STATISTICAL ARBITRAGE · บทที่ 19

Cost Analysis & Edge: คำนวณ Net Edge ก่อนเทรด

Gross spread − fee − bid/ask − slippage − funding − legging buffer = Net edge

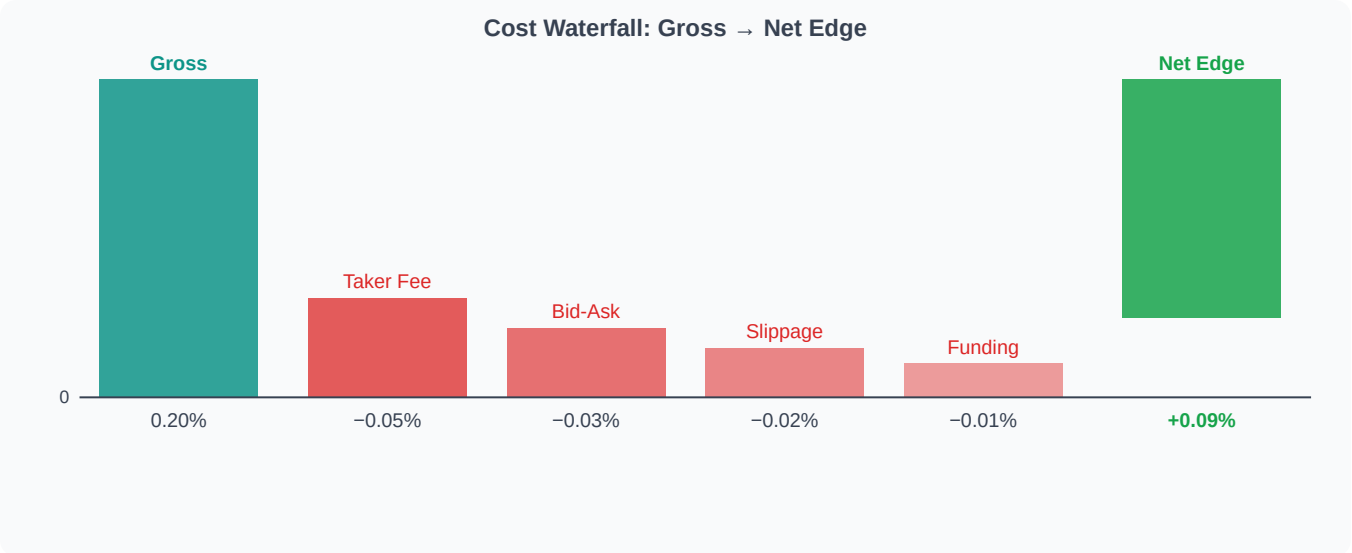
ถ้า net edge ≤ 0 → ไม่เทรด ไม่ว่า signal จะดูดีแค่ไหน

แก่นของบทนี้ — อ่านก่อน

Stat arb มีต้นทุนซ่อนอยู่หลายชั้น — signal ที่ดูมี edge 0.20% อาจมี net edge ลบหลัง cost หักออกทั้งหมด **คำนวณก่อนเสมอ — ห้าม assume**

```
Net Edge = Gross − Σ(fees + spread + slippage + funding + legging)
```

19.1 Edge Waterfall Diagram



Waterfall chart: แต่ละ cost component ลดลงจาก gross spread ของ 0.20% — เหลือ net edge เพียง 0.09%

19.2 Fee Structure

Exchange	Maker Fee	Taker Fee	หมายเหตุ
Bybit Perp	−0.025% (rebate)	+0.055%	VIP levels ลด taker ได้
Lighter Perp	−0.010%	+0.030%	clob-based, faster fills
Round trip cost (2 takers)			(0.055+0.030)×2 = 0.17%
Round trip cost (2 makers)			rebate: −0.07% (receive)

Maker vs Taker Strategy

ถ้า latency ไม่ critical → ใช้ limit order (maker) ทั้งคู่ — ลด cost จาก −0.17% เป็น −0.03% ต่อ round trip แต่ risk partial fill และ leg mismatch เพิ่มขึ้น

19.3 Bid-Ask Spread Cost

```
Cost per leg = (Ask − Bid) / 2 = half-spread สำหรับ 2 legs round trip: Total bid-
ask cost = half-spread_A + half-spread_B ตัวอย่าง Bybit BTC: Bid = 65,000, Ask =
65,002 Half-spread = $1 = 1/65,000 = 0.0015% Lighter BTC: Bid = 64,998, Ask =
65,001 Half-spread = $1.50 = 0.0023% Total: 0.0015% + 0.0023% = 0.0038% ต่อ
direction Round trip (entry + exit): 0.0076% ≈ 0.008%
```

19.4 Slippage Model

Market impact (slippage) เพิ่มขึ้นตาม size ที่เทียบกับ average daily volume:

```
Slippage ≈ σ_daily × sqrt(size / ADV) โดย: σ_daily = daily volatility ของ asset
size = ขนาด order (notional) ADV = average daily volume (notional) ตัวอย่าง: σ_daily
= 0.02 (2%) size = $100,000 ADV = $500,000,000 (Bybit BTC perp ~$500M/day)
Slippage = 0.02 × sqrt(100,000/500,000,000) = 0.02 × 0.014 = 0.00028 = 0.028%
```

19.5 Funding Cost

```
Funding cost ต่อ 8h = |funding_rate| × notional Total funding cost = funding_rate × hold_periods × notional ตัวอย่าง: Funding rate = 0.01% ต่อ 8h (positive → longs จ่าย) Hold time = 2 funding periods = 16h Notional = $100,000 Cost (ถ้า short perp) = +0.01% × 2 × $100,000 = +$20 (รับ) Cost (ถ้า long perp) = −0.01% × 2 × $100,000 = −$20 (จ่าย)
```

Funding Direction ขึ้นกับ Position

ถ้า basis > 0 และ short perp → รับ funding (ลด cost)

ถ้า basis > 0 และ long perp → จ่าย funding (เพิ่ม cost)

ต้องคำนวณรวมในทุก scenario ก่อน decide trade direction

19.6 Legging Buffer

```
Legging risk = ความเสี่ยงจากช่วงเวลาระหว่าง fill leg A และ fill leg B ในช่วงนั้น position
เป็น single-leg → exposed ต่อ market movement Legging buffer estimate: buffer =
σ_spread × sqrt(avg_fill_latency_s / seconds_per_period) ตัวอย่าง: σ_spread = 0.05%
per period avg_fill_latency = 0.5 วินาที seconds_per_period = 60 วินาที (1 นาที)
buffer = 0.05% × sqrt(0.5/60) = 0.05% × 0.091 = 0.0045%
```

19.7 Break-even Analysis

```
Minimum σ_stat เพื่อให้ trade คุ้มค่า: Break-even σ = total_cost / (z_entry − z_exit)
ตัวอย่าง: total_cost = 0.11% z_entry = 2.0, z_exit = 0.5 σ_stat = z × σ_spread →
spread ที่ entry = 2.0 × σ_spread Break-even: 0.11% / (2.0 − 0.5) = 0.073% → ต้องการ
σ_spread ≥ 0.073% เพื่อให้ entry at z=2.0 cover cost → ถ้า σ_spread = 0.05% → entry
spread = 0.10%, net edge = −0.01% (ไม่ trade!)
```

19.8 Pseudo-code compute_edge()

```
def compute_edge(spread_at_entry_pct, sigma_spread_pct, entry_z, exit_z,
fee_a=0.00055, fee_b=0.00030, half_spread_a=0.00002, half_spread_b=0.00003,
size=100_000, adv=500_000_000, sigma_daily=0.02, funding_per_8h=0.0001,
hold_periods=1, avg_fill_latency_s=0.5, seconds_per_period=60): gross =
spread_at_entry_pct / 100.0 # convert to decimal # Fee cost (round trip, taker
both sides) fee_cost = (fee_a + fee_b) * 2 # Bid-ask cost (round trip) ba_cost =
(half_spread_a + half_spread_b) * 2 # Slippage cost slip_cost = sigma_daily *
sqrt(size / adv) * 2 # entry + exit # Funding cost (sign depends on direction,
here assume cost) fund_cost = funding_per_8h * hold_periods # Legging buffer
leg_buffer = (sigma_spread_pct/100) * sqrt(avg_fill_latency_s/seconds_per_period)
total_cost = fee_cost + ba_cost + slip_cost + fund_cost + leg_buffer net_edge =
gross - total_cost breakeven_sigma = total_cost / (entry_z - exit_z) * 100 # as
percent return { "gross_pct": gross * 100, "total_cost_pct": total_cost * 100,
"net_edge_pct": net_edge * 100, "breakeven_sigma_pct": breakeven_sigma, "viable":
net_edge > 0 }
```

19.9 Running Example — Bybit↔Lighter Cost Breakdown

🔁 **Running Example: BTC Spread Trade Cost Analysis**

Component	Calculation	Cost (bps)
Gross spread at entry (z=2.0, σ=7.5bps)	2.0 × 7.5bps	+15.0 bps
Taker fee Bybit (0.055% × 2)	round trip	−1.1 bps
Taker fee Lighter (0.030% × 2)	round trip	−0.6 bps
Bid-ask spread (both venues)	2 × half-spread	−0.8 bps
Slippage ($100k on $500M ADV)	σ×sqrt(size/ADV)	−0.3 bps
Funding (1 period, 0.01%/8h)	hold 8h	−1.0 bps
Legging buffer (0.5s latency)	σ×sqrt(0.5/60)	−0.5 bps
Net Edge		**+10.7 bps (0.107%)**

สรุป: Gross 15bps แต่ net เพียง 10.7bps — cost กินไป 29% ของ gross edge

Break-even σ: total cost 4.3bps / (2.0 − 0.5) = 2.9bps → ต้องการ σ_spread ≥ 2.9bps เพื่อ trade คุ้มค่า (σ จริง = 7.5bps ✅)

19.10 แบบฝึกหัด

แบบฝึกหัดบทที่ 19

▶ ข้อ 1: ถ้า σ_spread = 3bps และ total cost = 4bps — ควรเทรดที่ z=2.0 ไหม?

▶ ข้อ 2: วิธีลด fee cost จาก 1.7bps เป็น 0.5bps — ทำได้อย่างไร?

▶ ข้อ 3: ถ้าต้องการ minimum net edge 5bps — คำนวณ minimum z_entry ที่ต้องใช้