mean** เหมาะกับค่าที่สมมาตร เช่น mid-price ระยะสั้น (microprice = mean ถ่วงน้ำหนัก!)

2. ความกระจาย — Variance & Standard Deviation (σ)

ค่ากลางบอก "อยู่ตรงไหน" แต่ไม่บอก "เหวี่ยงแค่ไหน" — นั่นคือหน้าที่ของ variance (σ^2) และ SD (σ)

```
Variance (population) σ² = Σ(xᵢ − μ)² / N
Variance (sample) s² = Σ(xᵢ − x̄)² / (n − 1) ← ใช้ n−1 เมื่อเป็น "ตัวอย่าง"
Standard Deviation σ = √(variance) ← หน่วยเดียวกับข้อมูล
```

ประชากร (population) = มีข้อมูลครบทุกตัว → หารด้วย N · **ตัวอย่าง (sample)** = สุ่มมาแค่บางส่วน → หารด้วย **n−1** (ชดเชยไม่ให้ประเมินความกระจายต่ำเกินไป) ในทางปฏิบัติเรามักมีแค่ "ตัวอย่าง" จึงใช้ n−1

SD ใหญ่ = ข้อมูลเหวี่ยงมาก (ผันผวนสูง); SD เล็ก = เกาะกลุ่มใกล้ค่าเฉลี่ย

> 🔗 **σ ตัวนี้คือตัวเดียวกับในเล่ม**
>
> **σ** (volatility) คือหัวใจของ **Avellaneda-Stoikov** (Part 10): reservation price `r = s − q·γ·σ²·(T−t)` โดย **q** = inventory ที่ถืออยู่, **γ** = ระดับความกลัวความเสี่ยง — ยิ่งราคา SD สูง (ผันผวน) MM ยิ่งกาง spread กว้าง · σ ยังโผล่เป็นตัวหารใน z-score (หัวข้อถัดไป)

3. Z-score — "ค่านี้ผิดปกติกี่ SD"

z-score แปลงค่าดิบให้เป็น "ห่างจากค่าเฉลี่ยกี่เท่าของ SD" — ทำให้เทียบข้ามสเกล/ข้ามเหรียญได้

```
z = (x − μ) / σ  z=0 คือค่าเฉลี่ยพอดี · z=+2 คือสูงกว่าเฉลี่ย 2 SD (หายาก)
```

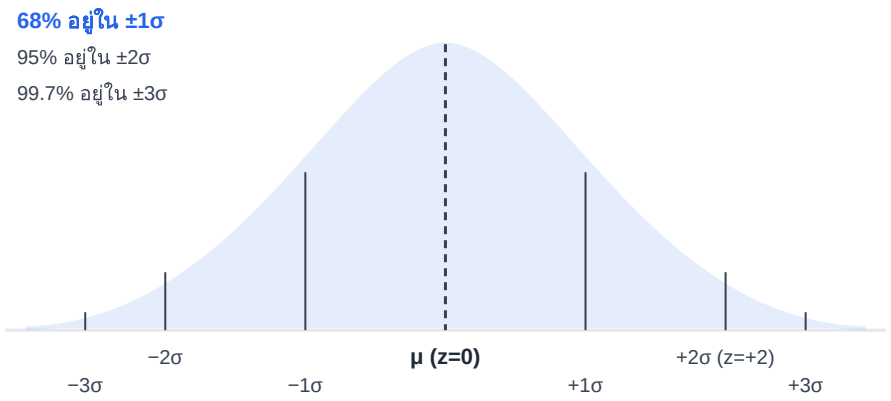
ตัวอย่าง: spread ปกติของ BTC เฉลี่ย μ=6, σ=1.5 ตอนนี้ spread พุ่งเป็น 12

```
z = (12 − 6) / 1.5 = +4.0 → สูงกว่าค่าเฉลี่ยจน percentile > 99.99% (one-tailed) = ผิดปกติมาก = สัญญาณข่าว/สภาพคล่องหด
```

> ✅ **z-score คือหัวใจของการ "normalize"**
>
> ใน Part 7 เรา **stationarize OFI** ก่อน regression — วิธีหนึ่งคือทำ z-score (หักค่าเฉลี่ยกลิ้ง หารด้วย SD กลิ้ง) เพื่อให้ OFI เทียบข้ามช่วงเวลา/เหรียญได้ · ใช้จับ "OBI/spread/CVD ผิดปกติ" ด้วย (เช่น z > 2 = เหตุการณ์หายาก ควรสนใจ)

4. Normal Distribution — เส้นโค้งระฆัง & กฎ 68-95-99.7

การแจกแจงปกติคือเส้นโค้งระฆังสมมาตรรอบ μ ภายใต้ normal: ~68% ของข้อมูลอยู่ใน ±1σ, ~95% ใน ±2σ, ~99.7% ใน ±3σ

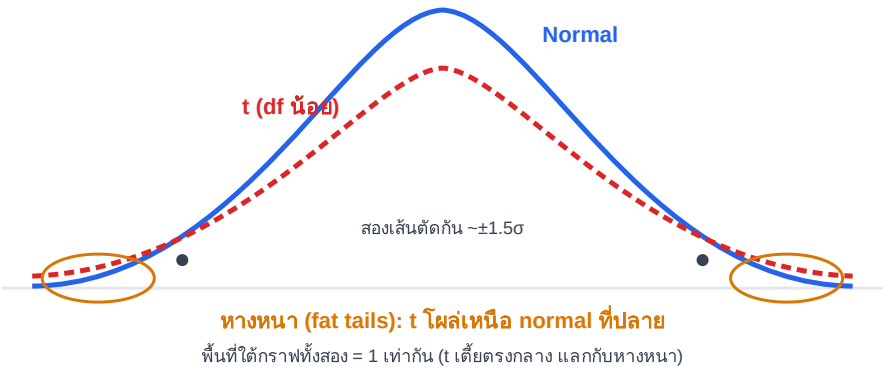


ภาพ S.2 · เส้นโค้งปกติ + กฎ 68-95-99.7 — z=+2 ตรงกับขอบ ±2σ (ครอบ 95%)

> 📌 **ใช้ใน order flow ยังไง**
>
> VPIN (Part 8) ใช้ **normal CDF (Φ)** ใน bulk volume classification: `V_buy = V·Φ(ΔP/σ)` — **Φ (CDF) = พื้นที่ใต้โค้งสะสม = ความน่าจะเป็นที่จะได้ค่าน้อยกว่าค่านี้** จึงแปลงการเปลี่ยนราคาเป็น "สัดส่วนที่น่าจะเป็นฝั่งซื้อ" · **หมายเหตุ:** σ ในสูตรนี้คือ SD ของ "การเปลี่ยนราคาต่อ bucket" — คนละตัวกับ σ volatility ของ A-S

5. T-distribution & T-score — โลกจริง "หางหนา"

t-distribution หน้าตาเหมือนระฆัง แต่ **หางหนากว่า normal** (peak เตี้ยกว่า แลกกับปลายสองข้างที่หนากว่า — พื้นที่ใต้กราฟยังรวมได้ 1 เท่ากัน) ใช้เมื่อ **ไม่รู้ σ จริง** (ต้องประมาณจากข้อมูล) หรือ **กลุ่มตัวอย่างเล็ก** ยิ่ง degrees of freedom (df = n−1) มาก t ยิ่งเข้าใกล้ normal



ภาพ S.3 · t-distribution หางหนากว่า normal — ตรงกับพฤติกรรมราคาคริปโต

T-score ใช้ทดสอบว่า "ค่าเฉลี่ยตัวอย่างต่างจากค่าอ้างอิงอย่างมีนัยหรือเป็นแค่บังเอิญ":

```
t = (x̄ − μ₀) / (s / √n)  df = n − 1
|t| มาก (เช่น > ~2) → ต่างอย่างมีนัยสำคัญ (ไม่น่าใช่บังเอิญ)
```

🔗 **ใช้ใน order flow ยังไง**

ตอน regress **ΔP = λ·OFI** (Kyle's λ, Part 7) ค่า **t-stat ของ λ** บอกว่าความสัมพันธ์ "มีจริง" หรือบังเอิญ (มักดู |t| > 2) — ในงาน order flow ข้อมูล n มักใหญ่มาก (หลายพันจุด) ที่ยังใช้ t เพราะ **"ไม่รู้ σ จริง ต้องประมาณจาก residual"** ไม่ใช่เพราะ n เล็ก · และ **fat tails** คือเหตุผลว่าทำไมต้องระวัง: คริปโตมีวันที่ราคากระชาก ±10% ซึ่ง normal บอกว่า "แทบเป็นไปไม่ได้" แต่เกิดจริงบ่อย — เชื่อมโดยตรงกับ VPIN/toxicity (Part 8) และ inventory risk ของ MM (Part 9–11)

⚠️ **กับดักมือใหม่**

- **สมมติทุกอย่างเป็น normal:** ราคาคริปโตหางหนา การใช้ normal จะ **ประเมิน tail risk ต่ำเกินจริง** — เหตุการณ์ "3σ" เกิดบ่อยกว่าที่คิดมาก
- **ใช้ mean กับข้อมูลเบ้:** วาฬไม่กี่ไม้ทำให้ mean ขนาดออเดอร์เพี้ยน → ใช้ median
- **z-score / t สูง ≠ กำไร:** มันบอกว่า "ผิดปกติ/มีนัยทางสถิติ" ไม่ได้บอกทิศหรือว่าหักต้นทุนแล้วคุ้ม

สรุป — สถิติแต่ละตัวใช้ใน Part ไหน

สถิติ	บอกอะไร	ใช้ใน Part
Mean (ถ่วงน้ำหนัก)	ค่ากลาง / microprice	0, 11
Median / Mode	ขนาดออเดอร์ "ทั่วไป" / เลขยอดนิยม	2, 6
Variance / SD (σ)	ความผันผวน	3 (spread), 10 (A-S)
Z-score	normalize / จับค่าผิดปกติ	7 (OFI), 4 (spread/OBI)
Normal + CDF (Φ)	ความน่าจะเป็น / bulk classification	8 (VPIN)
t-score / t-dist	นัยสำคัญ regression / fat tails	7 (λ), 8, 9–11 (tail risk)

🔗 **เชื่อมโยงเล่มอื่น**

ภาคนี้คือเวอร์ชันย่อที่ปรับให้เข้ากับ order flow — คำว่า "regression" ที่โผล่ในหัวข้อ t ($\Delta P = \lambda \cdot OFI$) คือ **linear regression / OLS** ซึ่งมีพื้นฐานเต็ม (พร้อมการพิสูจน์ + ตัวอย่าง) ในซีรีส์ **math (คณิตศาสตร์สำหรับ Options)** ที่ z-score/normal ใช้ร่วมกันได้ทันที

✍️ **แบบฝึก (ทำได้ด้วยข้อมูลคริปโตฟรี)**

1. ดึง trade 200 ไม้ล่าสุดของ BTC/USDT (aggTrade) เก็บ "ขนาด" — คำนวณ mean, median, mode เทียบกัน → ดูว่าวาฬดึง mean ห่างจาก median แค่ไหน (เบ้ขวา)
2. เก็บ spread ทุก 1 วินาที 5 นาที — คำนวณ μ, σ แล้วหา z-score ของ spread ล่าสุด ถ้า $z > 2$ แปลว่าผิดปกติ
3. เก็บผลตอบแทนราย 1 นาที 1 วัน — พล็อต histogram เทียบเส้น normal ดูว่า "หางหนา" จริงไหม (มีไม้สุดโต่งเกินที่ normal คาด)

มีเครื่องมือสถิติครบแล้ว → กลับไปลุย Part 5 (OBI) ถึง Part 11 (Market Making) ได้เต็มที่ →

อ่านกระดาน
Reading the Book

Part 5: Order Book Imbalance (OBI) — ฝั่งไหนหนัก (ภาพนิ่ง)

Order Flow & Microstructure สำหรับรายย่อย · เลนส์ execution ของซีรีส์

เริ่มต้นองก์ "อ่าน Flow": static(OBI) → flow(OFI) → toxicity(VPIN)

Part 5 — Order Book Imbalance

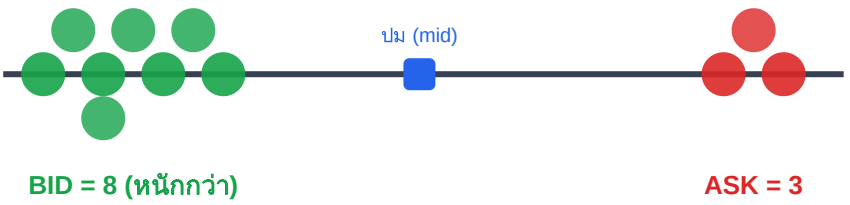
🎯 **อ่านจบ Part นี้คุณจะ**

- คำนวณ **OBI (Order Book Imbalance)** เป็น และเข้าใจว่ามันบอก "ฝั่งไหนหนัก" ในกระดาน ณ วินาทีนี้
- แยกสเกล 2 แบบ: **−1..+1** (กลาง=0) กับ **0..1** (กลาง=0.5) และแปลงไปมาได้
- เข้าใจคำเตือนหัวใจ: **OBI คือภาพนิ่ง (state) ไม่ใช่ flow** — บอกของที่ "ตั้งรออยู่" ไม่ใช่ของที่ "กำลังลงมือ"
- รู้ว่ามันทำนายทิศ mid ถัดไปได้ (R^2~65% ในบางตลาด) แต่ horizon สั้นมากและถูก spoof หลอกได้

นับหัวคนชักเย่อ — ยืนอยู่ ≠ ลงมือ

นึกถึงเกม **ชักเย่อ**: เชือกเส้นเดียว สองทีมยืนคนละข้าง คุณนับหัวคนได้ว่าฝั่งซ้าย 8 คน ฝั่งขวา 3 คน — ฝั่งซ้าย "หนัก" กว่า เชือก *น่าจะ* ถูกดึงไปทางซ้าย

OBI ก็คือการนับหัวคนชักเย่อบนกระดาน — ฝั่ง bid มีปริมาณรออยู่เท่าไร เทียบกับฝั่ง ask แต่ระวัง! **"ยืนถือเชือก" ไม่เท่ากับ "ออกแรงดึงจริง"** คนยืนเฉย ๆ อาจปล่อยมือ (cancel) เมื่อไรก็ได้ นี่คือเหตุผลว่าทำไม OBI เป็นแค่ภาพนิ่ง



OBI = (8−3)/(8+3) = +0.45 → เอียงฝั่งซื้อ

นิยาม OBI — สองสเกลที่ต้องไม่สับสน

OBI วัด **ความไม่สมดุลของปริมาณ** ระหว่างฝั่งซื้อกับฝั่งขาย ในศัพท์เทรดเดอร์เรียกว่า **"Volume Imbalance"** ส่วนในงานวิชาการเรียก **queue imbalance** — เป็นตัวเดียวกัน

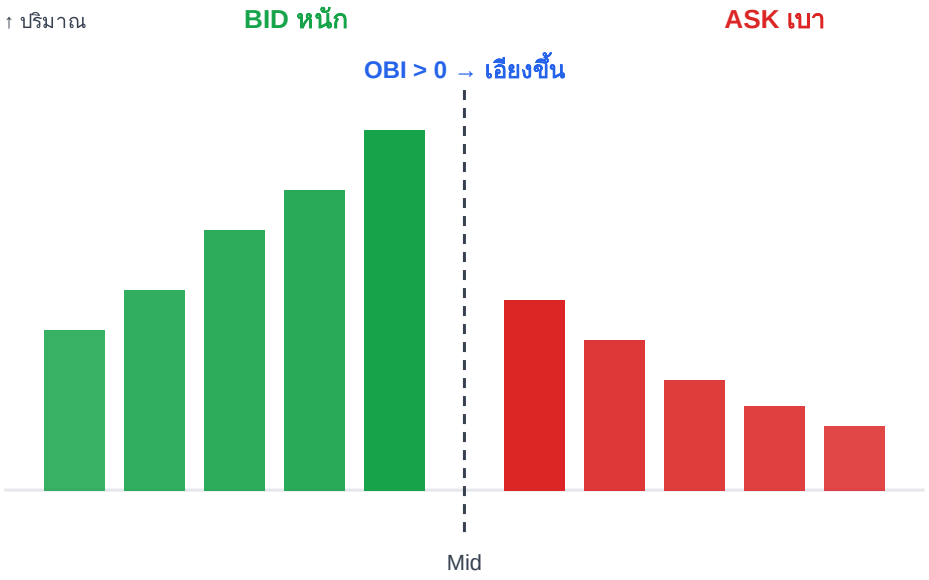
```
สเกล −1..+1 (กลาง = 0) ← ใช้ตัวนี้เป็นมาตรฐานทั้งเล่ม
  OBI = (V_bid − V_ask) / (V_bid + V_ask)
  ตัวอย่าง: V_bid=8, V_ask=3 → OBI = (8−3)/(8+3) = +0.45

สเกล 0..1 (กลาง = 0.5) ← บาง paper/เครื่องมือใช้แบบนี้
  Imbalance = V_bid / (V_bid + V_ask) = 8/11 = 0.73

แปลงกันได้: OBI = 2 × Imbalance − 1 (0.73 → 2×0.73−1 = +0.45)
```

ค่า OBI (−1..+1)	ความหมาย	ทิศที่ "น่าจะ" ไป
+1.0	มีแต่ฝั่ง bid (ask ว่าง)	กดดันขึ้นแรง
+0.4 ถึง +0.9	ฝั่งซื้อหนักกว่า	เอียงขึ้น
~0	สมดุล	ไม่ชัด
−0.4 ถึง −0.9	ฝั่งขายหนักกว่า	เอียงลง
−1.0	มีแต่ฝั่ง ask (bid ว่าง)	กดดันลงแรง

OBI หลายชั้น (multi-level)

OBI ที่ใช้แค่ **best bid/ask** (ชั้นบนสุด) ไวที่สุดแต่ก็ผันผวนที่สุด เราถ่วงน้ำหนักหลายชั้นเพื่อให้ภาพนิ่งขึ้น:

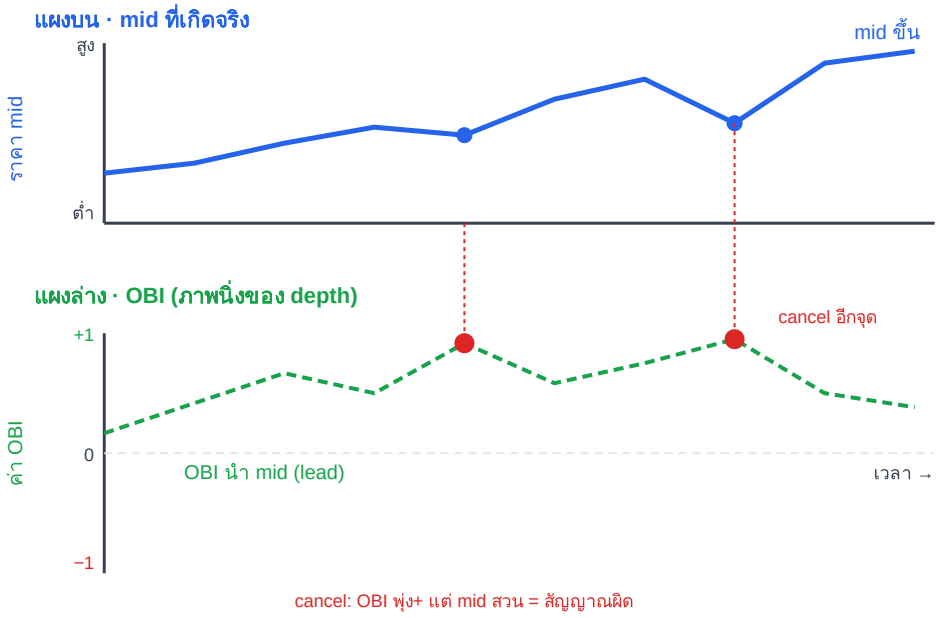
```
OBI_N = ( Σ w_k·V_bid,k − Σ w_k·V_ask,k ) / ( Σ w_k·V_bid,k + Σ
w_k·V_ask,k )
  k = 1..N (ชั้นที่ k จาก best เข้าไป), w_k = น้ำหนัก (มักให้ชั้นลึกน้อยลง)
  เช่น w_k = 1/k หรือ exp(−k) → ชั้น best มีน้ำหนักมากสุด
```



ภาพ 5.1 · มวลฝั่ง bid (เขียว) มากกว่า ask (แดง) → OBI เป็นบวก

OBI = ภาพนิ่ง (state) ไม่ใช่ flow

นี่คือใจความที่ต้องฝังไว้ก่อนข้าม Part 7: OBI ถ่าย "**snapshot**" ของกระดาน ณ วินาทีนี้ มันบอกว่า *ใครยืนถือเชือกอยู่บ้าง* ไม่ได้บอกว่า *เชือกกำลังขยับไปทางไหน* เมื่อมีคน cancel หรือยิง market order เข้ามา ภาพนิ่งก็เปลี่ยนทันที — แต่ OBI ตัวเดิมไม่ได้จับ "การเปลี่ยน" นั้น มันแค่ถ่ายรูปใหม่



ภาพ 5.2 · แยกสองแผง — OBI (ล่าง) มัก**นำ** mid (บน) แต่ที่จุด cancel (จุดแดง) OBI พุ่งบวกสวนทางกับ mid ที่เกิดจริง = สัญญาณลวง

⚠️ กับดักมือใหม่ — "OBI บวก = ขึ้นแน่นาทีหน้า"

OBI ทำนายได้แค่ระดับ **วินาที** ไม่ใช่นาทีหรือชั่วโมง! horizon ของมันสั้นมาก เพราะกระดานเปลี่ยนตลอด ยิ่งกว่านั้น **spoofer** วางกำแพง bid ปลอม ๆ ให้ OBI ดูบวก หลอกให้คุณซื้อ แล้วยกเลิกก่อนราคาแตะ (ดู Part 4) — **เห็น OBI บวกแล้วเปิด long ค้างข้ามคืนคือเข้าใจผิดเรื่อง horizon**

📑 งานวิจัยรองรับ — และข้อควรระวังเรื่องตัวเลข

Gould & Bonart (2015, arXiv:1512.03492) แสดงว่า queue imbalance (= OBI) ทำนายทิศ mid-price **หนึ่งก้าวถัดไป (one-tick-ahead)** ได้ดี โดยเฉพาะหุ้น *large-tick* (tick ใหญ่เทียบราคา คิวยาว) ตัวเลขที่มักอ้างกันว่า OBI อธิบายการขยับ mid ได้ **R^2~65%** นั้น **ขึ้นกับตลาด/timeframe/วิธี calibration อย่างมาก** — large-tick crypto อาจสูง แต่ small-tick หรือ timeframe ยาวจะต่ำลงมาก อย่ายึดเลขนี้เป็นค่าคงที่

🔗 เชื่อมโยงเล่มอื่น

OBI เป็น "ภาพนิ่ง" — **Part 7 (OFI)** จะยกระดับเป็น "วิดีโอการเปลี่ยน" และเปลี่ยนจากการ *เดาความสัมพันธ์* เป็นการ **วัดด้วยสมการถดถอย (regression)** ที่ความชัน (slope) = Kyle's λ ตรงนี้คือจุดที่เล่ม **math** (regression / PCA) เข้ามาเชื่อมเต็มตัว ส่วน **Part 6** จะอธิบายว่าทำไม market order (ของที่ "ลงมือจริง") ถึงมีข้อมูลมากกว่า depth ที่ "แค่ยืนรอ"

✍️ แบบฝึก (ทำได้ด้วยข้อมูลคริปโตฟรี)

ดึง order book ของ BTC/USDT จาก Binance REST `GET /api/v3/depth?symbol=BTCUSDT&limit=5` แล้วคำนวณ **OBI สองแบบ**: (1) ใช้แค่ best bid/ask, (2) ใช้ 5 ชั้นรวม (sum ปริมาณทั้ง 5) เทียบค่าทั้งสอง — คุณจะเห็นว่า best-level เด้งแรงกว่า ส่วน 5-level นิ่งกว่า ทำซ้ำทุก 1 วินาที 30 ครั้ง แล้วสังเกตว่า OBI เด้งไปมาเร็วแค่ไหน (= horizon สั้น)

ต่อ Part 6: ทำไม Market Order สำคัญ — ใครยอมจ่ายแพงเพื่อได้เดี๋ยวนี้ มักรู้อะไร →

อ่านกระดาน
Reading the Book

Part 6: ทำไม Market Order สำคัญ — ใครมีข้อมูล

Order Flow & Microstructure สำหรับรายย่อย · เลนส์ execution ของซีรีส์
จาก "ของที่ยืนรอ" (OBI) สู่ "ของที่ลงมือจริง" (market order / Delta / CVD)

Part 6 — ทำไม Market Order สำคัญ

🎯 **อ่านจบ Part นี้คุณจะ**

- เข้าใจว่าทำไม **market order มีข้อมูลมากกว่า limit/cancel** (informativeness) — เพราะ "ยอมจ่ายเพื่อได้ทันที"
- รู้จัก **adverse selection**: ทำไมคนตั้ง limit ถึงเสี่ยงโดน "คนรู้ข้อมูล" กิน
- เข้าใจพฤติกรรม **non-Markovian**: คิวหมดเพราะ market → ราคา *follow* / เพราะ cancel → ราคา *revert*
- วัด "แรง market order" ได้จริงด้วย **Delta (Δ) และ CVD** จากข้อมูล Binance aggTrade

ใครยอมจ่ายแพงเพื่อได้เดี๋ยวนี้ มักรู้อะไร

ลองนึก: ของชิ้นเดียวกัน คนหนึ่ง "ตั้งรอ" ขอซื้อราคาถูกลงหน่อย อีกคน "ควักจ่ายราคาเต็ม ขอเอาเดี๋ยวนี้" — ใครดูเหมือนมีเหตุผลเร่งด่วนกว่า? คนที่ **ยอมข้าม spread จ่าย ask เต็ม ๆ เพื่อให้ได้ของทันที** มักเป็นคนที่ *เชื่อว่าราคากำลังจะวิ่งหนี* — เขายอมจ่าย "ค่าผ่านทาง" เพราะคิดว่ารออีกนิดจะแพงกว่านี้

นั่นคือเหตุผลที่ **market order เป็นสัญญาณที่ "informative"** ส่วน limit order ที่ตั้งรอ (ยกเลิกฟรีเมื่อไรก็ได้) เป็นสัญญาณที่ *อ่อน* — มันแค่ "ความตั้งใจ" ที่ถอนได้

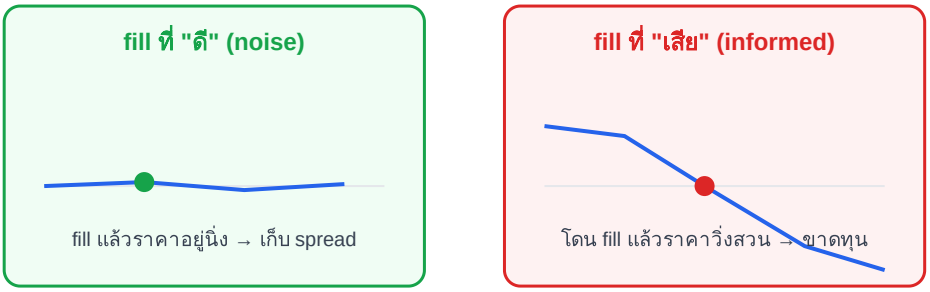
เหตุการณ์	ต้นทุน/ความมุ่งมั่น	ความ informative
Market order (taker)	จ่าย spread + เสี่ยง slippage	สูง — ลงมือจริง เอาเดี๋ยวนี้
Limit order (provide)	ฟรี ถอนได้	ต่ำ — แค่ตั้งรอ
Cancel	ฟรี	ต่ำ — ถอนความตั้งใจ

Adverse selection — กับดักของคนตั้ง limit

คุณตั้ง limit buy ที่ best bid รอกินครึ่ง spread แต่... ใครจะมาเทรดกับคุณ? ถ้าคนที่มา "ขายชน" limit ของคุณคือคนที่ **รู้ว่าราคากำลังจะลง** คุณก็เพิ่งซื้อของที่กำลังจะถูกลง — fill ของคุณคือ "ของเสีย" นี่คือ **adverse selection**: คุณมักได้ fill *ในจังหวะที่ไม่ดีสำหรับคุณ*

```
กำไรคาดหวังของ limit order (maker):
  E[กำไร] = (ครึ่ง spread) − E[ราคาวิ่งสวน | ได้ fill]
        = edge ที่หวัง − ต้นทุน adverse selection
```

```
ถ้า adverse move > ครึ่ง spread → maker ขาดทุนแม้ได้ fill!
```



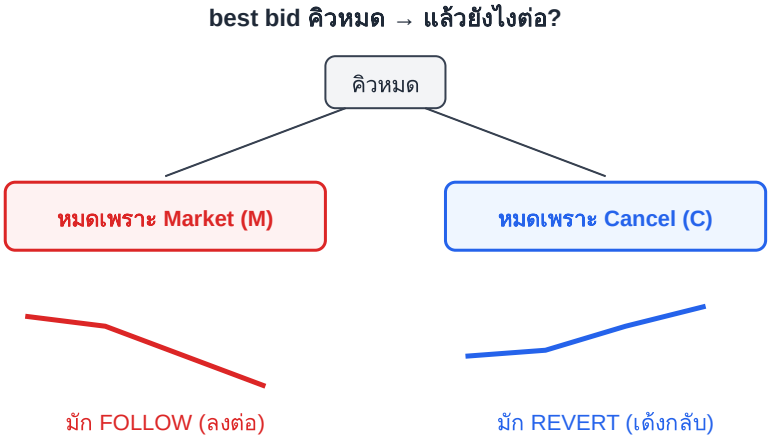
ภาพ 6.3 · maker ได้ fill ตอน "ดี" บ้าง "เสีย" บ้าง — informed trader ทำให้ฝั่งเสียหนัก

Non-Markovian — สาเหตุที่คิวหมดสำคัญกว่าตัวเลข

สมมติ best bid คิวหมดเกลี้ยง ราคากำลังจะขยับ — แต่ "หมดเพราะอะไร" สำคัญมาก:

- **หมดเพราะ market sell (M)** มากินจนเกลี้ยง → มีแรงขายจริงดันลง → ราคามักจะ **follow** (ลงต่อ)
- **หมดเพราะคน cancel (C)** ถอน limit ออกไปเอง → ไม่มีเงินเปลี่ยนมือ มักเป็นการขยับเชิงรับ → ราคามักจะ **revert** (เด้งกลับ)

นี่เรียกว่า **non-Markovian**: "สถานะปัจจุบัน" (คิวหมด) อย่างเดียวไม่พอ ต้องรู้ *ประวัติว่ามันหมดได้ยังไง* ด้วย — นี่คือเหตุผลหลักที่ OBI (ภาพนิ่ง) ไม่พอ และทำไม Part 7 (OFI) ต้องจับ "เหตุการณ์การเปลี่ยน"



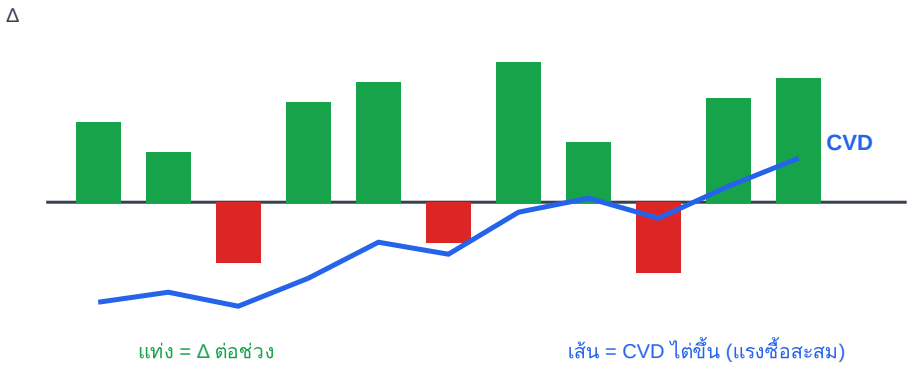
ภาพ 6.2 · (หัวใจ Part นี้) สาเหตุที่คิวหมด ชี้ทิศต่อ — M→follow / C→revert

> 📑 **งานวิจัยรองรับ — และข้อควรระวังเรื่องตัวเลข**
>
> **Lu & Abergel (2018)** และตระกูล queue-reactive models พบว่าเมื่อคิวที่ best หมดด้วย market order ราคามักเคลื่อนต่อทิศเดิม (continuation) ในสัดส่วนสูง — ตัวเลขที่มักอ้างราว **~84% continuation** นั้น **ขึ้นกับสินทรัพย์ / large-tick vs small-tick / นิยามและ calibration** ควรตรวจกับ paper ต้นฉบับและ market ของคุณเอง อย่ายึดเป็นค่าตายตัว

Delta (Δ) และ CVD — วัด "แรง market order" ได้จริง

เราวัด market order flow ได้ตรง ๆ ด้วย **Delta** = ปริมาณที่ taker ซื้อชน ลบด้วยปริมาณที่ taker ขายชน และสะสมเป็น **CVD (Cumulative Volume Delta)** เพื่อดูเทรนด์แรงซื้อ/ขาย:

```
Δ (Delta) = BuyVol − SellVol ← นับ "ของที่เทรดไปแล้ว" (aggressor)
CVD_t = Σ Δ (running sum) ← เทรนด์แรงซื้อ/ขายสะสม

การจำแนก aggressor (signed volume):
  Binance aggTrade ให้ flag m (isBuyerMaker):
    m = true → ผู้ซื้อเป็น maker → taker คือ "ขายชน" → Sell
    m = false → ผู้ขายเป็น maker → taker คือ "ซื้อชน" → Buy
  (ไม่ต้องเดาด้วย Lee-Ready เพราะ exchange บอก aggressor ให้ตรง ๆ)
```



ภาพ 6.4 · Δ แท่งเขียว(buy)/แดง(sell) + เส้น CVD สะสม

💼 Binance-ready Orderflow Toolkit — ผูกศัพท์เทรดเดอร์ ↔ ศัพท์วิชาการ ↔ paper

ศัพท์เทรดเดอร์	ศัพท์วิชาการ	วัดอะไร	Paper รองรับ
Delta (Δ)	Trade imbalance / signed volume	flow ที่เทรดไปแล้ว (taker)	Lee & Ready (1991); Chordia-Subrahmanyam (2004)
Cumulative Delta (CVD)	Cumulative signed order flow	trend แรงซื้อ/ขายสะสม	Chordia-Roll-Subrahmanyam (2002)
Volume Imbalance	Queue / order-book imbalance	state ของ depth ที่ตั้งรอ	Gould & Bonart (2015)

ข้อมูล Binance ที่ใช้: `@aggTrade` stream (ได้ flag `m` = aggressor) สำหรับ Δ/CVD · `@depth` สำหรับ Volume Imbalance/OBI

📑 Nuance จากงานวิจัย

(1) OFI > Trade Imbalance: Δ/CVD ใช้ได้ดี แต่จับเฉพาะ "ของที่เทรดไปแล้ว" — **OFI (Part 7)** รวม cancel และ limit ด้วย จึงมีข้อมูลมากกว่า นี่คือทางอัปเกรดที่ชี้ตรงไป Part 7 · **(2) พลังทำนายเสื่อมตามเวลา:** ทุกสัญญาณ orderflow มี horizon สั้น ตอกย้ำบทเรียน signal horizon จาก Part 4

⚠️ กับดักมือใหม่

- เห็น best หมดคิวแล้ว **เดาทิศทันที** — ต้องดู *สาเหตุ* (M หรือ C) ก่อน
- **CVD divergence ≠ กลับตัวเสมอ** — CVD ขึ้นแต่ราคาไม่ขึ้น อาจเป็น *absorption* (มีกำแพง limit ดูดแรงซื้อไว้) ไม่ใช่สัญญาณกลับตัวอัตโนมัติ
- จำแนก aggressor **ผิด** ทำให้ Δ ผิดทั้งแผง — ใช้ flag `m` ของ Binance ให้ถูก อย่าเดาเอง

🔗 เชื่อมโยงเล่มอื่น

"ใครมีข้อมูล" เป็นเลนส์เดียวกับเล่ม **eye** (mindset: อ่านว่าใครกำลังลงมือจริง) ส่วนแนวคิด adverse selection จะกลับมาเป็นหัวใจของ **Part 9–10 (Market Making)** — MM ต้อง skew quote เพื่อหนี informed flow และ **Part 7** จะเปลี่ยน "Δ/CVD" ให้กลายเป็น OFI ที่ใส่ลง regression วัด price impact (λ) ได้ (ตรงนี้เล่ม **math** เข้ามาเต็มตัว)

✍️ แบบฝึก (ทำได้ด้วยข้อมูลคริปโตฟรี)

เชื่อม Binance `@aggTrade` ของ BTC/USDT เก็บ 200 เทรด ติดป้าย buy/sell ด้วย flag `m` คำนวณ Δ ต่อช่วงเวลา (เช่นทุก 5 วินาที) แล้ว **plot CVD** เทียบกับเส้น mid — สังเกตช่วงที่ CVD ไต่ขึ้นแต่ราคาไม่ขึ้นตาม (= candidate absorption) นั่นคือจุดที่ "เดา CVD ตรง ๆ" จะพลาด

ต่อ Part 7: OFI & Price Impact — จากภาพนิ่งสู่วิดีโอ และวัด λ ด้วย regression →

อ่านกระดาน
Reading the Book

Part 7: Order Flow Imbalance & Price Impact — flow ดันราคายังไง

Order Flow & Microstructure สำหรับรายย่อย · เลนส์ execution ของซีรีส์

จากภาพนิ่ง (OBI) → วิดีโอการเปลี่ยน (OFI) → วัดแรงดันราคาด้วย Kyle's λ

Part 7 — OFI & Price Impact

🎯 **อ่านจบ Part นี้คุณจะ**

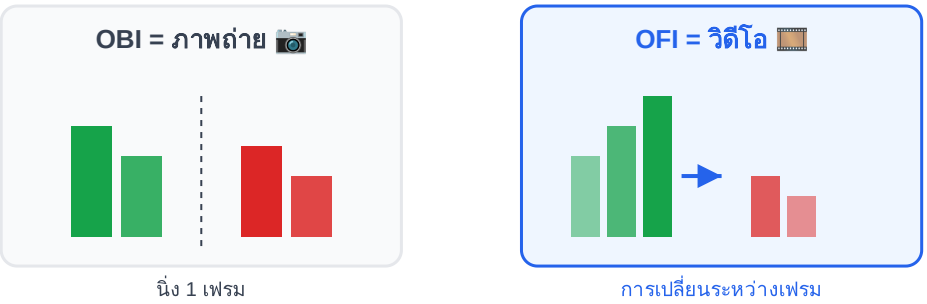
- แยก **OBI (state/ภาพนิ่ง)** ออกจาก **OFI (flow/การเปลี่ยน)** ได้ชัด (มีตารางเทียบ)
- คำนวณ OFI แบบ event-based ที่ best level ครบ 3 กรณี (ราคาขึ้น/เท่า/ลง)
- เข้าใจว่า **Kyle's λ = ความชันของเส้น regression** ΔP บน OFI — แรงดันราคาต่อหนึ่งหน่วย flow
- รู้จัก **multi-level OFI** (Cont 2023, รวมหลายชั้นด้วย PCA) และทำไมต้อง stationarize ก่อน regress

OBI = ภาพถ่าย / OFI = วิดีโอ

Part 5 สอน OBI ซึ่งเป็น "**ภาพถ่ายนิ่ง**" ของกระดาน ณ วินาทีหนึ่ง — บอกว่าตอนนี้ฝั่งไหนหนัก แต่ภาพนิ่งบอกไม่ได้ว่ามันกำลัง *เปลี่ยนไปทางไหน*

OFI (Order Flow Imbalance) คือ "**วิดีโอการเปลี่ยน**" — มันวัด *การเปลี่ยนแปลงสุทธิ* ของ depth ระหว่างสอง snapshot โดยรวมทั้ง 3 เหตุการณ์: limit เพิ่มเข้า, cancel ถอนออก, market กินออก ต่างจาก Δ/CVD (Part 6) ที่จับเฉพาะ market — OFI จับ **ทุกแรงที่ขยับกระดาน**

มิติ	OBI (Part 5)	OFI (Part 7)
วัดอะไร	**state** — ปริมาณที่ตั้งรอ ณ จุดเวลาหนึ่ง	**flow** — การเปลี่ยนสุทธิระหว่างสองช่วงเวลา
คำอุปมา	ภาพถ่ายนิ่ง	วิดีโอการเปลี่ยน
รวมเหตุการณ์	แค่ snapshot (ไม่รู้สาเหตุ)	limit + cancel + market รวมกัน
ความสัมพันธ์กับราคา	correlation (เดาทิศ)	**regression เชิงสาเหตุ** ΔP=λ·OFI
จับ "การเปลี่ยน" ไหม	ไม่ — เป็นภาพนิ่ง	ใช่ — เป็นแกนกลาง



ภาพ 7.1 · OBI (state) เทียบ OFI (flow)

นิยาม OFI ที่ best level — 3 กรณี

OFI event-based (Cont-Kukanov-Stoikov) วัดการเปลี่ยน depth ที่ best bid/ask ระหว่าง snapshot ก่อน (n−1) กับตอนนี้ (n) โดยขึ้นกับว่า **ราคา best ขยับหรือไม่**:

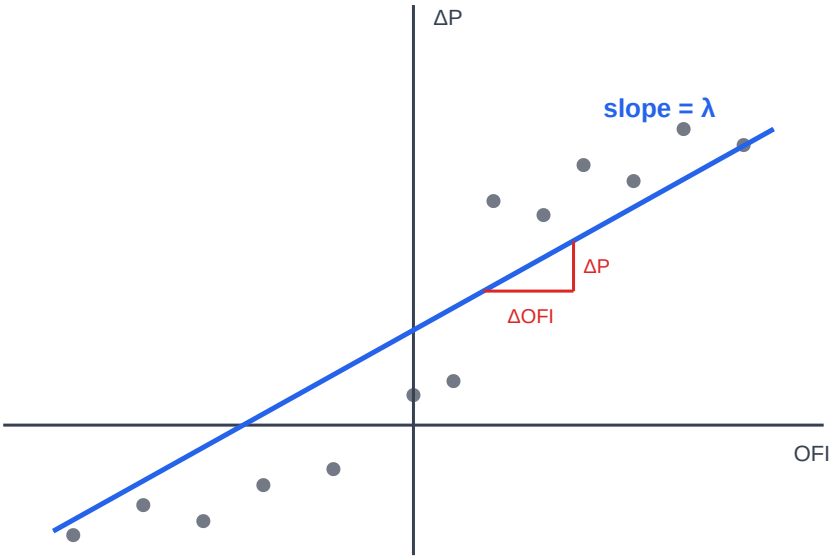
```
ฝั่ง bid (e^b_n) — เทียบ P^b กับ P^b ก่อนหน้า:
  ถ้า P^b_n > P^b_{n−1} → e^b = V^b_n (ราคา bid ขึ้น = แรงซื้อเข้า)
  ถ้า P^b_n = P^b_{n−1} → e^b = V^b_n − V^b_{n−1} (ราคาเท่า = ดูปริมาณเปลี่ยน)
  ถ้า P^b_n < P^b_{n−1} → e^b = −V^b_{n−1} (ราคา bid ลง = แรงขายกิน)

ฝั่ง ask (e^a_n) — สมมาตรกลับด้าน:
  ถ้า P^a_n < P^a_{n−1} → e^a = V^a_n
  ถ้า P^a_n = P^a_{n−1} → e^a = V^a_n − V^a_{n−1}
  ถ้า P^a_n > P^a_{n−1} → e^a = −V^a_{n−1}

OFI_n = e^b_n − e^a_n (บวก = แรงดันขึ้น, ลบ = แรงดันลง)
```

Kyle's λ — แรงดันราคาต่อหนึ่งหน่วย flow

คำถามสำคัญ: OFI หนึ่งหน่วยดันราคาเท่าไร? ตอบด้วย **regression เชิงเส้น** ของการเปลี่ยน mid (ΔP) บน OFI ความชัน (slope) ที่ได้คือ **Kyle's λ** — "ค่าความลึก/แพงของตลาด" (price impact coefficient)

```
ΔP = λ · OFI + ε ← สมการ price impact (linear)

λ = ความชันเส้น regression (OLS slope):
  λ = Cov(ΔP, OFI) / Var(OFI) ← สูตร OLS ที่เล่ม math สอน
```

```
λ สูง = ตลาดบาง (flow นิดเดียวดันราคาแรง)
λ ต่ำ = ตลาดลึก (flow มากแต่ราคาขยับน้อย — liquid)
```



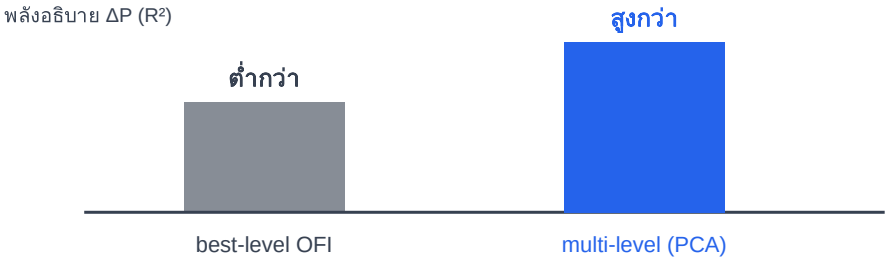
แต่ละจุด = หนึ่งช่วงเวลา · เส้นผ่านจุดกำเนิด (two-sided) · ความชัน = price impact

ภาพ 7.2 · (ภาพสำคัญ) scatter ΔP บน OFI — เส้น regression ที่ความชัน = Kyle's λ

Multi-level OFI + การ stationarize

OFI ที่ best level เดียวอาจไม่พอ **Cont, Cucuringu & Zhang (2023)** รวม OFI หลายชั้น (multi-level) ซึ่งมักมี correlation สูงระหว่างกัน จึงใช้ **PCA** ยุบเหลือองค์ประกอบหลัก (integrated/PC1 OFI) ก่อนใส่ regression — เพิ่มพลังอธิบายได้

```
multi-level: OFI^(k) สำหรับชั้น k = 1..L → integrated OFI =
PCA(OFI^(1)..OFI^(L)) เอา PC1

stationarize ก่อน regress (กันผลลวง/spurious):
  OFI_norm = OFI / DepthBar (หารด้วย depth เฉลี่ยช่วงนั้น → สเกลคงที่)
```



ภาพ 7.3 · multi-level OFI มัก explain ΔP ได้ดีกว่า best-level (เชิงสัมพัทธ์ — ค่าจริงขึ้นกับตลาด)

📑 **งานวิจัยรองรับ — และข้อควรระวังเรื่องตัวเลข**

Cont, Kukanov & Stoikov (2014) วาง OFI event-based และความสัมพันธ์เชิงเส้น ΔP=λ·OFI · **Cont, Cucuringu & Zhang (2023)** ขยายเป็น multi-level + cross-impact ด้วย PCA ตัวเลขที่มักอ้างว่า OFI อธิบาย ΔP ได้ **R^2~65%** (สูงกว่า OBI/trade imbalance) นั้น **ขึ้นกับสินทรัพย์ / timeframe / จำนวนชั้น / วิธี stationarize** อย่ายึดเป็นค่าตายตัว — ให้ผู้อ่าน calibrate กับ market ตัวเองเสมอ

⚠️ **กับดักมือใหม่**

- **regress OFI ดิบโดยไม่ stationarize** → ได้ R^2 หลอก (spurious) เพราะ depth สเกลเปลี่ยนตามเวลา ต้อง normalize ด้วย DepthBar ก่อน
- สับสน OBI กับ OFI — **OBI=ภาพนิ่ง, OFI=การเปลี่ยน** ใส่ OBI ลงสมการ price impact ผิดที่
- เชื่อว่า λ คงที่ — λ เปลี่ยนตามสภาพ liquidity (เวลาข่าว/ตลาดบางก็พุ่ง)

🔗 **เชื่อมโยงเล่มอื่น (เน้น math)**

Part นี้คือจุดที่เล่ม **math** เชื่อมเต็มตัว: **Kyle's λ = OLS slope** = Cov(ΔP,OFI)/Var(OFI) ตรงกับบทถดถอยเชิงเส้น · **R^2** = สัดส่วนความแปรปรวนที่อธิบายได้ · **multi-level OFI = PCA** ยุบตัวแปร correlated ตรงกับบท PCA ทั้งหมดนี้นำไปต่อ **Part 13 (stat-arb ด้วย log-OFI)** และเป็นรากของ **Part 8 (VPIN)** ที่เปลี่ยนจาก "แรงดันราคา" เป็น "ความเป็นพิษของ flow"

✍️ **แบบฝึก (ทำได้ด้วยข้อมูลคริปโตฟรี)**

ดึง Binance `@depth` ของ BTC/USDT มาทำ snapshot ต่อเนื่อง คำนวณ OFI ที่ best level ต่อช่วง 1 วินาที และ ΔP (การเปลี่ยน mid) แล้ว **รัน regression สองชุด**: (1) ΔP ~ OFI, (2) ΔP ~ OBI เทียบ **R^2** ของทั้งสอง — คุณจะเห็นด้วยมือว่าทำไม "การเปลี่ยน (OFI)" อธิบายราคาได้ดีกว่า "ภาพนิ่ง (OBI)" ลองเพิ่มการ stationarize (หาร DepthBar) แล้วดูว่า R^2 เปลี่ยนไหม

ต่อ Part 8: VPIN — เมื่อ flow ไม่ใช่แค่ดันราคา แต่ "เป็นพิษ" จนตลาดพัง →

อ่านกระดาน
Reading the Book

Part 8: VPIN — ความเป็นพิษของ flow (toxicity)

Order Flow & Microstructure สำหรับรายย่อย · เลนส์ execution ของซีรีส์

ปิดท้ายองก์ Flow: static(OBI) → flow(OFI) → **toxicity(VPIN)**

Part 8 — VPIN / ความเป็นพิษของ flow

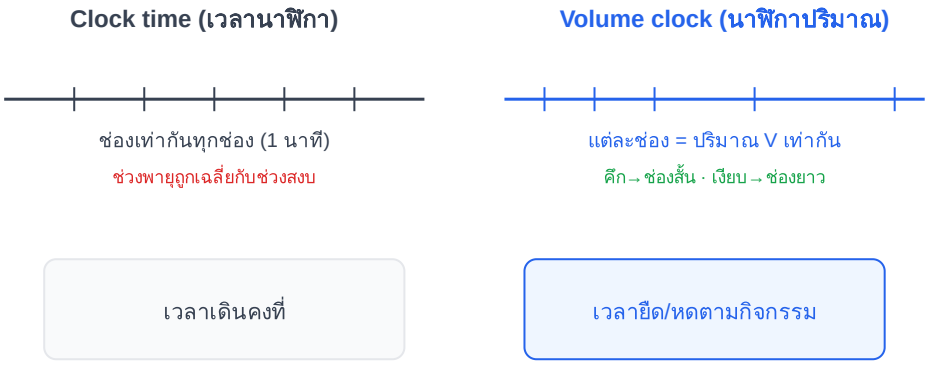
🎯 **อ่านจบ Part นี้คุณจะ**

- เข้าใจ **volume clock**: ทำไมต้องวัดเวลาเป็น "ปริมาณ" ไม่ใช่ "วินาที"
- คำนวณ **VPIN** (Volume-synchronized Probability of Informed Trading) จาก volume buckets
- เข้าใจว่า toxicity = adverse selection สะสมของ MM และทำไม VPIN พุ่งก่อน flash crash
- รู้ **ข้อโต้แย้ง**: VPIN ไม่ใช่ปุ่มทำนาย crash — ผลขึ้นกับ calibration อย่างมาก

นาฬิกาเดินตามปริมาณ ไม่ใช่เวลา

นาฬิกาปกติเดินทุก 1 วินาทีเท่ากันหมด — แต่ตลาดไม่ได้ "เกิดเหตุการณ์" สม่ำเสมอ บางวินาทีเงียบกริบ บางวินาที (ตอนข่าวออก) มีเทรดเป็นพัน ๆ ถ้าวัดทุกอย่างด้วย "เวลานาฬิกา" คุณจะเฉลี่ยช่วงพายุกับช่วงสงบเข้าด้วยกัน ทำให้พลาดสัญญาณ

VPIN ใช้ "นาฬิกาปริมาณ" (volume clock) — แทนที่จะนับ "ผ่านไป 1 นาที" มันนับ "ผ่านไปอีก V หน่วยปริมาณที่เทรด" หนึ่งช่อง (bucket) = ปริมาณคงที่ V เสมอ ช่วงตลาดคึก buckets จะเต็มเร็ว ช่วงเงียบจะเต็มช้า — เวลาจึง "ยืด/หด" ตามกิจกรรมจริง

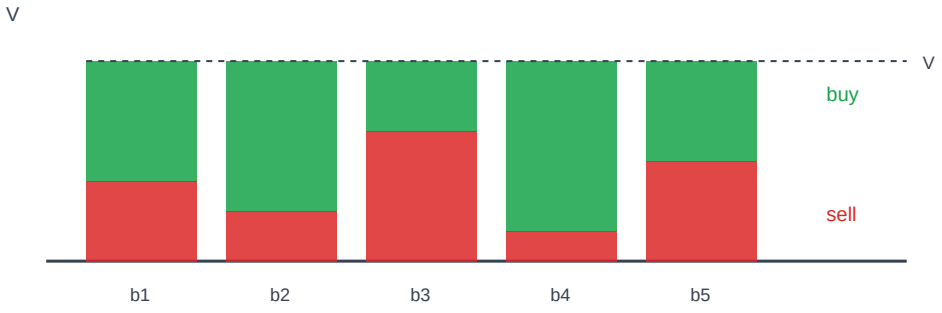


ภาพ 8.1 · clock time (ช่องเท่ากัน) เทียบ volume clock (ปริมาณเท่ากัน)

Volume buckets และ bulk classification

เราหั่นกระแสเทรดเป็น buckets ปริมาณเท่ากัน (เช่น V = ปริมาณเฉลี่ยต่อวัน/50) ในแต่ละ bucket เราประมาณว่า **ปริมาณนั้นเป็นฝั่งซื้อกี่ส่วน ขายกี่ส่วน** ด้วย *bulk volume classification* (ใช้การเปลี่ยนราคาในช่วง bucket ผ่านฟังก์ชัน Φ = CDF ของ normal):

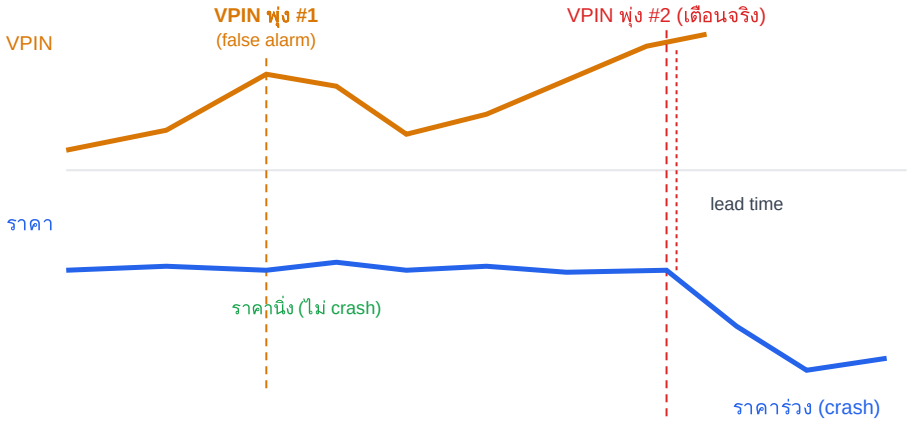
```
1 bucket = ปริมาณ V คงที่ (กำหนดล่วงหน้า)

แบ่งฝั่งใน bucket ด้วย bulk classification:
  V_buy = V · Φ( ΔP / σ_ΔP ) ← Φ = CDF ของ standard normal (ไม่ใช่ z-score)
  V_sell = V − V_buy
  (ΔP = การเปลี่ยนราคาในช่วง bucket, σ_ΔP = ส่วนเบี่ยงเบนของ ΔP)

VPIN = (1 / (n·V)) · Σ_{i=1..n} | V_buy,i − V_sell,i |
  (เฉลี่ยความไม่สมดุล |buy−sell| ข้าม n buckets ล่าสุด)
```



ภาพ 8.2 · ทุก bucket ปริมาณ V เท่ากัน แยกสัดส่วน buy/sell — VPIN = เฉลี่ย |buy−sell|

Toxicity = adverse selection สะสม → เตือน crash

VPIN สูง = flow "**ลำเอียงไปฝั่งเดียวหนัก**" ติดต่อกัน = น่าจะมี informed trader ลุยอยู่ = MM ที่ทำหน้าที่ตั้ง quote กำลังโดน **adverse selection** หนัก (โดนกินฝั่งเดียว) เมื่อ MM เจ็บมากพอ พวกเขาจะ **ถอนสภาพคล่อง** spread กว้างขึ้น และถ้าหนีพร้อมกัน → **flash crash**



ภาพ 8.3 · VPIN พุ่งแล้วราคาร่วงจริง (#2) ก็มี — แต่ **VPIN พุ่งแล้วไม่ crash ก็มี (false alarm, #1)** จึงเป็นมาตรวัดความเสี่ยง ไม่ใช่ปุ่มทำนาย

📑 **งานวิจัยรองรับ — และข้อโต้แย้งเรื่อง calibration**

Easley, López de Prado & O'Hara (2012) เสนอ VPIN และแสดงว่ามันสูงผิดปกติก่อน Flash Crash 6 พ.ค. 2010 งานสาย crypto (nowcasting Bitcoin crash with order imbalance) ก็พบสัญญาณคล้ายกัน **แต่มีข้อโต้แย้งสำคัญ**: นักวิจัยหลายกลุ่ม (เช่น Andersen-Bondarenko) ชี้ว่าพลังทำนายของ VPIN **อ่อนไหวต่อ calibration อย่างมาก** — ขนาด bucket V, จำนวน buckets n, วิธี classify ฝั่ง ล้วนเปลี่ยนผลลัพธ์ และบางส่วนของสัญญาณอาจเป็นแค่ผลของ volume/volatility ทั่วไป ไม่ใช่ "informed trading" จริง **อย่าใช้ VPIN เป็นปุ่มทำนาย crash เด็ดขาด**

⚠️ **กับดักมือใหม่**

- ใช้ VPIN เป็น **"ปุ่มทำนาย crash"** — มันเป็น *มาตรวัดความเสี่ยง/toxicity* ไม่ใช่สัญญาณซื้อขายตรง ๆ และมี false alarm
- คำนวณ VPIN ด้วย **เวลานาฬิกา** (time bars) — ผิดทั้งกระบวน! ต้องใช้ **volume clock** (volume buckets เท่านั้น)
- ลืม normalize/calibrate แล้วเทียบค่า VPIN ข้ามเหรียญ — ค่าดิบเทียบกันตรง ๆ ไม่ได้

🔗 **เชื่อมโยงเล่มอื่น**

VPIN ปิดเส้นเรื่อง **static(OBI) → flow(OFI) → toxicity(VPIN)**: OBI ดูภาพนิ่ง, OFI ดูการเปลี่ยน+price impact (λ), VPIN ดู "อันตราย" ของ flow แนวคิด **adverse selection** ที่สะสมใน VPIN คือต้นทุนเดียวกับที่ MM ต้องบริหารใน **Part 9–10 (Market Making)** และจะถูกใช้เป็น **kill switch หยุด grid** ใน **Part 12** (VPIN>θ → ระงับไม้ทวนเทรนด์) ส่วนวิธีคิด volume clock เชื่อมกับการทำ feature ใน **Part โบนัส (ML)**

✍️ **แบบฝึก (ทำได้ด้วยข้อมูลคริปโตฟรี)**

ดึง Binance `@aggTrade` ของ BTC/USDT 1 ชั่วโมง หั่นเป็น **volume buckets** ขนาด V คงที่ (เลือก V ให้ได้ราว 50 buckets) ในแต่ละ bucket แยก buy/sell ด้วย bulk classification แล้วคำนวณ **VPIN แบบ rolling** (เฉลี่ย n=50 buckets ล่าสุด) plot เทียบราคา — จากนั้น **ลองเปลี่ยนขนาด V และ n** แล้วดูว่ากราฟ VPIN เปลี่ยนแค่ไหน นี่คือบทเรียนตรง ๆ ว่าทำไม calibration สำคัญ (และทำไมต้องระวังการ over-trust)

จบองก์ "อ่าน Flow" · ต่อ Part 9: Market Making พื้นฐาน — กินสปรดยังไงโดยไม่โดน inventory ทับ →

อ่านกระดาน
Reading the Book

Part 9: MM-1 พื้นฐาน — กินสปรดยังไงให้รอด

Order Flow & Microstructure สำหรับรายย่อย · องก์ IV Market Making

อ่านจบแล้วเข้าใจกลไกกินสปรด, maker vs taker + rebate, inventory risk และสัญชาตญาณ skew quote

Part 9 — MM-1 พื้นฐาน: กินสปรดยังไง

🎯 **อ่านจบ Part นี้คุณจะ**

- เข้าใจว่า market maker (MM) ทำเงินจาก **"กินสปรด"** ได้อย่างไร — ตั้ง bid และ ask พร้อมกัน แล้วเก็บส่วนต่าง
- แยก **maker vs taker** และรู้ว่าทำไม rebate ในตลาดคริปโตทำให้ MM ได้เปรียบ
- เข้าใจ **inventory risk** — ความเสี่ยงที่อันตรายที่สุดของ MM ไม่ใช่สปรดแคบ แต่คือสต็อกค้างมือ
- จับสัญชาตญาณของ **skew quote** — เอียงราคาทั้งสองฝั่งเพื่อระบายสต็อก (ปูทางสู่สูตร Avellaneda-Stoikov ใน Part 10)

เริ่มจากร้านแลกเงินสนามบิน

นึกถึงร้านแลกเงินที่สนามบิน — เขาไม่ได้เก็งกำไรว่าค่าเงินจะขึ้นหรือลง เขาแค่ **"รับซื้อถูก ขายแพง"** ตลอดทั้งวัน คนเดินทางมาเรื่อย ๆ บางคนเอาดอลลาร์มาแลกบาท บางคนเอาบาทมาแลกดอลลาร์ ร้านอยู่ตรงกลางคอยจับคู่และเก็บส่วนต่างทุกครั้ง

Market maker ก็คือร้านแลกเงินบนกระดานเทรด — ตั้ง bid (รับซื้อ) ต่ำกว่า mid นิดหน่อย และตั้ง ask (ขาย) สูงกว่า mid นิดหน่อย ถ้ามีคนมา "กิน" ทั้งสองฝั่งสลับกัน MM ก็เก็บสปรดไปเรื่อย ๆ โดยไม่ต้องเดาทิศทางราคาเลย

แต่มีปัญหาหนึ่งที่ร้านแลกเงินก็เจอ — **ลิ้นชักเต็มไปด้วยดอลลาร์ แต่ไม่มีคนเอาบาทมาแลกคืน** ถ้าดอลลาร์ราคาตกระหว่างนั้น ร้านขาดทุนทันที นี่แหละคือ *inventory risk* หัวใจของทั้ง Part นี้

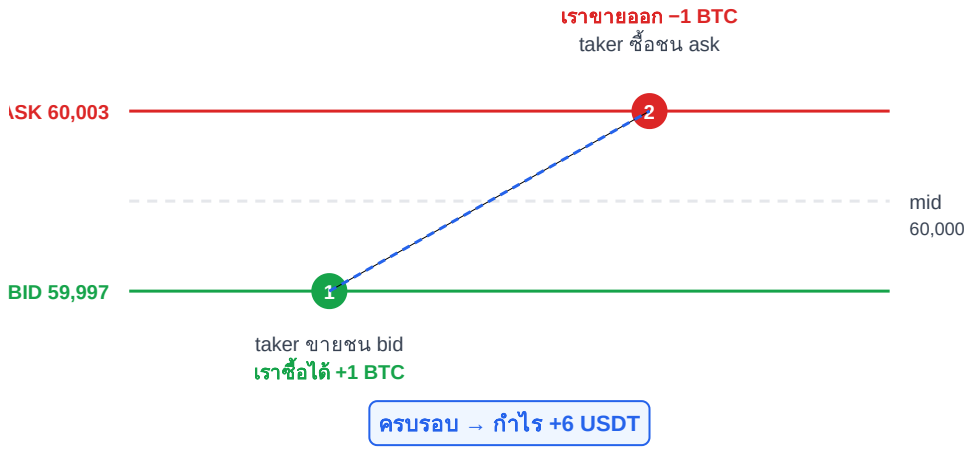
เก็บสปรด 6 ทุกครั้งที่จับคู่ครบรอบ (ซื้อ→ขาย)
ไม่เดาทิศทาง — กลัวอย่างเดียวคือลิ้นชักเอียงข้างเดียว

กลไกกินสปรด: ทำเงินโดยไม่เดาทิศ

สมมติ BTC มี mid = 60,000 MM ตั้ง quote สองฝั่ง:

- **Bid 59,997** — รอรับซื้อ (เป็น limit order ฝั่งซื้อ → เป็น maker)
- **Ask 60,003** — รอขาย (เป็น limit order ฝั่งขาย → เป็น maker)

ถ้ามี taker คนหนึ่ง "ขายทันที" มาชน bid ของเรา → เราได้ BTC มาที่ 59,997 จากนั้นมี taker อีกคน "ซื้อทันที" มาชน ask ของเรา → เราขาย BTC ออกที่ 60,003 ครบรอบหนึ่ง **เราเก็บ 6 USDT** โดยที่ราคา BTC แทบไม่ขยับเลย



ภาพ 9.1 · กลไกกินสปรด — ซื้อที่ bid ขายที่ ask ไม่ต้องเดาทิศ

```
PnL ต่อรอบ (ครบ buy→sell) = (ask − bid) = 60,003 − 59,997 = 6 USDT
ถ้ามี maker rebate r = (ask − bid) + 2·r (ได้ rebate ทั้งสองขา)
ตัวอย่าง rebate 0.002% = 6 + 2·(0.00002·60,000) = 6 + 2.4 = 8.4 USDT/รอบ
```

Maker vs Taker + Rebate คริปโต

หัวใจที่ทำให้ MM มีกำไรไม่ใช่แค่สปรด แต่คือ **โครงสร้างค่าธรรมเนียม** ในตลาดส่วนใหญ่:

- **Maker** = คนตั้ง limit รอ (เติมสภาพคล่อง) → จ่าย fee ต่ำ บางตลาดถึงขั้น **ได้เงินคืน (rebate)**
- **Taker** = คนยิง market กินทันที (ดึงสภาพคล่องออก) → จ่าย fee สูงกว่า + จ่ายสปรด

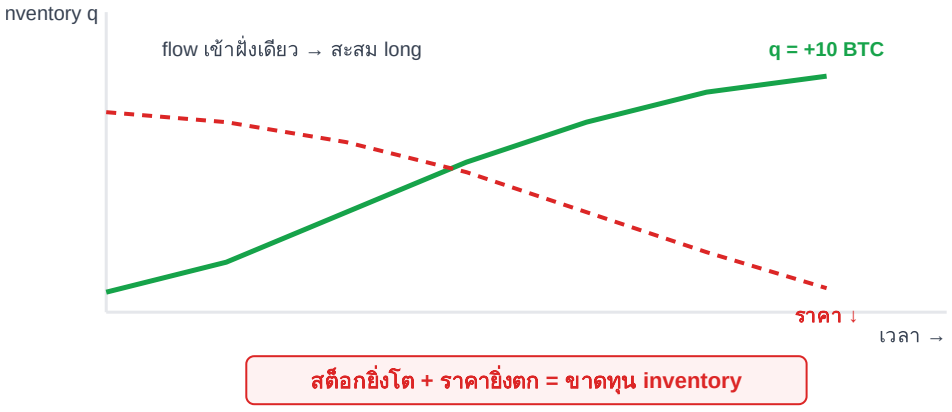
มิติ	Maker (MM)	Taker
บทบาท	เติมสภาพคล่อง (limit)	กินสภาพคล่อง (market)
ค่าธรรมเนียม	ต่ำ / ได้ rebate	สูงกว่า
ได้ fill ทันทีไหม	ไม่ (queue risk)	ได้ทันที
ความเสี่ยงหลัก	adverse selection + inventory	slippage

> 📑 **fee/rebate เปลี่ยนได้ — เช็คก่อนใช้จริง**
>
> โครงสร้าง maker/taker fee และ rebate ของ Binance/Bybit **เปลี่ยนตามโปรโมชันและระดับ VIP เสมอ** บางคู่ในบางช่วง maker fee = 0% หรือติดลบ (rebate) ตัวเลขในตัวอย่างเป็นเพียงการสาธิต — ก่อนรันจริงต้องเปิดหน้า fee schedule ของตลาดนั้นและคำนวณใหม่ทุกครั้ง เพราะ **edge ของ MM รายย่อยบาง ๆ มาก ค่าธรรมเนียมที่คลาดเคลื่อนนิดเดียวพลิกจากกำไรเป็นขาดทุนได้**

Inventory risk: ศัตรูตัวจริงของ MM

กลับมาที่ร้านแลกเงิน — ถ้าทั้งวันมีแต่คนเอาดอลลาร์มาแลกบาท ไม่มีใครเอาบาทมาแลกคืน ลิ้นชักของร้านจะ **เต็มไปด้วยดอลลาร์ฝั่งเดียว** ตอนนี้ร้านไม่ได้เป็นกลางอีกแล้ว — ร้านกำลัง "ถือดอลลาร์" และถ้าดอลลาร์ตก ร้านขาดทุน

บนกระดานก็เหมือนกัน ถ้า flow ไม่สมดุล (มีคนขายชน bid ของเรารัว ๆ) เราจะ **สะสม inventory ฝั่ง long** เรื่อย ๆ — เก็บสปรดได้ก็จริง แต่ถ้าราคาร่วงลงต่อ ขาดทุนจาก inventory จะกลบสปรดที่เก็บมาทั้งหมด



ภาพ 9.2 · inventory สะสมเมื่อ flow ไม่สมดุล

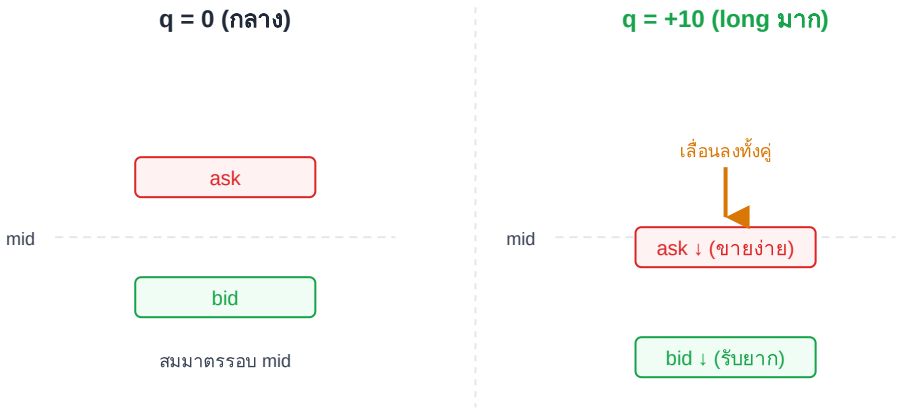
```
PnL รวม = PnL_spread + PnL_inventory
PnL_inventory = q · Δp (q = สต็อกสุทธิ, Δp = ราคาเปลี่ยน)
ตัวอย่าง: q = +10 BTC, ราคาตก Δp = −50 → PnL_inv = 10·(−50) = −500 USDT
สปรดที่เก็บมา 30 รอบ × 6 = +180 → รวมสุทธิ = −320 (ขาดทุน!)
```

> ⚠️ **กับดักมือใหม่ — "สปรดกว้าง = ปลอดภัย"**
>
> มือใหม่มัก "ตั้งสปรดกว้าง ๆ ไว้ก่อนจะได้กำไรเยอะต่อรอบ" — **ผิดมุม!** สปรดกว้างทำให้ quote ของคุณอยู่ไกล mid โอกาส fill ต่ำ และที่สำคัญกว่า: **คนที่ยอมมา fill quote ห่าง ๆ ของคุณ มักเป็นคนที่ "รู้ว่าราคากำลังจะวิ่งมาทางคุณ"** (adverse selection — Part 3/6) งานจริงของ MM ไม่ใช่ตั้งสปรดให้กว้าง แต่คือ **จัดการ inventory** ให้ลิ้นชักไม่เอียงข้างเดียว สปรดเป็นแค่ค่าตอบแทน ความเสี่ยงตัวจริงคือสต็อกค้างมือ

สัญชาตญาณ skew: เอียงราคาเพื่อระบายสต็อก

ถ้าลิ้นชักเต็มไปด้วย BTC (long มากไป) MM ฉลาดจะทำอะไร? — **"อยากขายมากกว่าซื้อ"** วิธีง่ายที่สุดคือเลื่อน quote ทั้งสองฝั่งลง: ลด ask ให้คนมาซื้อจากเราง่ายขึ้น (ระบาย BTC ออก) และลด bid ให้คนมาขายใส่เรายากขึ้น (ไม่อยากรับเพิ่ม) — นี่คือ **skew**

สังเกตว่า skew ไม่ได้แปลว่า "เดาว่าราคาจะลง" — มันแปลว่า "ฉันถือเยอะไป ขอเลื่อนราคายุติธรรมในใจฉันลงมาเพื่อจูงใจให้ตลาดช่วยระบายสต็อก" นี่คือ **เมล็ดพันธุ์ของ reservation price** ที่จะกลายเป็นสูตรเต็มใน Part 10



ภาพ 9.3 · skew quote — ถือ long มาก เลื่อน quote ลงเพื่อระบายสต็อก (ปูทาง Part 10)

```
สูตรสัญชาตญาณ (เวอร์ชันง่าย ก่อนเข้า A-S เต็ม):
bid_skew = mid − δ/2 − k·q
ask_skew = mid + δ/2 − k·q
  δ = สปรดฐาน, q = inventory สุทธิ, k = ความแรงของการ skew (k>0)
q > 0 (long) → ทั้ง bid,ask เลื่อนลง → อยากขายมากกว่าซื้อ
q < 0 (short) → ทั้ง bid,ask เลื่อนขึ้น → อยากซื้อมากกว่าขาย
```

📑 งานวิจัยรองรับ

แนวคิด "MM ต้องถ่วงดุล inventory ด้วยการเอียงราคา" ถูกวางรากฐานโดย **Ho & Stoll (1981)** ที่มองว่า dealer ปรับ quote ตามระดับสต็อกเพื่อคุมความเสี่ยง และ **Glosten & Milgrom (1985)** ที่อธิบายว่าสปรดส่วนหนึ่งมีไว้ป้องกัน adverse selection — ทั้งสองงานคือฐานที่ **Avellaneda & Stoikov (2008)** นำมาเขียนเป็นสูตร reservation price ใน Part 10 และต่อยอดสู่ยุค reinforcement learning ในเวลาต่อมา

🔗 เชื่อมโยงเล่มอื่น

PnL ของ MM = PnL_spread + q·Δp นั้นคือ **payoff ที่มี inventory เป็นตัวคูณความเสี่ยง** — ตรงกับมุมมองใน **Payoff Mastery (pm)** ที่ว่ากำไรขาดทุนต้องคิดบนตำแหน่งจริง ไม่ใช่แค่ราคา ส่วนเครื่องมือคำนวณ k, δ และการ optimize จะใช้ความรู้ regression/optimization จากเล่ม **math** เต็มตัวใน Part 10

✍️ แบบฝึก (ทำได้ด้วยข้อมูลคริปโตฟรี)

จำลองด้วยมือ: สมมติ mid = 60,000 คงที่, สปรด δ = 6, คุณตั้ง bid 59,997 / ask 60,003 ระหว่างวันมี taker มา fill **bid 7 ครั้ง และ ask 3 ครั้ง** (อย่างละ 1 BTC)

1. คำนวณ inventory สุทธิ q หลังจบวัน (เฉลย: q = +7 − 3 = +4 BTC)
2. คำนวณ PnL_spread ที่เก็บได้จากคู่ที่จับครบ — **1 รอบ = คู่ buy+sell ที่จับครบ** ดังนั้น fill 10 ครั้ง (bid 7 + ask 3) จับเป็นคู่ได้แค่ min(7,3) = 3 รอบ ส่วน bid อีก 4 ครั้งค้างเป็น inventory (3 รอบ × 6 = 18 USDT)
3. ถ้าตอนปิดราคาตก 50 USDT คำนวณ PnL_inventory = q·Δp (= 4·(−50) = −200) แล้วรวมสุทธิ
4. ลองใส่ skew k = 1.5: คำนวณ bid_skew/ask_skew ใหม่ตอน q = +4 แล้วอธิบายว่าทำไมมันช่วยลดความเสี่ยง

จากนั้นเปิด Binance ดู maker/taker fee จริงของคู่ BTC/USDT แล้วคำนวณว่าสปรด 6 USDT ยังเหลือกำไรหลังหัก fee ไหม

ต่อ Part 10: MM-2 Optimal — วาง quote ตรงไหนถึงจะ optimal (Avellaneda-Stoikov → RL) →

อ่านกระดาน
Reading the Book

Part 10: MM-2 Optimal — วาง quote ตรงไหนถึงดีที่สุด

Order Flow & Microstructure สำหรับรายย่อย · องก์ IV Market Making

Avellaneda-Stoikov: reservation price, optimal spread, adverse selection, value function → ยุค RL

Part 10 — MM-2 Optimal: Avellaneda-Stoikov

🎯 **อ่านจบ Part นี้คุณจะ**

- เข้าใจ **reservation price** = "ราคายุติธรรมในใจพ่อค้า" ที่ขยับตามสต็อก — หัวใจของ Avellaneda-Stoikov
- อ่านสูตร **$r = s - q \cdot \gamma \cdot \sigma^2 \cdot (T-t)$** ออกว่าแต่ละตัวแปรหมายถึงอะไรและทำไมมันสมเหตุสมผล
- เข้าใจ **optimal spread** ว่ามาจากการชั่งน้ำหนักระหว่างความเสี่ยง inventory กับโอกาส fill
- เห็นชัดว่า **naive MM (quote สมมาตรไม่ skew) ตายเพราะ adverse selection** ขณะ optimal รอด — ผ่านกราฟ P&L
- เห็นภาพ **เส้นทางวิวัฒนาการ** จาก Ho-Stoll → A-S → Guéant → reinforcement learning

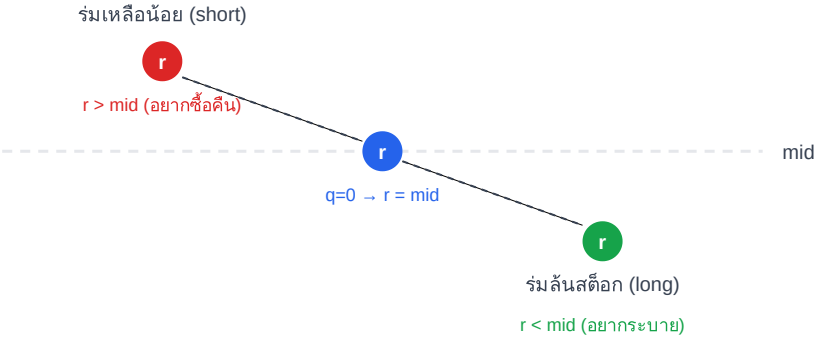
ราคายุติธรรมในใจพ่อค้าขายร่ม

นึกถึงพ่อค้าขายร่มข้างถนน เช้านี้เขามีร่ม 50 คัน "ราคายุติธรรมในใจ" ของเขาคือ 100 บาท เขาจึงตั้งรับซื้อ 95 ขาย 105 — สมมาตรรอบ 100

ผ่านไปครึ่งวัน **ขายดีมากจนเหลือร่มแค่ 5 คัน** ฝนกำลังจะตก เขากลัวของหมดก่อนปิดร้าน — "ราคายุติธรรมในใจ" ของเขา **ขยับขึ้นเป็น 110** โดยอัตโนมัติ เขาไม่อยากขายถูกแล้ว เขาตั้งขาย 115 รับซื้อ 105 (อยากซื้อร่มเข้าสต็อกคืน)

กลับกัน ถ้าเหลือร่ม **200 คัน** และใกล้ฟ้าจะเปิด เขากลัวของค้าง "ราคาในใจ" จะ **ตกลงเหลือ 90** เขายอมขายถูกเพื่อระบายสต็อก

"ราคายุติธรรมในใจที่ขยับตามสต็อก" นี่แหละคือ reservation price — มันไม่ใช่ราคาตลาด (mid) แต่เป็นราคาที่พ่อค้าใช้เป็นจุดศูนย์กลางในการตั้ง quote ของตัวเอง ยิ่งถือเยอะ ยิ่งดึงให้ต่ำกว่า mid เพื่อเร่งระบาย



ภาพ 10.1 · reservation price ถูกดึงออกจาก mid ตามทิศและขนาดของ inventory

สูตร reservation price

Avellaneda & Stoikov (2008) เขียน "ราคาในใจ" นี้เป็นสมการ:

```
r = s − q · γ · σ² · (T − t)

r = reservation price (ราคายุติธรรมในใจ MM)
s = mid price ปัจจุบัน
q = inventory สุทธิ (long = บวก, short = ลบ)
γ = ความกลัวความเสี่ยง (risk aversion) — γ สูง = ขี้กลัว
σ² = ความผันผวน (variance ของราคา)
(T − t) = เวลาที่เหลือก่อนปิดรอบ — ใกล้ปิดยิ่งกลัว inventory น้อยลง
```

อ่านสูตรแบบไม่ต้องท่อง — แค่ดูว่า **"ตัวลบ" q·γ·σ²·(T−t) ดึง r ออกจาก mid เท่าไหร่**:

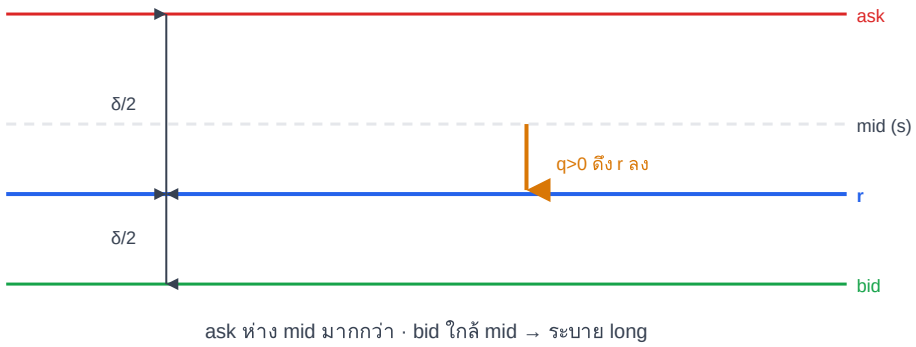
- **q บวกมาก (long เยอะ)** → ตัวลบใหญ่ → r ต่ำกว่า mid มาก → quote เอียงลง → เร่งขายระบาย (ตรงกับพ่อค้าร่มที่ของล้น)
- **γ สูง (ขี้กลัว)** → ตัวลบใหญ่ → ยิ่งกลัวยิ่งดึง r หนีจาก mid เร็ว
- **σ² สูง (ผันผวน)** → ถือ inventory อันตรายขึ้น → ดึง r หนีแรงขึ้น
- **(T−t) เล็ก (ใกล้ปิด)** → ตัวลบเล็กลง → ใกล้หมดเวลาแล้ว ถือ inventory ไม่กี่วินาทีไม่น่ากลัว → r เข้าใกล้ mid อีกครั้ง

Optimal spread: กว้างเท่าไหร่ถึงดี

มี r แล้วต้องตั้ง bid/ask ห่างจาก r เท่าไหร่? A-S ให้สูตร **optimal spread** ที่ชั่งน้ำหนัก 2 แรง: ความเสี่ยง inventory (อยากกว้างไว้ปลอดภัย) กับ ความถี่ fill (อยากแคบไว้ได้เทรดบ่อย)

```
δ_total = γ·σ²·(T − t) + (2/γ)·ln(1 + γ/κ)
  δ_total = สปรดรวมสองฝั่ง (bid+ask) — คนละตัวกับ δ ฐานใน Part 9
  ก้อนแรก = ค่าชดเชยความเสี่ยง inventory (เหมือนใน r)
  ก้อนสอง = ค่าชดเชยจาก order arrival — κ คือความไวของ fill ต่อระยะ quote

bid = r − δ_total/2
ask = r + δ_total/2
```

จุดสำคัญที่มือใหม่มักพลาด: **bid/ask ของ A-S วางสมมาตรรอบ r ไม่ใช่รอบ mid** เพราะ r เอียงไปตาม inventory อยู่แล้ว ผลลัพธ์คือ quote สองฝั่งจริง ๆ ไม่สมมาตรรอบ mid — ฝั่งที่อยากระบายจะอยู่ใกล้ mid กว่า



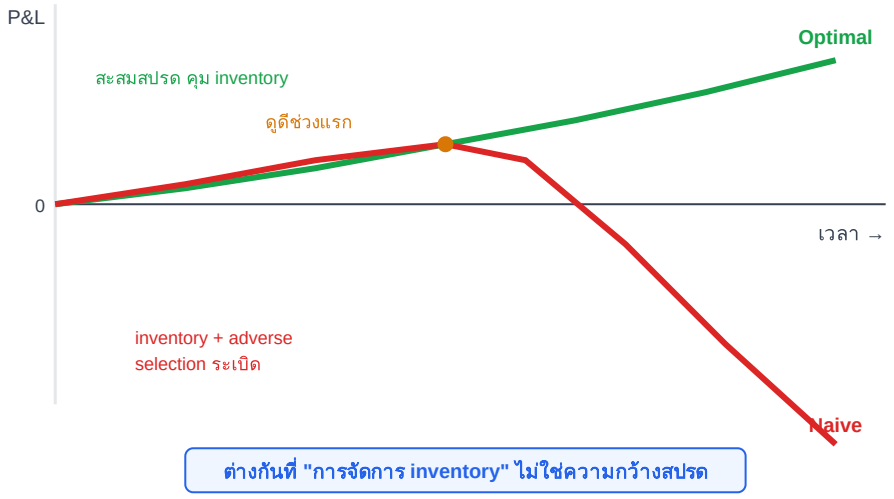
ภาพ 10.2 · bid/ask สมมาตรรอบ r (ที่ถูกดึงต่ำกว่า mid เพราะ long)

ทำไม naive MM ถึงตาย — adverse selection

Naive MM = ตั้ง bid/ask สมมาตรรอบ *mid* เสมอ ไม่สนใจ inventory ฟังดูเป็นกลางดี แต่มันตายเพราะ 2 เหตุผลที่ทบกัน:

1. **Inventory วิ่งหนีไม่หยุด** — ไม่มีกลไก skew ดึงกลับ ตลาดเทรนด์เดียวจะอัด inventory ฝั่งเดียวจนพอร์ตระเบิด
2. **Adverse selection** — คนที่มา fill quote ของเรา "บ่อยกว่า" คือคนที่ถูกข้าง เวลาราคากำลังจะขึ้น คนแห่มากิน ask ของเรา (เราขายถูกให้เขา) เวลาราคากำลังจะลง คนแห่มาชน bid ของเรา (เรารับแพง) — **MM มักโดน fill ในจังหวะที่ผิดเสมอ**

Optimal MM สู้ adverse selection ด้วยการ skew (ผ่าน r): พอ inventory เริ่มเอียง r จะดึง quote ให้ฝั่งที่เสี่ยงอยู่ไกล mid ขึ้น ลดโอกาสโดน fill ซ้ำเติม นี่คือเหตุผลที่กราฟ P&L ต่างกันราวฟ้ากับเหว



ภาพ 10.3 · P&L optimal vs naive — naive ดูดีช่วงแรกแล้วระเบิดเพราะ adverse selection + inventory

> ⚠️ **กับดักมือใหม่ — "เป็นกลางรอบ mid = ปลอดภัยสุด"**
>
> หลายคนคิดว่าการตั้ง quote สมมาตรรอบ mid คือ "ไม่เลือกข้าง = ปลอดภัย" — **ตรงกันข้าม** มันคือสูตรของการโดน adverse selection เต็ม ๆ เพราะคุณปล่อยให้ตลาดเลือกข้างให้คุณ (คนถูกข้างมา fill คุณก่อนเสมอ) ความปลอดภัยจริงมาจากการ **ยอมเอียง quote ตาม inventory** เพื่อบังคับให้สต็อกกลับเข้าศูนย์ — γ ไม่ใช่ "ความขลาด" แต่คือ "วินัยคุมความเสี่ยง" ที่ใส่เป็นตัวเลขในสมการ

Value function: ที่มาของสูตร (มองผ่าน ๆ พอ)

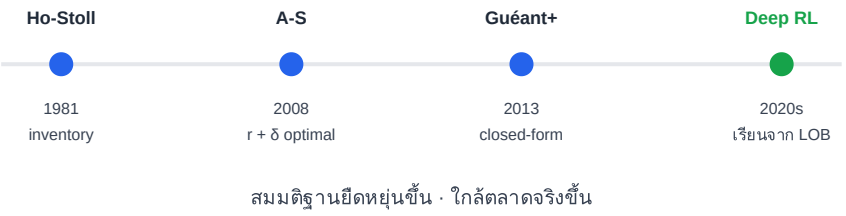
สูตร r และ δ ไม่ได้เดาเอา — มันมาจากการแก้ปัญหา optimization ที่ถามว่า "ตั้ง quote ยังไงให้ **มูลค่าคาดหวังของความมั่งคั่งสุดท้าย** สูงสุด โดยหักโทษความเสี่ยง" คณิตศาสตร์เบื้องหลังคือ *value function* u(s, q, t) ที่แก้ผ่านสมการ HJB (Hamilton-Jacobi-Bellman)

```
แนวคิด (ไม่ต้องแก้เอง):
u(s, q, t) = max E[ −exp(−γ · wealth สุดท้าย) ]
  wealth สุดท้าย = เงินสด + q · (ราคาเลิกกิจการ)
แก้ HJB → ได้ r และ δ_total ข้างต้นเป็นคำตอบ
```

ไม่ต้องแก้สมการเองได้ ขอแค่จับ **สัญชาตญาณ**: u ลงโทษ q ที่ใหญ่ (ผ่าน −exp(−γ·...)) ดังนั้นคำตอบที่ optimal จึงต้อง "พยายามดึง q กลับศูนย์เสมอ" — ซึ่งออกมาเป็นกลไก skew ที่เราเห็นใน r พอดี

จาก A-S สู่ยุค Reinforcement Learning

A-S สวยแต่ตั้งอยู่บนสมมติฐานที่สวยเกินจริง (ราคาเป็น Brownian, order arrival เป็น Poisson คงที่, ไม่มี market impact) โลกจริงไม่เป็นแบบนั้น เลยมีการต่อยอดเรื่อยมา:



ภาพ 10.4 · วิวัฒนาการ inventory model → A-S → closed-form → RL

ยุค **Reinforcement Learning** เปลี่ยนจาก "แก้สมการหา quote" เป็น "ให้เอเจนต์ลองตั้ง quote ในสภาพแวดล้อมจำลองล้านครั้งแล้วเรียนรู้นโยบายเอง" โดยมีฟังก์ชันรางวัลที่ลงโทษ inventory คล้ายเดิม:

```
RL objective: max E[ Σ (PnL_t − λ·q_t²) ]
  PnL_t = กำไรแต่ละก้าว, λ·q_t² = โทษการถือ inventory (แทน γ ใน A-S)
```

```
เอเจนต์เรียน policy π(state) → action (ตั้ง bid/ask offset) เอง
state = [inventory, spread, OBI, OFI, volatility, ...]
```

📑 งานวิจัยรองรับ (A-S → RL)

Avellaneda & Stoikov (2008) คือต้นฉบับของ reservation price + optimal spread ต่อยอดจาก **Ho & Stoll (1981)** · **Guéant-Lehalle-Fernandez-Tapia (2013)** ให้คำตอบ closed-form ที่จัดการ inventory limit ได้ดีขึ้น · ยุคหลังมีงาน RL เช่น *Market Making with Deep RL from LOB* (arXiv 2305.15821) และ *Resolving Latency & Inventory Risk* (arXiv 2505.12465) ที่ให้เอเจนต์เรียน quoting policy จากข้อมูล LOB จริง — **แต่ทั้งหมดยังตั้งบนหลักเดียวกัน: ลงโทษ inventory เพื่อสู้ adverse selection**

🔗 เชื่อมโยงเล่มอื่น

สูตร A-S คือการ **optimize ภายใต้ความไม่แน่นอน** — ใช้เครื่องมือ calculus/optimization และแนวคิด variance/risk จากเล่ม **math** โดยตรง (HJB คือ optimization แบบ dynamic) ส่วนการมองกำไรหลังหักโทษความเสี่ยง (PnL − λq^2) คือ **risk-adjusted payoff** ที่ตรงกับหัวใจของ **Payoff Mastery (pm)** — เทรดที่ดีไม่ใช่กำไรสูงสุด แต่คือกำไรต่อหน่วยความเสี่ยงที่ดีที่สุด

✍️ แบบฝึก (เขียนโค้ดสั้น ๆ ได้)

เขียนฟังก์ชันคำนวณ reservation price และ quote แล้วทดลองเล่นตัวแปร:

```
def quote(s, q, gamma, sigma, T_minus_t, kappa):
    r = s - q * gamma * sigma**2 * T_minus_t
    delta = gamma*sigma**2*T_minus_t +
(2/gamma)*math.log(1+gamma/kappa)
    bid = r - delta/2
    ask = r + delta/2
    return r, bid, ask
```

1. เรียกด้วย s=60000, q=0, แล้วเทียบ q=+10 และ q=−10 — สังเกตว่า r เลื่อนไปทางไหน และ quote ฝั่งไหนเข้าใกล้ mid
2. เพิ่ม γ จาก 0.1 → 1.0 ดูว่า quote กว้างขึ้นหรือแคบลง และ r หนีจาก mid เร็วขึ้นไหม
3. ลด (T−t) ลงเรื่อย ๆ จนใกล้ 0 ดูว่า r เข้าใกล้ mid อีกครั้งไหม (ใกล้ปิดรอบ)
4. เขียน loop จำลอง fill ตาม Poisson เทียบ P&L ของ optimal (มี skew) กับ naive (สมมาตรรอบ mid) บนชุดราคาเทรนด์เดียว — ดูว่า naive ระเบิดจริงไหม

ต่อ Part 11: MM-3 ลงมือจริงคริปโต — จาก quote สู่ระบบรันได้ + กับดัก backtest →

อ่านกระดาน
Reading the Book

Part 11: MM-3 ลงมือจริง — จาก quote สู่ระบบรันได้

Order Flow & Microstructure สำหรับรายย่อย · องก์ IV Market Making

MM loop, microprice, post-only, inventory limit + kill switch และกับดัก backtest 3 ข้อ

Part 11 — MM-3 ลงมือจริง + กับดัก backtest

🎯 **อ่านจบ Part นี้คุณจะ**

- เห็นโครง **MM loop** ที่รันได้จริง: รับ L2 + trade stream → คำนวณ → ตั้ง/ยกเลิก quote วน
- รู้จัก **microprice** — ราคาอ้างอิงที่ดีกว่า mid เพราะถ่วงน้ำหนักด้วยความหนาแน่นสองฝั่ง
- ใช้ **post-only** เพื่อการันตีเป็น maker และตั้ง **inventory limit + kill switch** กันพอร์ตระเบิด
- รู้ทัน **กับดัก backtest 3 ข้อ** ที่ทำให้กราฟสวยกลายเป็นขาดทุนจริง: fill assumption, latency, market impact

backtest คือกระจกห้องลองเสื้อ

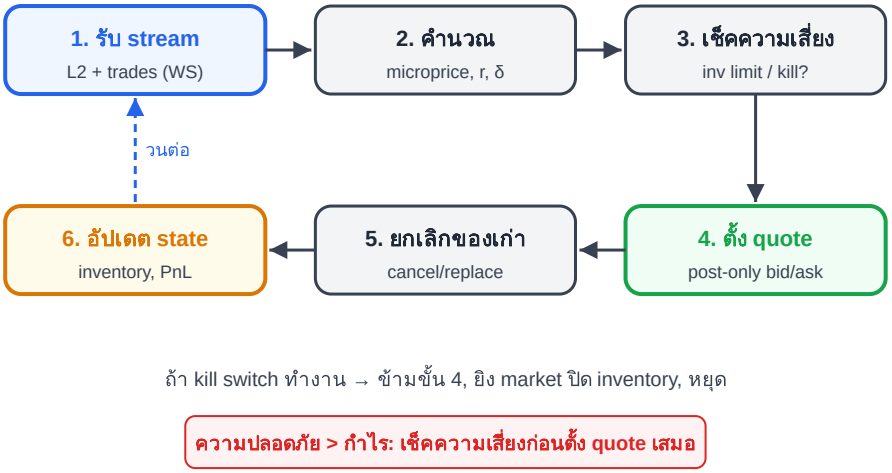
เคยลองเสื้อในห้องลองที่ห้างไหม? — กระจกเอียงนิด ไฟวอร์มนิด ทุกตัวดูดีไปหมด พอซื้อกลับบ้านมาส่องกระจกตัวเองถึงรู้ว่า "มันไม่เหมือนในห้องลองเลย"

Backtest ของกลยุทธ์ MM ก็คือกระจกห้องลองเสื้อ — มันถูกออกแบบ (โดยตั้งใจหรือไม่ตั้งใจ) ให้ผลลัพธ์ดูสวยกว่าความจริง เพราะมันตั้งสมมติฐานเข้าข้างตัวเองหลายอย่าง โดยเฉพาะเรื่อง "เราได้ fill ทุก quote ที่ราคาแตะ" ซึ่งในตลาดจริง **ไม่จริงเลย**

Part นี้แบ่งสองครึ่ง: ครึ่งแรกประกอบ "ร่างกาย" ของ MM ที่รันได้จริง ครึ่งหลังคือ "กระจกเงาความจริง" — สามกับดัก backtest ที่ต้องรู้ก่อนเอาเงินจริงไปเสี่ยง

MM loop: ร่างกายของระบบ

ระบบ MM ที่รันได้จริงคือลูปที่หมุนตลอดเวลา รับข้อมูลสองสาย (L2 order book + trade stream ผ่าน WebSocket) แล้วตัดสินใจตั้ง/ยกเลิก quote:

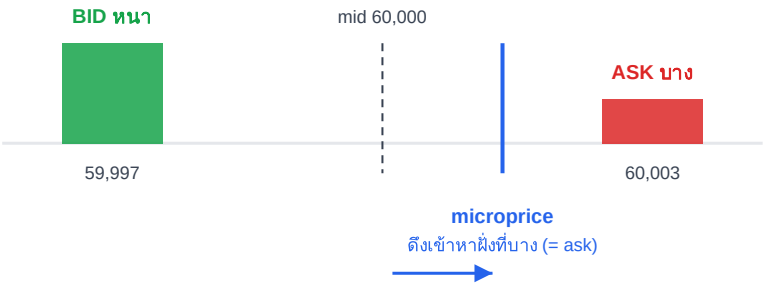


ภาพ 11.1 · MM loop — รับ stream → คำนวณ → เช็คความเสี่ยง → ตั้ง/ยกเลิก quote → วน

Microprice: ราคาอ้างอิงที่ดีกว่า mid

mid = (bid+ask)/2 มองสองฝั่งเท่ากันเสมอ แต่ถ้าฝั่ง bid หนากว่าฝั่ง ask มาก ๆ ราคา "จริง ๆ" มีแนวโน้มถูก **ดึงเข้าหาฝั่งที่บาง (= ask) microprice** แก้จุดนี้ด้วยการถ่วงน้ำหนักด้วยปริมาณฝั่งตรงข้าม:

```
microprice = (p_bid · V_ask + p_ask · V_bid) / (V_bid + V_ask)

สังเกต: ฝั่งที่ "หนากว่า" จะดึง microprice เข้าหาฝั่งที่บาง (ฝั่งตรงข้าม)
V_bid > V_ask (แรงซื้อหนา) → microprice ดึงเข้าหาฝั่งที่บาง (= ask) → มีแนวโน้มขึ้น
เชิงสถิติระยะสั้น
ใช้ microprice แทน s ในสูตร reservation price ของ Part 10 ได้
```



ภาพ 11.1b · microprice ดึงเข้าหาฝั่งที่บาง (= ask) เมื่อ bid หนากว่า ask

Post-only, inventory limit, kill switch

เครื่องมือ	ทำอะไร	ทำไมขาดไม่ได้
post-only	limit ที่ยกเลิกอัตโนมัติถ้าจะกลายเป็น taker	การันตีเป็น maker เสมอ → ได้ rebate / ไม่จ่าย taker fee + spread โดยบังเอิญ
inventory limit	หยุดตั้ง quote ฝั่งที่ทำให้ |q| เกินเพดาน	กัน inventory วิ่งหนีฝั่งเดียวจนเกินที่รับความเสี่ยงไหว
kill switch	ปิด inventory ทันที + หยุดบอท เมื่อขาดทุน/ผันผวนเกินกำหนด	กันหายนะตอนตลาดเกิดเหตุผิดปกติ (flash crash, ข่าวแรง)

```
# pseudo-code: ก้อนความปลอดภัยใน MM loop
if abs(inventory) >= INV_LIMIT:
    skip_quote_on_growing_side() # ไม่เติมฝั่งที่ทำให้ q โตขึ้น
if pnl_drawdown > MAX_DD or vol > VOL_CAP:
    market_close_all(); halt() # kill switch
else:
    place_order(bid, post_only=True)
    place_order(ask, post_only=True)
```

กับดัก backtest 3 ข้อ (หัวใจของ Part)

นี่คือ "กระจกเงาความจริง" — สามเหตุผลที่กราฟ backtest สวยมักโกหก:

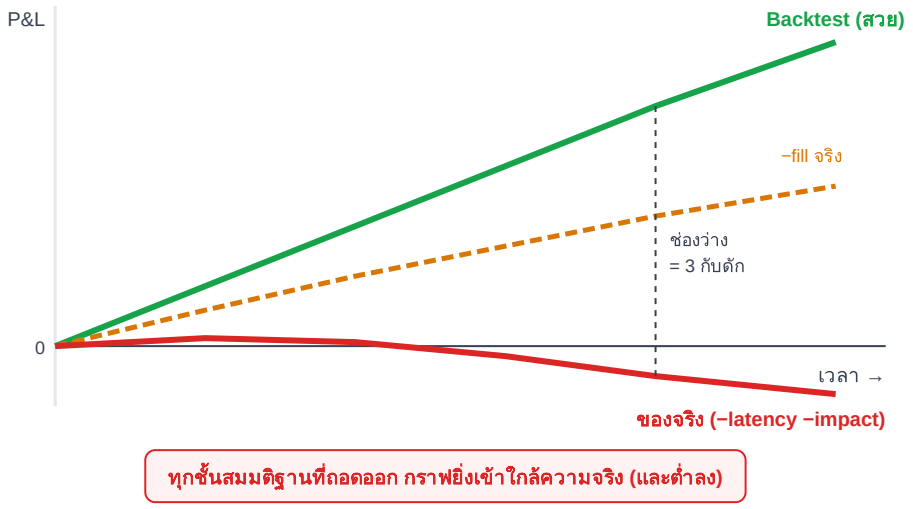
⚠️ **กับดัก 1 — Fill assumption (อันตรายที่สุด)**

Backtest ส่วนใหญ่สมมติว่า "**ถ้าราคาแตะ quote ของเรา = เราได้ fill**" — ผิดมหันต์! ในความจริงคุณอยู่ **ท้ายคิว** (Part 1) ราคาต้องแตะ *แล้วเทรดทะลุคิวข้างหน้าคุณหมด* คุณถึงได้ fill และร้ายกว่านั้น: quote ที่ได้ fill ง่ายมักเป็นจังหวะ **adverse selection** (ราคากำลังวิ่งสวนคุณ) backtest ที่ assume fill ฟรีจึงนับกำไรที่ไม่มีจริง และมองข้ามการขาดทุนที่มีจริง

⚠️ **กับดัก 2 — Latency (ความหน่วง)**

Backtest มองเห็นข้อมูลและตอบสนอง "ทันที 0 วินาที" แต่ของจริงมี latency ทั้งขารับข้อมูลและขายิงคำสั่ง กว่าคุณจะเห็น microprice ขยับแล้วยกเลิก quote เก่า ตลาดเคลื่อนไปแล้ว — **quote เก่าที่ยังค้างอยู่คือ quote ที่โดน "เก็บ" ในจังหวะที่แย่ที่สุด** รายย่อยที่แข่ง latency กับ HFT คือเอาจักรยานไปแข่งกับรถแข่ง

> ⚠️ **กับดัก 3 — Market impact (เราคือส่วนหนึ่งของตลาด)**
>
> Backtest รันบนข้อมูลในอดีต **ที่ไม่มีตัวเราอยู่ในนั้น** มันสมมติว่า order ของเราไม่กระทบกระดานเลย — แต่ของจริง quote ของเราเปลี่ยน OBI, เบียดคิวคนอื่น, และถ้าเราต้องยิง market ปิด inventory เราจะ walk the book ดันราคาสวนตัวเอง (Part 2) **ยิ่งไม้ใหญ่ ยิ่ง impact มาก กำไรในกระดาษหายไปกับ impact ที่ backtest มองไม่เห็น**



ภาพ 11.2 · backtest สวย vs ความจริง — fill assumption + latency + market impact กินกำไรหมด

บันได backtest → paper → live

อย่ากระโดดจาก backtest ไป live ตรง ๆ ให้ไต่บันได เพื่อให้ความจริงค่อย ๆ เผยตัว:

- **Backtest** — เร็ว ถูก แต่โกหกได้ (3 กับดัก) ใช้คัดไอเดียที่แย่ชัด ๆ ออกเท่านั้น
- **Paper / forward test** — รันสด ๆ บนตลาดจริงด้วยเงินปลอม เห็น latency จริง เห็นว่า quote ได้ fill จริงไหม
- **Live ไม้เล็กสุด** — เงินจริงจำนวนน้อยมาก วัด **adverse fill rate (AFR)** และ PnL_net หลังหัก fee จริง ค่อยขยายถ้ารอด

```
microprice = (p_bid·V_ask + p_ask·V_bid)/(V_bid+V_ask)
PnL_net = PnL_spread + q·Δp + rebate − taker_fee_on_closes − slippage
AFR = adverse_fills / total_fills (ยิ่งสูง = โดน adverse selection เยอะ)
```

📑 งานวิจัยรองรับ

ความสำคัญของ queue position และ fill ที่ไม่ใช่ "แตะแล้วได้" สะท้อนในงานสาย queue dynamics เช่น **Lu & Abergel (2018)** และ *Importance of Order Sizes* (arXiv 2405.18594) ส่วนการต่อยอด A-S ให้รับมือ latency/inventory จริงมีใน *Resolving Latency & Inventory Risk in MM* (arXiv 2505.12465) และงาน RL บน LOB ที่ไม่นิ่ง (arXiv 2509.12456) — ทั้งหมดตอกย้ำว่า **ช่องว่างระหว่าง backtest กับ live คือ fill realism, latency และ impact** ไม่ใช่ความฉลาดของสูตร

🔗 เชื่อมโยงเล่มอื่น

microprice ใช้ OBI/depth จาก **Part 5** และ adverse selection จาก **Part 6** โดยตรง · PnL_net คือ payoff ที่หักต้นทุนจริงครบทุกชั้น ตรงกับวินัยของ **Payoff Mastery (pm)** ที่ย้ำว่าต้องคิดต้นทุนการเข้า-ออกจริง · ส่วน effective price ตอนปิด inventory (walk the book) คือต้นทุนเดียวกับที่เล่ม **arb** ใช้คำนวณว่าโอกาส arb คุ้มจริงไหม

✍️ แบบฝึก (ทำได้ด้วยข้อมูลคริปโตฟรี)

เป้าหมาย: "จับโกหก" ตัวเองให้ได้ก่อนเสียเงินจริง

1. ดู pseudo-code ลูปด้านล่าง หา **fill assumption ที่อันตราย** แล้วอธิบายว่าทำไมมันทำให้ backtest สวยเกินจริง
2. แก้ฟังก์ชัน fill ให้สมจริงขึ้น: fill ได้ก็ต่อเมื่อ volume ที่เทรดผ่านราคานั้น > คิวที่อยู่ข้างหน้าเรา
3. ใส่ latency จำลอง: หน่วงการเห็นข้อมูลและการยกเลิก quote สัก 100–300 ms แล้ววัด AFR ใหม่
4. เทียบ P&L_net 3 เวอร์ชัน: (ก) fill ฟรี (ข) fill ตามคิว (ค) fill ตามคิว + latency — ดูว่ากราฟต่ำลงทีละชั้นจริงไหม

```
# pseudo-code มีกับดัก — หา fill assumption ที่ผิด
for tick in backtest_data:
    bid, ask = quote(microprice(tick), inventory, ...)
    if tick.low <= bid: # <-- กับดัก! แตะ = fill?
        inventory += 1; cash -= bid
    if tick.high >= ask: # <-- กับดักเดียวกัน
        inventory -= 1; cash += ask
# ของจริง: ต้องเช็คคิวข้างหน้า + latency + ไม่นับ fill ที่ราคาเพิ่งทะลุผ่านเฉย ๆ
```

ต่อ Part 12: Grid Trading กับกระดาน — ยกระดับ grid ด้วย OBI/OFI และ VPIN กันกริดแตก →

อ่านกระดาน
Reading the Book

Part 12: Grid Trading กับกระดาน — ยกระดับ grid ด้วย order flow

Order Flow & Microstructure สำหรับรายย่อย · เลนส์ execution ของซีรีส์

เปลี่ยน grid bot ตายตัวให้ "มองเห็นกระดาน" — เอียงตาม imbalance, หยุดเมื่อ flow เป็นพิษ

Part 12 — Grid Trading กับกระดาน

🎯 **อ่านจบ Part นี้คุณจะ**

- เข้าใจว่า **grid trading คือ market making ที่มี "กฎตายตัว"** — วาง bid/ask เป็นชั้น ๆ รอกินส่วนต่าง
- รู้จุดอ่อนคลาสสิกของ grid: **เทรนด์เดียวล้างพอร์ต** และ **กำไรต่อไม้ < fee×2**
- **เอียง (skew) grid ด้วย OBI/OFI** เพื่อหลบฝั่งที่กระดานกำลังกดดัน
- ใช้ **VPIN เป็นสวิตช์หยุด grid** ก่อนกระดานจะ "แตก" และตั้ง **spacing ไดนามิก** ตาม spread/volatility จริง

เริ่มจากตาข่ายจับปลาหลายชั้น

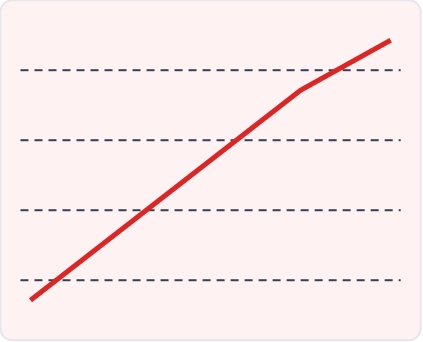
นึกถึงชาวประมงวาง **ตาข่ายหลายชั้นขวางลำน้ำ** — ปลาที่ว่ายไป-มาในระดับน้ำปกติจะติดตาข่ายเรื่อย ๆ ทีละตัว นั่นคือกำไรของ grid ในตลาด **sideway** (ราคาแกว่งขึ้นลงในกรอบ)

แต่ถ้า **น้ำหลาก** (เทรนด์แรงทิศเดียว) — ตาข่ายทุกชั้นจะถูกกระแสน้ำพัดไปกองรวมกันด้านเดียว แล้วขาดยับ นั่นคือเวลาที่ grid bot "ถือของฝั่งผิดเต็มมือ" แล้วล้างพอร์ต **กระดาน (order book) คือเครื่องวัดกระแสน้ำ** ที่ช่วยให้เรารู้ก่อนว่าน้ำกำลังจะหลาก

น้ำนิ่ง (sideway) → กำไร

ติดตาข่ายทีละตัว = กินส่วนต่าง

น้ำหลาก (trend) → ตาข่ายขาด



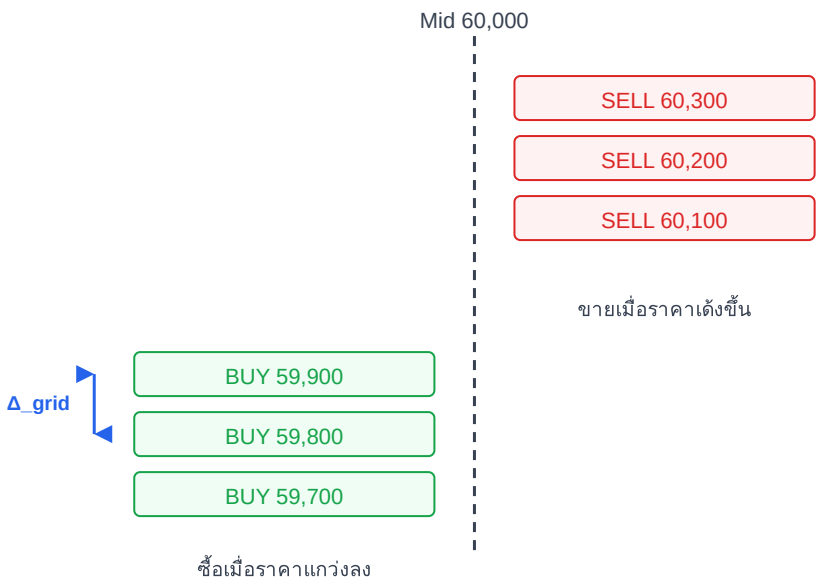
ถือของฝั่งผิดเต็มมือ = ล้างพอร์ต

ชั้นที่ 1 — grid คือ MM ที่มีกฎตายตัว

ใน Part 9–11 เราเรียน market making ที่ปรับ quote ตลอดเวลา **grid trading คือเวอร์ชันที่ "ล็อกราคา quote ไว้ล่วงหน้า" เป็นชั้น ๆ** ห่างเท่ากัน ไม่ขยับตามทุก tick:

- วาง **limit buy (bid)** เป็นบันไดลงด้านล่าง mid — เหมือน MM ฝั่งให้สภาพคล่องซื้อ

- วาง **limit sell (ask)** เป็นบันไดขึ้นด้านบน mid — ฝั่งให้สภาพคล่องขาย
- เมื่อราคาแกว่งลงมาโดน bid → ซื้อได้ แล้วตั้ง sell ที่ชั้นบนถัดไป → ขายปิดกินส่วนต่าง **Δ_grid**



ภาพ 12.1 · grid ladder — บันได bid/ask รอบ mid (ต่อยอดจากเทมเพลตแม่)

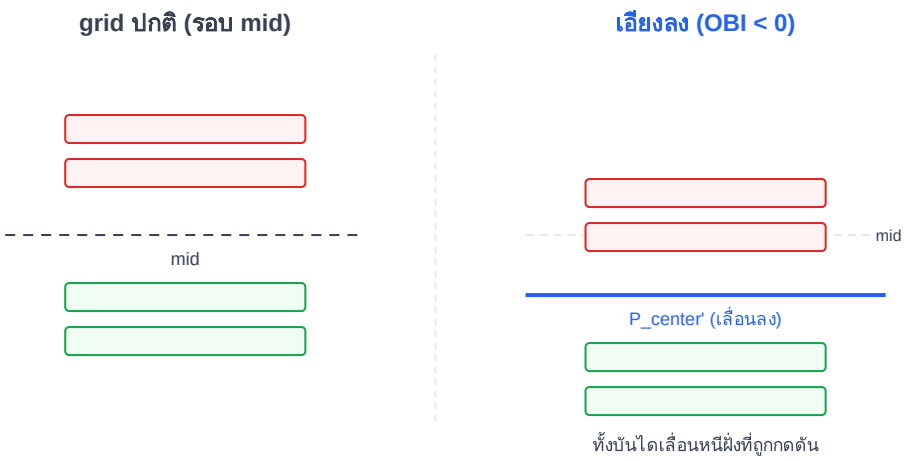
> ⚠️ **ความจริงรายย่อย — grid ที่ "ตาบอด" คือเครื่องเก็บ fee ให้ exchange**
>
> grid bot ทั่วไปไม่มองกระดานเลย — มันเด้งกินส่วนต่างไป-มาในตลาด sideway ดูสวย แต่มี **2 จุดตายที่คนส่วนใหญ่เจ็บ**: (1) **กำไรต่อไม้ < ค่าธรรมเนียม×2** — ทุกไม้ต้องจ่าย fee ทั้งขาเข้าและขาออก ถ้า Δ_grid แคบกว่า fee 2 เด้ง คุณเด้งบ่อยแต่ยิ่งเด้งยิ่งจน (2) **เทรนด์เดียวล้างพอร์ต** — พอราคาวิ่งทางเดียว grid จะ "ซื้อตลอดทาง" สะสมของฝั่งผิดเต็มมือแล้วลอยตัวขาดทุนหนัก order book ช่วยแก้ทั้งสองข้อนี้ได้

ชั้นที่ 2 — เอียง grid ด้วย OBI/OFI

grid ตายตัววางศูนย์กลางไว้ที่ mid เสมอ แต่ถ้ากระดานบอกว่า **ฝั่งขายหนักกว่าฝั่งซื้อ (OBI ติดลบ / OFI ไหลออก)** แปลว่ามีโอกาสราคาจะลงต่อ — เราควร **เลื่อนศูนย์กลาง grid ลง** เพื่อไม่ให้ bid ของเราโดนกินก่อนเวลาอันควร (หลีกเลี่ยง adverse selection จาก Part 6)

นี่คือการยืม **reservation price จาก Avellaneda-Stoikov (Part 10)** มาใช้: แทนที่จะวาง grid รอบ mid เฉย ๆ เราวางรอบ "ราคากลางที่ปรับด้วยแรงกระดาน"

ให้ชัดเรื่องทิศ: **OBI < 0 (ขายหนัก) → P_center' เลื่อนลง = ลดไม้ฝั่งซื้อไม่ให้โดนกินก่อนราคาวิ่งหนีลง** และในทางกลับกัน OBI > 0 (ซื้อหนัก) → P_center' เลื่อนขึ้น ทั้งบันได grid จึง "หนีฝั่งที่กระดานกำลังกดดัน" เสมอ — นี่คือความหมายเดียวกับ "เอียงตามแรง (α)" ในสูตร ไม่ได้ขัดกัน

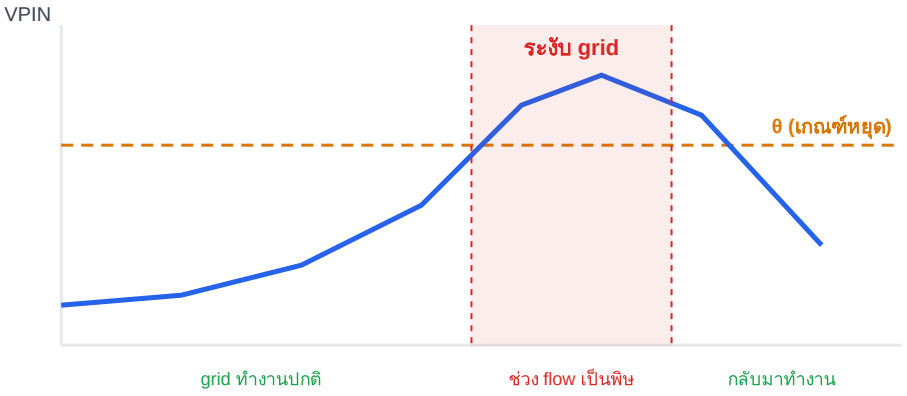


ภาพ 12.2 · เอียง grid ตาม imbalance — เลื่อนศูนย์กลางหนีฝั่งที่กระดานกดดัน

ชั้นที่ 3 — VPIN เป็นสวิตช์หยุด grid

จาก Part 8 เราเรียนว่า **VPIN วัด "ความเป็นพิษ" ของ flow** — เมื่อ flow ข้างเดียวรุนแรง (มัก = มีคนรู้ข้อมูล กำลังลากราคา) VPIN จะพุ่งขึ้น นั่นคือสัญญาณว่า "น้ำกำลังหลาก" และ grid กำลังจะถูกพัดขาด

กฎง่าย ๆ: **เมื่อ VPIN > θ (เกณฑ์) ให้ระงับการวางไม้ที่ "ทวนเทรนด์"** — เช่น ถ้า flow ไหลขายแรง อย่าวาง buy ดักรับเพิ่ม ปล่อยให้ราคาลงไปก่อน แล้วค่อยกลับมาวาง grid เต็มเมื่อ flow สงบ นี่คือเกราะกัน "เทรนด์เดียวล้างพอร์ต"



ภาพ 12.3 · VPIN หยุด grid — ตัดไม้ทวนเทรนด์ในช่วง flow เป็นพิษ

สูตรหลัก

```
// 1) spacing ไดนามิก — เอาค่ามากกว่าระหว่าง spread เฉลี่ย กับ volatility
Δ_grid = max( c1 · spread̄ , c2 · σ · √τ )
                c1,c2 = ตัวคูณกันชน (เช่น 1.5, 1.0)
                τ = horizon / holding time เป้าหมายของไม้ (กรอบเวลาที่คาดว่าจะถือก่อนปิด)
```

```
// 2) เลื่อนศูนย์กลาง grid ด้วย imbalance (ยืม reservation price)
P_center' = P_mid + α · OBI · Δ_grid   (α > 0 = ตามแรง, OBI<0 เลื่อนลง)

// 3) สวิตช์ VPIN
ถ้า VPIN > θ → ระงับไม้ที่ทวนทิศ flow (กันเทรนด์เดียวล้างพอร์ต)

// เงื่อนไขกำไรต่อไม้ (กับดักข้อ 1)
Δ_grid > 2 · fee_taker (หรือ 2·fee_maker ถ้าใช้ post-only) — ไม่งั้นยิ่งเด้งยิ่งจน
```

พารามิเตอร์	ตายตัว (grid ดิบ)	มองกระดาน (grid + flow)
spacing Δ_grid	คงที่ (เช่น 0.5%)	max(spread, σ√τ) — กว้างเมื่อผันผวน
ศูนย์กลาง	mid เสมอ	P_center' เอียงตาม OBI/OFI
ช่วง flow เป็นพิษ	วางต่อ → โดนลาก	VPIN>θ → ระงับไม้ทวนเทรนด์
fee	มักลืม → ขาดทุนเงียบ	บังคับ Δ_grid > 2×fee

📑 งานวิจัยรองรับ

การเอียง quote ตาม inventory และ signal มีรากจาก **Avellaneda-Stoikov (2008)** ที่ derive reservation price `r = s − q·γσ²(T−t)` — grid ที่เอียงด้วย OBI คือการนำหลักเดียวกันมาใช้กับ "แรงกระดาน" แทน inventory ส่วนการใช้ **VPIN (Easley-López de Prado-O'Hara 2012)** เป็นตัวตัดความเสี่ยง toxic flow ก็คือการ "ปิดร้านชั่วคราว" ตามที่ MM มืออาชีพทำจริง อย่างไรก็ดี ค่า θ และ α ต้อง **calibrate กับตลาด/timeframe เอง** ไม่มีเลขวิเศษตายตัว

🔗 เชื่อมโยงเล่มอื่น & ในซีรีส์

Part 12 มัด 3 เครื่องมือที่เรียนมารวมกัน: **OBI (Part 5)** เอียง grid, **OFI (Part 7)** ทำนายทิศแม่นขึ้น, **VPIN (Part 8)** เป็นเบรก และโครง MM ทั้งหมดมาจาก **Part 9–11** ในแง่คณิตศาสตร์ `Δ_grid=max(...)` และ `P_center'` คือ optimization ภายใต้ความผันผวน (เชื่อม **math**) ส่วนการคิด P&L ต่อไม้และ inventory ที่สะสมก็คือ payoff จริงในเล่ม **pm/payoff** — อย่าลืมว่าราคาเข้า-ออกคือ bid/ask ไม่ใช่ mid

✍️ แบบฝึก (ทำได้ด้วยข้อมูลคริปโตฟรี)

เลือกเหรียญหนึ่ง (เช่น ETH/USDT บน Binance) ดึงราคาย้อนหลัง 2 วัน: **วันที่ sideway** กับ **วันที่ trending** จำลอง grid 10 ชั้น spacing 0.4% แล้ว **หัก taker fee 0.04% ทุกไม้ทั้งขาเข้า-ออก** เทียบ P&L สองวัน — คุณจะเห็นว่า grid ดิบกำไรวัน sideway แต่ขาดทุนหนักวัน trending จากนั้นเพิ่มกฎ "ถ้า OBI < −0.3 หยุดวาง buy" แล้ววัดว่าวัน trending ขาดทุนน้อยลงไหม

ต่อ Part 13: Statistical Arbitrage ด้วย OFI — ทำนายราคาด้วยแรง flow →

อ่านกระดาน
Reading the Book

Part 13: Statistical Arbitrage ด้วย OFI — ทำนายราคาด้วยแรง flow

Order Flow & Microstructure สำหรับรายย่อย · เลนส์ execution ของซีรีส์

OFI มาก่อนราคา — แปลงให้นิ่ง ทำ regression แล้วต่อยอดเป็น cross-asset (BTC ดัน ETH)

Part 13 — Statistical Arbitrage ด้วย OFI

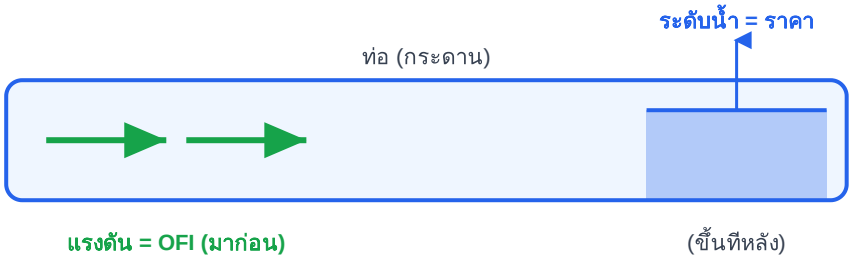
🎯 **อ่านจบ Part นี้คุณจะ**

- เข้าใจว่า **OFI (order flow imbalance) "มาก่อน" การเปลี่ยนแปลงราคา** — เป็น predictor ไม่ใช่ภาพสะท้อน
- รู้ว่าทำไมต้อง **แปลง OFI ให้ stationary (log/normalize)** ก่อนทำ regression ไม่งั้นได้ผลลวง (spurious)
- ทำ **lead-lag stat-arb (ราคาตาม flow)**: เข้าเทรดเมื่อราคาคาดการณ์จาก OFI เบนจากราคาจริงเกินต้นทุน
- ต่อยอดเป็น **cross-asset / cross-impact**: flow ของ BTC ทำนายราคา ETH ได้ (เมทริกซ์ γ)

เริ่มจากแรงดันน้ำในท่อ

นึกถึงน้ำในท่อประปา — ก่อนที่ **ระดับน้ำ** (ราคา) จะขยับ ต้องมี **แรงดัน** (flow) ดันมาก่อนเสมอ ถ้าคุณวัดได้แค่ระดับน้ำ คุณรู้ทีหลังเสมอ แต่ถ้าคุณวัด "แรงดัน" ได้ คุณรู้ **ก่อน** ว่าน้ำกำลังจะขึ้นหรือลง

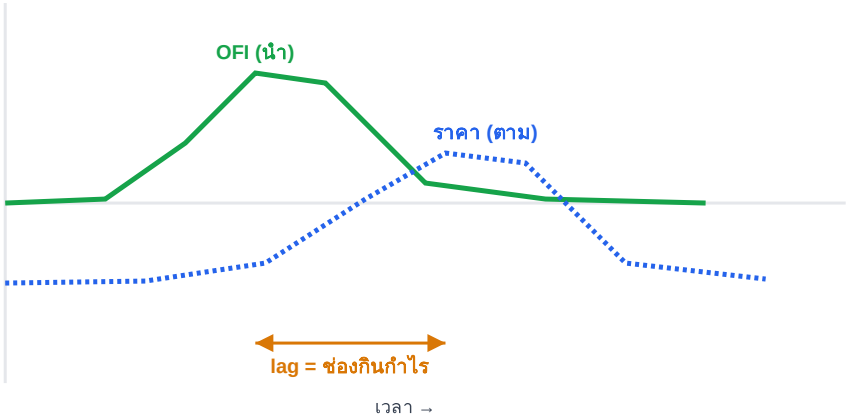
OFI คือเกจวัดแรงดันของกระดาน — มันรวมทั้ง limit ที่เติมเข้า, cancel ที่ถอนออก, และ market ที่กิน เป็นตัวเลขเดียวที่บอก "แรงสุทธิ" ที่กำลังดันราคา และเมื่อ **ท่อหลายเส้นต่อกัน** (BTC, ETH, SOL อยู่ในระบบเดียวกัน) แรงดันในท่อใหญ่ย่อมส่งผลถึงท่อเล็ก — นั่นคือ cross-asset



วัดแรงดันได้ = รู้ก่อนว่าราคาจะไปทางไหน

ชั้นที่ 1 — OFI นำหน้าราคา

จาก Part 7 เรารู้ว่า **ΔP ≈ λ·OFI** (Kyle's λ) — แต่สิ่งที่ทำให้มันเป็น stat-arb ได้คือ **ความล่าช้า (lag)**: OFI ที่ช่วงเวลานี้มักทำนาย ΔP ของช่วงถัดไปได้เล็กน้อย เพราะตลาดใช้เวลา "ย่อย" flow ไม่ทันที ถ้าราคา ณ ตอนนี้ **ยังไม่ขยับตามที่ OFI บอก** นั่นคือช่องว่าง lead-lag ที่กินได้ (เข้าตามทิศ flow ก่อนราคาจะตามมา)

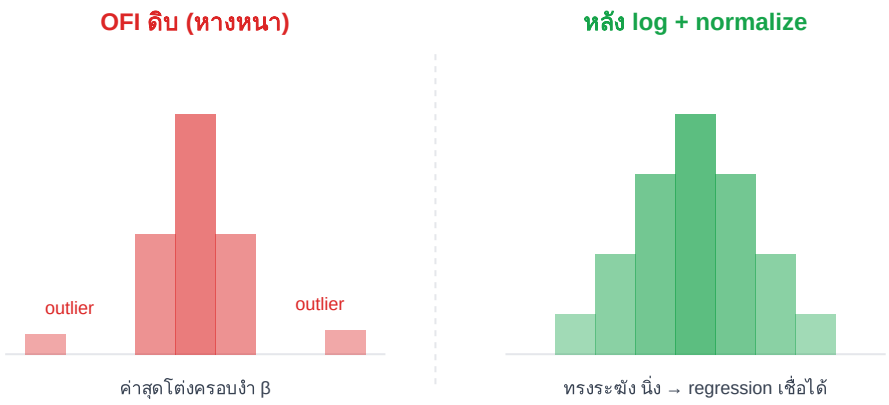


ภาพ 13.1 · OFI นำหน้าราคา — flow พุ่งก่อน ราคาขยับตามทีหลัง

ชั้นที่ 2 — แปลง OFI ให้นิ่งก่อน regression

นี่คือจุดที่รายย่อยส่วนใหญ่พลาด: **OFI ดิบมีหางหนา (fat tails) และไม่ stationary** — บางวินาที OFI กระโดดเป็นหมื่น บางวินาทีเกือบศูนย์ ถ้าเอาไป regress ตรง ๆ ค่า β จะถูกครอบงำด้วยค่าสุดโต่งไม่กี่ตัว ได้ความสัมพันธ์ลวง (spurious regression)

วิธีแก้: **(1) บีบหางด้วย log** `sign(OFI)·log(1+|OFI|)` และ **(2) normalize ด้วย depth** เพื่อให้เทียบข้ามช่วงเวลา/ข้ามเหรียญได้ — ผลคือกระจายตัวเข้าใกล้ปกติและนิ่งขึ้น regression จึงเชื่อถือได้



ภาพ 13.2 · stationarize ก่อน/หลัง — log บีบหางหนาให้เข้าใกล้ปกติ

ชั้นที่ 3 — cross-asset / cross-impact (BTC → ETH)

เหรียญใหญ่ในคริปโตเคลื่อนพร้อมกันสูง — flow ที่ดัน BTC มัก "ลาก" ETH ตามในเสี้ยววินาทีถัดมา (cross-impact) เราขยาย regression จากตัวแปรเดียวเป็น **เมทริกซ์**: ราคาของเหรียญ i ขึ้นกับ OFI ของตัวเอง *บวก* OFI ของเหรียญอื่น ๆ ถ่วงด้วยสัมประสิทธิ์ **γ_ij**

เมทริกซ์ cross-impact γ_ij

	ofi BTC	ofi ETH	ofi SOL
Δp BTC	0.42	0.08	0.03
Δp ETH	0.21	0.38	0.05
Δp SOL	0.18	0.12	0.34

นอกแนวทแยง = แรงข้ามเหรียญ (BTC→ETH = 0.21)

ภาพ 13.3 · เมทริกซ์ γ — flow ของ BTC ส่งผลถึงราคา ETH/SOL (ตัวเลขสาธิต ไม่ใช่ค่าจริง)

สูตรหลัก

```
// 1) แปลง OFI ให้นิ่ง (log + normalize ด้วย depth D)
ofi = sign(OFI) · log(1 + |OFI|) / D

// 2) cross-asset regression (เหรียญ i ขึ้นกับ flow ตัวเอง + ตัวอื่น)
Δp_i = β·ofi_i + Σ_j γ_ij · ofi_j + ε

// 3) สัญญาณ lead-lag / ราคาตาม flow (ราคาคาดการณ์ vs ราคาจริง)
Δp̂ = ราคาที่โมเดลทำนาย ; เข้าเทรดเมื่อ |Δp̂| > cost
   cost = fee×2 + spread + slippage  (edge ต้องชนะต้นทุนก่อน)
```

⚠️ **ความจริงรายย่อย — edge บางมาก ต้นทุนกินหมด**

stat-arb ด้วย OFI เป็น **กลยุทธ์ที่รายย่อยทำได้จริงที่สุดในกลุ่ม advanced** (จะเทียบให้เห็นชัดใน Part 14) เพราะใช้ข้อมูล L2 ฟรีและไม่ต้องแข่ง latency ระดับ HFT — **แต่** edge ต่อไม้เล็กมาก (มัก < 1 bps) ถ้า cost (fee×2 + spread + slippage) ไม่ต่ำกว่า edge คุณจะ "ทำนายถูกแต่ขาดทุน" และระวัง **spurious regression** จาก OFI ดิบ: ถ้าไม่ stationarize ก่อน R^2 ที่เห็นในอดีตจะหลอกคุณ พอรันจริงพังทันที

📑 งานวิจัยรองรับ

นิยาม OFI event-based มาจาก **Cont-Kukanov-Stoikov (2014)** ส่วนการขยายเป็น **multi-level + cross-impact** มาจาก **Cont-Cucuringu-Zhang (2023)** ซึ่งแสดงว่า OFI ข้ามหลักทรัพย์ช่วยอธิบายการเคลื่อนราคาได้เพิ่ม และมีงาน **HF Stat-Arb with Stationarized OFI** ที่ยืนยันว่าการแปลงให้ stationary สำคัญต่อความเสถียรของสัญญาณ *ข้อควรระวัง:* ค่า β, γ และ R^2 ขึ้นกับตลาด/timeframe และ **เสื่อมตามเวลา** ต้อง re-fit สม่ำเสมอ

🔗 เชื่อมโยงเล่มอื่น & math

Part 13 คือการต่อยอด **OFI/Kyle's λ (Part 7)** ตรง ๆ ในเชิงคณิตศาสตร์นี่คือ **multivariate linear regression** เต็มตัว (เชื่อม **math**: OLS, R^2, และเมทริกซ์ γ ก็คือ multi-output regression / สัมพันธ์กับ PCA ของ flow) การ stationarize ด้วย log เชื่อมกับบท log/returns ในเล่ม math ส่วนสัญญาณ `|Δp̂| > cost` คือการเปรียบ payoff ที่คาดหวังกับต้นทุนจริง — หลักเดียวกับการหา edge ใน **arb** และคิด P&L ใน **pm**

✍️ แบบฝึก (ทำได้ด้วยข้อมูลคริปโตฟรี)

ดึง L2 depth + trade stream ของ **ETH/USDT และ BTC/USDT** พร้อมกันสัก 1 ชั่วโมง คำนวณ OFI ทุก ๆ 1 วินาที แล้วทำ regression สองแบบเทียบกัน: **(ก)** `Δp(ETH) ~ ofi(ETH)` กับ **(ข)** `Δp(ETH) ~ ofi(ETH) + ofi(BTC)` ดูว่า R^2 เพิ่มขึ้นไหมเมื่อใส่ flow ของ BTC เข้าไป จากนั้นลองไม่ stationarize vs stationarize แล้วสังเกตว่า β เสถียรต่างกันแค่ไหน

ต่อ Part 14: Cross-Exchange / Latency / Triangular Arb — arb จริงทำได้แค่ไหน →

อ่านกระดาน
Reading the Book

Part 14: Cross-Exchange / Latency / Triangular Arb — arb จริงทำได้แค่ไหน

Order Flow & Microstructure สำหรับรายย่อย · เลนส์ execution ของซีรีส์

spread บนจอ ≠ กำไร — คิดต้นทุนครบวงจรด้วย effective price แล้วดูตารางความจริงรายย่อย

Part 14 — Cross-Exchange / Latency / Triangular Arb

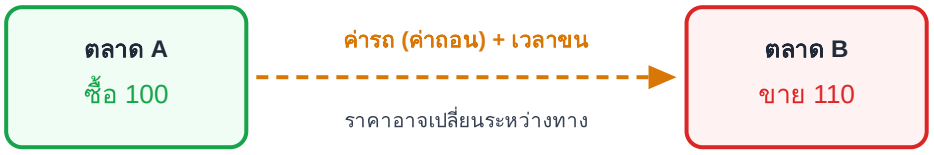
🎯 **อ่านจบ Part นี้คุณจะ**

- แยก arb 3 ชนิด: **cross-exchange** (สองตลาด), **latency** (แข่งความเร็ว), **triangular** (สามคู่เหรียญ)
- คิด **effective price / depth** — ราคาจริงเมื่อ "ไต่เล่ม" ไม่ใช่ best price บนจอ
- คิด **ต้นทุนครบวงจร**: fee + ค่าถอน + slippage + เวลาโอน — ตัวฆ่ากำไรตัวจริง
- เห็น **ตารางความจริงรายย่อย**: อันไหนทำได้ อันไหนอย่าแตะ และทำไม OFI stat-arb ถึงเหมาะที่สุด

เริ่มจากซื้อทุเรียนตลาด A ขายตลาด B

ทุเรียนที่ตลาด A ลูกละ 100 บาท ตลาด B รับซื้อ 110 บาท — กำไร 10 บาทใช่ไหม? **ยังก่อน** คุณต้องจ่าย **ค่ารถขนของ** (ค่าถอน/ค่าโอน), **เสียเวลาขน** (ระหว่างขนราคาอาจเปลี่ยน), และถ้าคุณซื้อหลายลูก **ลูกท้าย ๆ ราคาแพงขึ้น** (ไต่เล่ม) ส่วนตลาด B ก็รับซื้อถูกลงเรื่อย ๆ

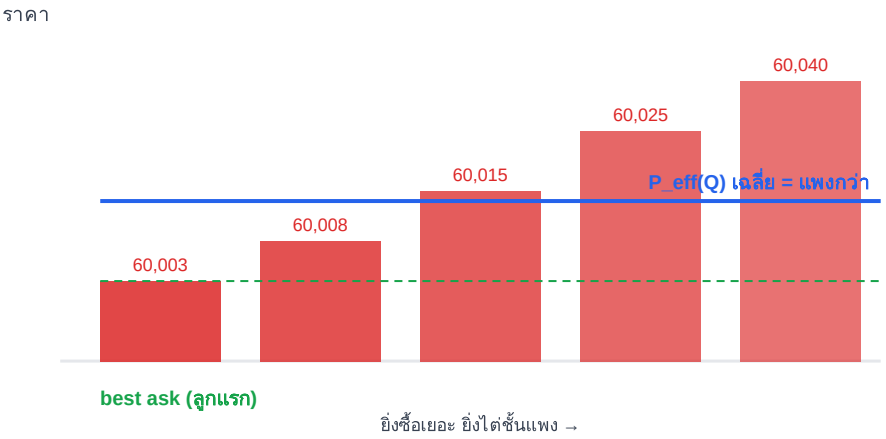
นี่คือหัวใจของ arb จริง: **ส่วนต่างราคาบนจอเป็นแค่ "ราคาตั้ง" ของลูกแรก** กำไรจริงต้องหักต้นทุนครบวงจรและคิดที่ effective price



กำไรจริง = 10 − ค่ารถ − slippage − fee

ชั้นที่ 1 — effective price: ราคาจริงเมื่อไต่เล่ม

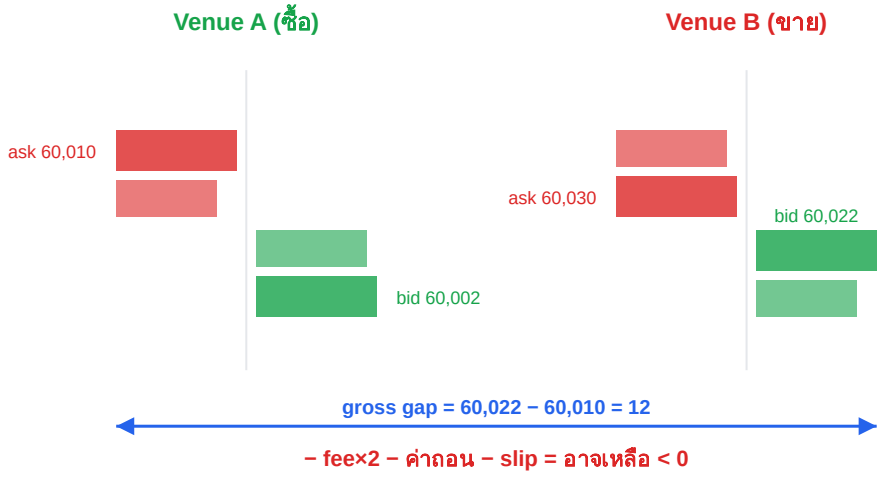
ถ้าคุณซื้อปริมาณ Q ที่มากกว่าของที่ best ask มีอยู่ คุณต้อง "กิน" ชั้นถัดไปที่แพงขึ้นเรื่อย ๆ (walk the book จาก Part 2) ราคาเฉลี่ยที่คุณได้จริงคือ **effective price** ซึ่งแย่กว่า best price เสมอ — และยิ่ง Q ใหญ่ ยิ่งแย่



ภาพ 14.1 · effective price ไต่เล่ม — ราคาเฉลี่ยจริงแพงกว่า best ask

ชั้นที่ 2 — cross-exchange: หัก fee สองฝั่งแล้วเหลืออะไร

cross-exchange arb คือซื้อที่ venue A (effective ask) ขายที่ venue B (effective bid) แต่กำไรบนจอมักหายหมดเมื่อหัก **fee ทั้งสองฝั่ง + ค่าถอน + slippage** และที่ร้ายกว่าคือ **เวลาโอนเหรียญข้าม chain** ที่ราคาอาจวิ่งหนีก่อนคุณขนของถึง



ภาพ 14.2 · cross-exchange 2 venue — gross gap สวย แต่หักต้นทุนแล้วมักติดลบ

ชั้นที่ 3 — triangular: วนสามคู่ในตลาดเดียว

triangular arb วนเงินผ่านสามคู่ในตลาดเดียวกัน เช่น USDT → BTC → ETH → USDT ถ้าผลคูณอัตราแลกเปลี่ยน **R > 1** (หลังหัก fee สามขา) ก็มีกำไร ข้อดีคือ **ไม่ต้องโอนข้าม venue** (ตัดค่าถอน/เวลาโอนออก) แต่ต้องเร็วมากเพราะช่องปิดในเสี้ยววินาที และ fee สามขากินหนัก

ผูกกับสูตรให้ชัด: ในวง USDT → BTC → ETH → USDT **ขา "ซื้อ" ใช้ 1/P** (จ่ายเงินอ้างอิงเพื่อได้เหรียญ) ส่วน **ขา "ขาย" ใช้ ×P** (ขายเหรียญกลับเป็นเงินอ้างอิง) — ขาที่ 1 ซื้อ BTC ด้วย USDT ใช้ **$1/P_1$** (P_1 = ราคา BTC/USDT) ขาที่ 2 ซื้อ ETH ด้วย BTC ใช้ **$1/P_2$** (P_2 = ราคา ETH/BTC) และขาที่ 3 ขาย ETH กลับเป็น USDT ใช้ **$\times P_3$** (P_3 = ราคา ETH/USDT) — จึงได้ $R = (1/P_1)\cdot(1/P_2)\cdot P_3$ ก่อนหัก fee

```
// effective price เมื่อไต่เล่ม (เฉลี่ยถ่วงปริมาณแต่ละชั้น)
P_eff(Q) = Σ p_k · q_k / Q   (q_k = ปริมาณที่กินในชั้น k)

// กำไรสุทธิ cross-exchange (ครบวงจร)
Π_net = (P_sell,B − P_buy,A) − fee − w_withdraw − slip
        P_buy,A = P_eff ฝั่งซื้อ ; P_sell,B = P_eff ฝั่งขาย

// triangular: มีกำไรเมื่อผลคูณอัตรา (หลัง fee 3 ขา) > 1
R = (1/P₁) · (1/P₂) · P₃ · (1−f)³ ; เข้าเมื่อ R > 1
```

ตารางความจริงรายย่อย — อันไหนทำได้จริง

นี่คือหัวใจของ Part นี้ — เราจะ **ซื่อสัตย์** ว่ารายย่อยทำ arb แต่ละชนิดได้จริงแค่ไหน:

ชนิด arb	รายย่อยทำได้?	ทำไม
Latency arb	❌ อย่าแตะ	แข่งความเร็วระดับไมโครวินาที — ต้อง colocation + โครงสร้าง HFT ที่รายย่อยสู้ไม่ได้ ไม่ว่าจะมองกระดานเก่งแค่ไหน
Cross-exchange (spot)	⚠️ ระวังมาก	ทำได้บนกระดาษ แต่ fee×2 + ค่าถอน + เวลาโอนข้าม chain มักกินกำไรหมด และเสี่ยงราคาวิ่งหนีระหว่างโอน
Triangular	⚠️ ระวัง	ไม่ต้องโอนข้าม venue (ดีกว่า) แต่ fee สามขากินหนัก ช่องปิดเร็ว ต้องเร็วกว่าบอตทั่วไป
OFI stat-arb (Part 13)	✅ เหมาะสุด	ใช้ L2 ฟรี ไม่แข่ง latency สุดขีด เป็น "ยกระดับกลยุทธ์เดิม" ด้วยกระดาน — แต่ edge บาง ต้องคุม cost เข้ม

> ⚠️ **ความจริงรายย่อย — spread บนจอ ≠ กำไร**
>
> เห็น "BTC ที่ Binance ถูกกว่า Bybit 0.1%" แล้วคิดว่าได้เงินฟรี — **นี่คือกับดักคลาสสิก** ตัวเลขนั้นคือ best price ของลูกแรก ไม่ใช่ effective price และยังไม่หัก fee สองฝั่ง ค่าถอน เวลาโอน กว่าคุณจะขนของถึง ราคามักปิดช่องไปแล้ว **latency arb โดยเฉพาะ — อย่าลองเด็ดขาด** คุณกำลังแข่งกับบอตที่เร็วกว่าคุณล้านเท่า สิ่งที่รายย่อยทำได้จริงคือ OFI stat-arb ที่ใช้ "ความเข้าใจกระดาน" ไม่ใช่ "ความเร็วของเครื่อง"

📑 งานวิจัยรองรับ

แนวคิด effective price / market impact ของการไต่เล่มมาจากตระกูล market impact models (Kyle 1985 และงานต่อยอด) ส่วนการคิดต้นทุนการเทรดครบวงจร (transaction cost analysis) เป็นมาตรฐานในงาน execution ทุกแขนง สำหรับคริปโต **โครงสร้าง fee/rebate และค่าถอนของ Binance/Bybit เปลี่ยนตามโปรโมชัน** — ก่อนคำนวณจริงให้เช็คหน้า fee schedule ปัจจุบันเสมอ เพราะมันคือตัวแปรที่ตัดสินว่ากำไรเป็นบวกหรือลบ

🔗 เชื่อมโยงเล่มอื่น — สะพานสู่เล่ม arb

Part 14 คือ **สะพานเชื่อมตรงสู่เล่ม arb**: เล่ม arb หา "โอกาส" ส่วนตรงไหนที่ effective price/depth บอกว่า **ต้นทุนจริงของ arb เท่าไหร่** คือสิ่งที่เล่มนี้เติมให้ `P_eff(Q)` และ `Π_net` คือเครื่องมือที่ทำให้ตัวเลขในเล่ม arb "เป็นจริงในกระดาน" ในเชิง **math** นี่คือผลรวมถ่วงน้ำหนักและเงื่อนไข R>1 ส่วนการคิดกำไรขาดทุนหลัง cost ทั้งหมดก็คือ payoff จริงในเล่ม **pm/payoff**

✍️ แบบฝึก (ทำได้ด้วยข้อมูลคริปโตฟรี)

ดึง order book L2 ของเหรียญเดียวกันจาก **สอง exchange** พร้อมกัน (เช่น Binance vs Bybit) คำนวณ **P_eff ฝั่งซื้อที่ A และ P_eff ฝั่งขายที่ B** สำหรับขนาด $1,000 แล้วหัก taker fee สองฝั่ง + ค่าถอนโดยประมาณ นับดูว่าในช่วง 1 ชั่วโมงมีกี่วินาทีที่ **Π_net > 0 จริง** — คุณจะพบว่าน้อยมาก และส่วนใหญ่ปิดเร็วเกินกว่าจะคว้าทัน นี่คือบทพิสูจน์ว่าทำไมตารางความจริงถึงให้ cross-exchange แค่ ⚠️

ต่อ Part โบนัส: เครื่องจักรอ่านกระดาน — DeepLOB, Transformers และทำไม ML คือปลายทาง →

อ่านกระดาน
Reading the Book

Part โบนัส: เครื่องจักรอ่านกระดาน — DeepLOB, Transformers

Order Flow & Microstructure สำหรับรายย่อย · เลนส์ execution ของซีรีส์

ML คือปลายทาง ไม่ใช่จุดเริ่ม — "feature ดี > โมเดลลึก" และอย่าทุ่ม architecture แทนข้อมูล

Part โบนัส — เครื่องจักรอ่านกระดาน

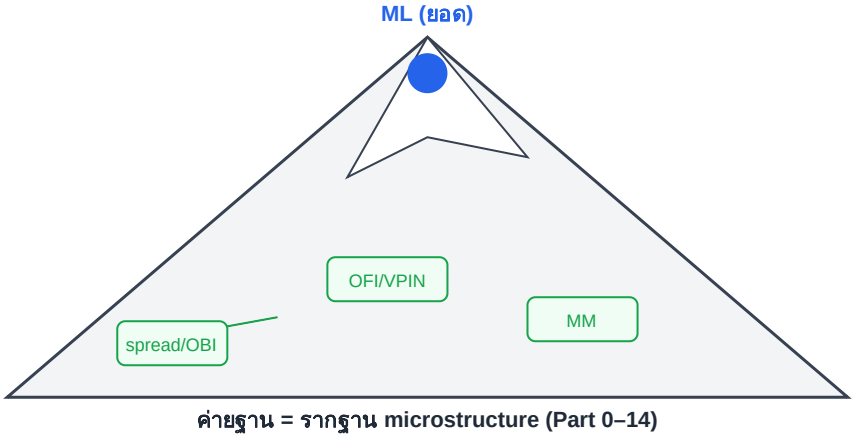
🎯 **อ่านจบ Part นี้คุณจะ**

- เข้าใจ pipeline ของโมเดล deep learning อ่านกระดาน: **LOB → CNN → LSTM → ทำนายทิศ mid** (DeepLOB)
- รู้จัก **Transformer สำหรับ LOB** (TLOB/LiT) และจุดที่มันต่างจาก CNN+LSTM
- ซึมหลักสำคัญ: **"feature ดี > โมเดลลึก"** — OFI/imbalance ที่ออกแบบดีชนะ network ใหญ่ที่ป้อนข้อมูลดิบ
- เข้าใจว่า **ML คือปลายทาง ไม่ใช่จุดเริ่ม** — ต้องมีรากฐาน microstructure ทั้ง 14 Part ก่อน

เริ่มจากยอดภูเขา ไม่ใช่ค่ายฐาน

คนที่พิชิตยอดเอเวอเรสต์ไม่ได้เริ่มจากการกระโดดขึ้นยอดเลย — เขาเริ่มจาก **ค่ายฐาน** ปรับตัวกับความสูง ฝึกหายใจ แล้วค่อยไต่ขึ้นทีละแคมป์ **ML สำหรับอ่านกระดานคือ "ยอดภูเขา"** — สวยงาม แต่ถ้าคุณไม่มีค่ายฐาน (เข้าใจ spread, OBI, OFI, VPIN, market making) คุณจะไม่รู้ด้วยซ้ำว่าโมเดลมันเรียนรู้อะไรหรือพังตรงไหน

อีกอุปมา: **สายตาเทรดเดอร์ที่ฝึกดูกระดานมาเป็นล้านครั้ง** — neural network ก็คือการจำลองสายตานั้นด้วยข้อมูลมหาศาล แต่ "สายตา" จะแม่นได้ต้องป้อน **สิ่งที่มีความหมาย** ให้ดู ไม่ใช่ตัวเลขดิบมั่ว ๆ

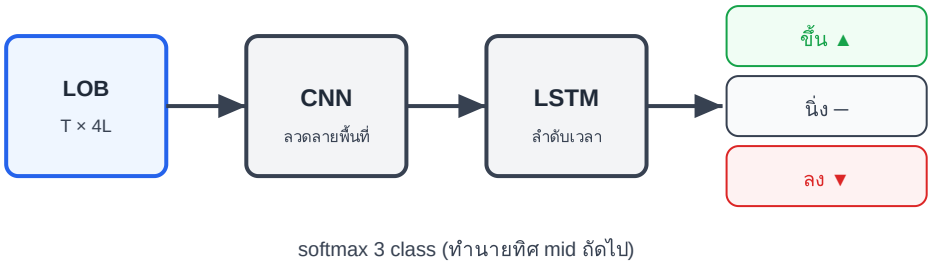


ภาพ B.1 · ML คือยอดภูเขา — ต้องมีค่ายฐาน microstructure ก่อนถึงจะไต่ขึ้นได้

ชั้นที่ 1 — pipeline DeepLOB (CNN + LSTM)

DeepLOB รับ **snapshot ของกระดานหลายชั้น ต่อกันเป็นลำดับเวลา** (เมทริกซ์ขนาด T × 4L: แต่ละชั้น L มีราคา/ปริมาณ ของ bid และ ask) แล้วส่งผ่าน:

- **CNN** — กวาดหา "ลวดลายเชิงพื้นที่" ในกระดาน (เช่น รูปทรง imbalance ข้ามชั้น) แทนการดูตัวเลขทีละตัว
- **LSTM** — จับ "ลำดับเวลา" ว่าลวดลายเปลี่ยนยังไงในไม่กี่วินาที
- **softmax 3 class** — ทำนายว่า mid ช่วงถัดไปจะ ขึ้น / นิ่ง / ลง



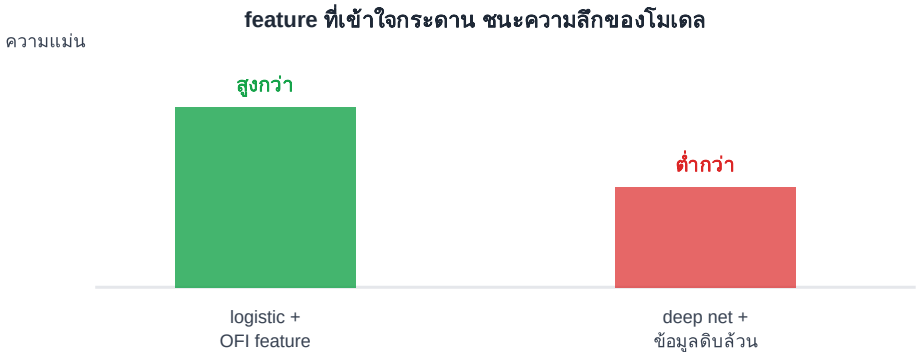
ภาพ B.2 · pipeline LOB → CNN → LSTM → ทำนาย 3 class

ชั้นที่ 2 — Transformer สำหรับ LOB (TLOB/LiT)

Transformer ใช้ **attention** แทน convolution + recurrence — มันเรียนได้เองว่า "ชั้นไหน/เวลาไหนของกระดาน" สำคัญต่อการทำนาย โดยไม่ต้องไล่ลำดับทีละสเต็ปแบบ LSTM งานอย่าง **TLOB** และ **LiT** แสดงว่าสถาปัตยกรรมนี้แข่งขันได้และบางทีดีกว่าในบางชุดข้อมูล — **แต่** มันกินข้อมูลและ compute หนักกว่ามาก และ *ไม่ใช่* กระสุนเงิน

ชั้นที่ 3 — feature ดี > โมเดลลึก (หัวใจของ Part นี้)

นี่คือบทเรียนที่แพงที่สุดและคนข้ามบ่อยที่สุด: **โมเดลลึกที่ป้อนข้อมูลดิบ มักแพ้โมเดลตื้นที่ป้อน feature ที่ออกแบบดี** เพราะ OFI, imbalance, microprice ที่คุณเรียนมาทั้งเล่มคือ "ความรู้ microstructure" ที่ย่อยมาให้โมเดลแล้ว — network ไม่ต้องเสียพลังไป "ค้นพบใหม่" จากศูนย์



ภาพ B.3 · feature ดี > โมเดลลึก — ความรู้ microstructure ที่ย่อยมาแล้วมีค่ากว่า depth

สูตรหลัก

```
// อินพุต: ลำดับ snapshot กระดาน T ช่วงเวลา × 4L ฟีเจอร์ต่อชั้น
X ∈ R^{T × 4L}  (p_bid,v_bid,p_ask,v_ask ของ L ชั้น)

// label: ทิศ mid ในอนาคต แบ่ง 3 class
y ∈ {ขึ้น, นิ่ง, ลง} (ใช้ threshold บนการเปลี่ยนของ mid)

// objective: ลด cross-entropy ของ softmax
min CrossEntropy( softmax(f(X)) , y )
   f = CNN+LSTM (DeepLOB) หรือ Transformer (TLOB/LiT)
```

⚠️ ความจริงรายย่อย — กับดักของ ML กับกระดาน

(1) ทุ่ม architecture แทน feature/ข้อมูล — เพิ่มเลเยอร์ไม่ช่วย ถ้า feature ห่วยหรือข้อมูลน้อย เริ่มที่ feature engineering ด้วยความรู้ microstructure ก่อนเสมอ **(2) ไม่กัน overfit** — LOB มี noise สูงและ regime เปลี่ยน โมเดลที่ "แม่นในอดีต" มักพังกับข้อมูลใหม่ ต้องแยก train/validation/test ตามเวลาอย่างเคร่งครัด และระวัง look-ahead bias **(3) คิดว่า ML แทนความเข้าใจได้** — ผิด ML ขยายความเข้าใจที่คุณมี ถ้าคุณไม่เข้าใจ OFI/imbalance คุณจะ debug โมเดลที่พังไม่เป็น **ML คือปลายทาง ไม่ใช่จุดเริ่ม**

📑 งานวิจัยรองรับ

DeepLOB (Zhang et al. 2018, arXiv 1808.03668) เป็นงานวางมาตรฐาน CNN+LSTM อ่าน LOB ทำนายทิศ mid ส่วน **TLOB (arXiv 2502.15757)** และ **LiT** นำ Transformer/attention มาใช้ และมีงานสำรวจ microstructure ในคริปโต LOB โดยเฉพาะ (**arXiv 2506.05764**) *ข้อควรระวัง:* ผลความแม่นในเปเปอร์มัก backtest บนข้อมูลเฉพาะตลาด/ช่วงเวลา — อย่าคาดหวังตัวเลขเดียวกันกับคริปโตที่ regime เปลี่ยนเร็ว และต้อง validate ด้วยข้อมูลของคุณเอง

🔗 เชื่อมโยงเล่มอื่น & ปิดซีรีส์

Part โบนัสมัดทั้งซีรีส์เข้าด้วยกัน: **feature** ที่ป้อนโมเดลคือ OFI (Part 7), OBI (Part 5), microprice (Part 11), VPIN (Part 8) ที่เรียนมาทั้งหมด ในเชิง **math** นี่คือ softmax/cross-entropy, logistic regression, และ optimization ที่ต่อยอดจากบท regression/PCA โดยตรง บทเรียนสุดท้ายของทั้งเล่ม "อ่านกระดาน" คือ: **order book เป็นเครื่องมือยกระดับกลยุทธ์เดิม** ไม่ใช่ระบบเบ็ดเสร็จ — และ ML ก็เป็นเครื่องมือยกระดับ "ความเข้าใจกระดาน" ของคุณ ไม่ใช่ตัวแทนของมัน

✍️ แบบฝึก (ทำได้ด้วยข้อมูลคริปโตฟรี)

เก็บข้อมูล L2 + trade ของ BTC/USDT สัก 2–3 ชั่วโมง สร้าง feature **OFI** (จาก Part 7) แล้วเทรน **logistic regression** ทำนายทิศ mid ถัดไป 3 class เทียบกับ baseline ง่าย ๆ คือ "เดาตาม sign(OFI)" ดูว่าโมเดลที่เรียนรู้ค่าน้ำหนักชนะการเดาตรง ๆ ไหม จากนั้น (ถ้าอยากท้าทาย) ลองป้อนข้อมูลกระดานดิบเข้า network เล็ก ๆ แล้วเทียบ — คุณจะได้สัมผัสด้วยตัวเองว่า "feature ดี > โมเดลลึก" จริงแค่ไหน

— จบซีรีส์ "อ่านกระดาน" · ขอให้อ่านกระดานออก และเทรดอย่างมีต้นทุนที่รู้จริง —