Statistical Arbitrage
Synthetic Process & Epsilon Arbitrage

Part I: รากฐาน — Synthetic Process · Stochastic Calculus · OU Process

บทที่ 1–3: สร้าง process ใหม่จาก 2 assets · อัตราส่วนที่ถูกต้อง · พลวัต mean-reversion

ตัวอย่างหลัก: Bybit BTCUSDT Perpetual ↔ Lighter BTCUSDT Perpetual

บทที่ 1: Synthetic Process คืออะไร?

ก่อนจะเรียน statistical arbitrage ให้เข้าใจก่อนว่า เราไม่ได้ trade *ราคาสินทรัพย์* — เราสร้าง **process ใหม่ที่ไม่มีอยู่ในตลาด** แล้ว trade process นั้นแทน

1.1 ปัญหาของการ trade single asset

ทำไม random walk ถึงอันตราย

ถ้าราคา BTC เดิน Brownian motion: $P(T) = P(0) + \mu T + \sigma\sqrt{T} \cdot Z$

ไม่มีหลักประกันว่าราคาจะ "กลับ" มาที่ระดับใดเลย หากคุณ short ที่ $65,000 แล้วราคาวิ่งไป $100,000 — ความเสียหายไม่มีขอบเขต

การทำ stat arb ต้องการ process ที่มี **แรงดึงกลับ (mean-reversion)** — ยิ่งห่างจากค่ากลางมาก แรงดึงยิ่งแรง

1.2 Idea: สร้าง Process ใหม่จากการผสม 2 สินทรัพย์

สมมติมีสินทรัพย์ A และ B ที่ราคาเดินตาม random walk ทั้งคู่ แต่ทั้งสองมีความสัมพันธ์กัน (cointegrated) เราสามารถสร้าง:

SYNTHETIC PROCESS — นิยามพื้นฐาน

$$\varepsilon = F(A) - \beta \cdot F(B)$$

โดยที่:

- **F(·)** คือ transformation เช่น log price, basis, implied volatility
- **β** คือ *อัตราส่วนที่เหมาะสม* ที่ทำให้ ε มีคุณสมบัติ mean-reverting
- **ε** คือ spread หรือ "epsilon" — process ใหม่ที่เราจะ trade

สิ่งสำคัญที่สุด: β ≠ 1 เสมอไป

ถ้าตั้ง β=1 แล้ว trade 1:1 (Long 1 หน่วย A, Short 1 หน่วย B) — ε อาจยังเป็น random walk อยู่ก็ได้ ขึ้นอยู่กับสเกลราคา ความผันผวน และหน่วยของสินทรัพย์

β ที่ถูกต้องคือ β ที่ทำให้ ε **stationary** — กลับหาค่ากลางในระยะยาว

1.3 β คือ Entry Ratio ไม่ใช่แค่ Slope

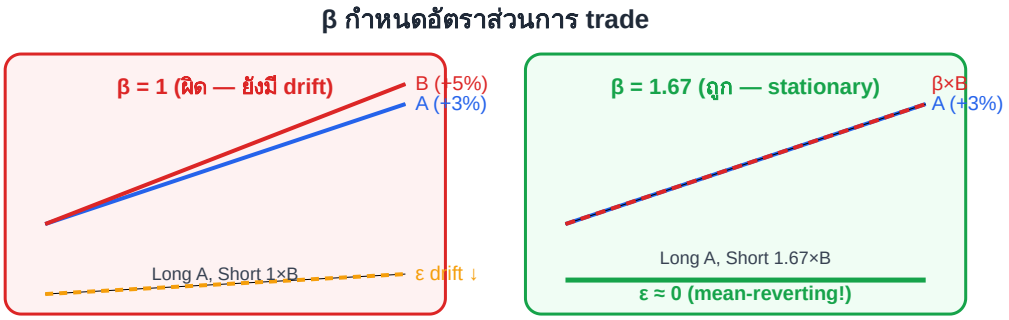
นี่คือจุดที่หลายคนเข้าใจผิด: β ไม่ใช่แค่ตัวเลขในสูตร — มันกำหนดว่า **คุณจะ trade ในอัตราส่วนเท่าไร**

ค่า β	การ trade จริง	ความหมาย
β = 1.0	Long 1 unit A, Short 1.0 unit B	อัตราส่วน 1:1 (กรณีพิเศษ)
β = 1.5	Long 1 unit A, Short 1.5 unit B	B มีความผันผวนน้อยกว่า ต้อง short มากกว่า
β = 0.7	Long 1 unit A, Short 0.7 unit B	B มีความผันผวนมากกว่า ต้อง short น้อยกว่า
β = 2.0	Long 1 unit A, Short 2.0 unit B	A sensitive กว่า B สองเท่า

Intuition: ทำไมต้องปรับอัตราส่วน?

สมมติ A ขึ้น 1% ทุกครั้งที่ B ขึ้น 2%
ถ้า trade 1:1 → Long A ขึ้น 1%, Short B ลง 2% → ε ขึ้นทุกครั้ง (drift!)
ถ้า trade 1:0.5 → Long A ขึ้น 1%, Short 0.5×B ลง 0.5×2%=1% → ε ≈ 0 ✓

β เป็นตัวที่ทำให้ drift ของทั้งสองฝั่ง "หักล้างกัน" พอดี



β = 3%/5% = 0.6 → ใช้ β=0.6 ไม่ใช่ β=1

Entry Position ที่ถูกต้อง

Long: N_A หน่วยของ A

Short: β × N_A หน่วยของ B

โดยที่ β หาจากข้อมูลย้อนหลัง (ดู Ch.4)

1.4 ทำไม β ≠ 1 ถึงสำคัญในทางปฏิบัติ

ลองนึกถึงเหตุผลที่ราคา A และ B อาจเคลื่อนไหวไม่เท่ากัน:

- **สเกลราคาต่างกัน:** A = $50,000/BTC, B = $50,100/BTC แต่ funding rate ต่างกัน → sensitivity ต่อ market ต่าง
- **Liquidity ต่างกัน:** B มี bid/ask กว้างกว่า → ราคาขยับน้อยกว่าต่อ order เดียวกัน
- **Contract size ต่างกัน:** Bybit BTCUSDT perpetual vs Lighter BTCUSDT — contract specification อาจต่างกัน
- **Beta ต่อ market:** A อาจ track BTC spot ได้ดีกว่า B → ความสัมพันธ์กับ underlying ต่าง

ตัวอย่างหลัก: Bybit ↔ Lighter BTC Perpetual

ตลอดเล่มนี้จะใช้คู่นี้เป็น running example:

- **A** = Bybit BTCUSDT Perpetual (price ≈ $65,000, funding rate เปลี่ยน 3×/วัน)
- **B** = Lighter BTCUSDT Perpetual (price ≈ $65,000, funding rate ต่าง)
- **F(·)** = log price: log(P_A) − β·log(P_B)

แม้ราคาทั้งสองจะใกล้เคียงกัน แต่ β อาจไม่เท่ากับ 1 เสมอ เพราะ:

- Funding rate mechanism ของสองตลาดต่างกัน → ราคา perpetual drift ต่างกันในช่วงสั้น
- Execution delay ต่างกัน → ราคาที่ observed ไม่ synchronous พอดี

ในตัวอย่างจริง สมมติ β = 1.003 (ไม่ใช่ 1 พอดี) → ถ้า long 100,000 USDT ใน Bybit ต้อง short 100,300 USDT ใน Lighter

1.5 รูปแบบ Synthetic Process ที่พบบ่อย

ประเภท	สูตร ε	ตัวอย่าง
Log Price Spread	log(P_A) − β·log(P_B)	Bybit perp vs Lighter perp
Perpetual Basis	P_perp − P_spot	BTC perp vs BTC spot index
Calendar Basis	P_front − β·P_back	BTC June futures vs September futures
Cross-venue Basis	P_Binance − β·P_Bybit	Same asset คนละตลาด
Options Parity	Call − Put − (F−K)e^{−rT}	Put-call parity deviation

1.6 สูตรทั่วไป: Multi-leg Synthetic Process

กรณีทั่วไปที่ผสมมากกว่า 2 สินทรัพย์:

MULTI-LEG SYNTHETIC PROCESS

$$\varepsilon_t = \Sigma_i \; w_i \cdot F_i(X_i(t))$$

โดยที่ w_i คือ weight (positive = long, negative = short) และต้องหาชุด w_i ที่ทำให้ ε stationary สำหรับ 2 legs: $w_1=1$, $w_2=-\beta$ ซึ่งเป็นรูปแบบที่เราจะโฟกัสตลอดเล่มนี้

ตรวจสอบความเข้าใจ 1.1

Bybit BTC perp ราคา $64,850 ขึ้น 2% ในชั่วโมงที่แล้ว Lighter BTC perp ราคา $64,900 ขึ้น 1.8% β ที่คำนวณได้ (จาก OLS บน log returns) = 1.05

ถาม: ถ้าจะ trade spread ด้วย notional $200,000 ใน Bybit ต้อง short Lighter เท่าไร?

▶ ดูคำตอบ

บทที่ 2: Stochastic Calculus พื้นฐาน

บทนี้ให้ความรู้ที่จำเป็นในการอ่านสมการพลวัตของ spread ในบทถัดๆ ไป เรียนแบบ bottom-up: Brownian Motion → Itô's Lemma → SDE

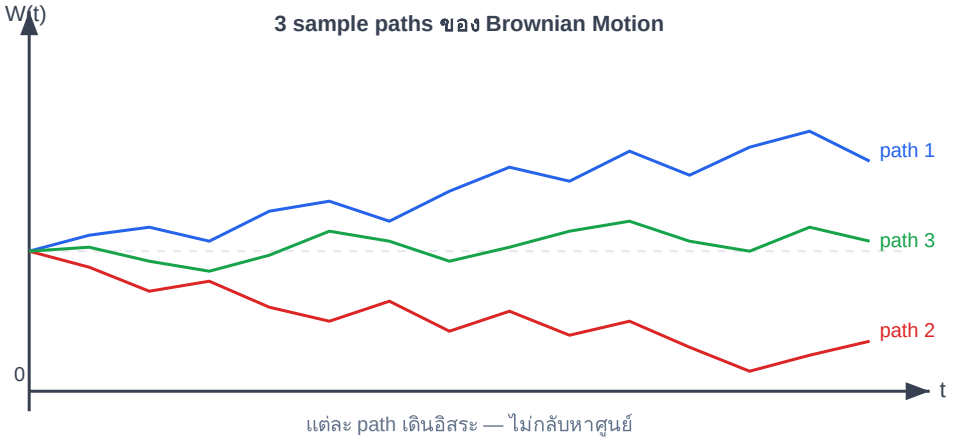
2.1 Brownian Motion (Wiener Process)

นิยาม: Standard Brownian Motion W(t)

W(t) คือ process สุ่มที่มีคุณสมบัติ 4 ข้อ:

1. W(0) = 0 เริ่มต้นที่ศูนย์
2. Increments เป็นอิสระต่อกัน: W(t)−W(s) ⊥ W(s)−W(r) สำหรับ r ≤ s ≤ t
3. Increments มีการแจกแจงปกติ: W(t)−W(s) ~ N(0, t−s)
4. Sample paths ต่อเนื่อง แต่ไม่ differentiable ที่ไหนเลย

ข้อ 3 บอกว่าในช่วงเวลา Δt ราคาขยับประมาณ σ√Δt — ยิ่งนานยิ่งขยับมาก แต่ขยับตาม √t ไม่ใช่ t



แต่ละ path เดินอิสระ — ไม่กลับหาศูนย์

2.2 Properties ของ dW

ในการคำนวณ stochastic calculus เราใช้ notation dW_t แทน W(t+dt)−W(t)

```
dW_t ~ N(0, dt) E[dW_t] = 0 Var[dW_t] = dt (dW_t)² = dt ← กฎ Itô: ไม่เป็นศูนย์!
dW_t · dt = 0 ← higher order terms ตัดทิ้ง
```

สำคัญ: $(dW)^2 = dt$ ไม่ใช่ 0

ใน calculus ธรรมดา $(dx)^2 \to 0$ ตัดทิ้ง

ใน stochastic calculus $(dW)^2 = dt$ ไม่ตัด — เพราะ W oscillates อย่างรวดเร็ว ทำให้ $(dW)^2$ สะสมเป็นค่าจริง

นี่คือที่มาของ "second-order term" ใน Itô's Lemma

2.3 Stochastic Differential Equation (SDE)

SDE คือสมการที่บอกว่า process X เปลี่ยนอย่างไรในแต่ละช่วงเวลา:

รูปแบบทั่วไปของ SDE

$$dX_t = \mu(X_t, t)\, dt + \sigma(X_t, t)\, dW_t$$

ส่วน	ชื่อ	ความหมาย
dX_t	increment	การเปลี่ยนแปลงของ X ในช่วง dt
$\mu(X_t,t)\, dt$	drift term	การเคลื่อนไหวที่คาดการณ์ได้ (deterministic)
$\sigma(X_t,t)\, dW_t$	diffusion term	ความสุ่ม (random shock)

ตัวอย่างที่คุ้นเคย: Geometric Brownian Motion (GBM) ที่ใช้ใน Black-Scholes:

```
dS_t = μ S_t dt + σ S_t dW_t (drift และ diffusion ∝ ราคา)
```

2.4 Itô's Lemma: Chain Rule สำหรับ SDE

ถ้า X_t เดิน SDE และ $f(X,t)$ เป็นฟังก์ชันที่ differentiable แล้ว f ก็เดิน SDE ด้วย:

ITÔ'S LEMMA (1 มิติ)

$$df = (\partial f/\partial t + \mu\, \partial f/\partial X + \tfrac{1}{2}\sigma^2\, \partial^2 f/\partial X^2)\, dt + \sigma\, \partial f/\partial X\, dW$$

เทียบกับ chain rule ธรรมดา

Chain rule ธรรมดา: $df = (\partial f/\partial t)\, dt + (\partial f/\partial x)\, dx$

Itô's Lemma: $df = (\partial f/\partial t)\, dt + (\partial f/\partial x)\, dX + \mathbf{\tfrac{1}{2}(\partial^2 f/\partial x^2)(dX)^2}$

ต่างกันตรง term สุดท้าย $\frac{1}{2}\sigma^2(\partial^2 f/\partial x^2)dt$ ซึ่งมาจาก $(dW)^2=dt$

ตัวอย่าง: log price ถ้า $S \sim$ GBM และ $X = \log(S)$ แล้ว:

```
f(S) = log(S), ∂f/∂S = 1/S, ∂²f/∂S² = −1/S² dX = d(log S) = (1/S) dS +
½(−1/S²)(dS)² = (1/S)(μS dt + σS dW) + ½(−1/S²)(σ²S² dt) = μ dt + σ dW − ½σ²
dt = (μ − ½σ²) dt + σ dW
```

นี่คือที่มาของ log return มี drift $= \mu - \frac{1}{2}\sigma^2$ (ไม่ใช่ μ) ซึ่งเรียกว่า Itô correction

2.5 Mean-Reverting SDE: Preview OU Process

สมการที่เราต้องการสำหรับ spread ε:

ORNSTEIN-UHLENBECK PROCESS

$$d\varepsilon_t = \kappa(\theta - \varepsilon_t)\, dt + \sigma\, dW_t$$

พจน์ drift $= \kappa(\theta - \varepsilon_t)$ ทำงานอย่างไร:

- ถ้า $\varepsilon > \theta \rightarrow$ drift เป็นลบ $\rightarrow \varepsilon$ ถูกดึงลงมา
- ถ้า $\varepsilon < \theta \rightarrow$ drift เป็นบวก $\rightarrow \varepsilon$ ถูกดึงขึ้นไป
- ถ้า $\varepsilon = \theta \rightarrow$ drift $= 0 \rightarrow$ ไม่มีแรงดึง

บทที่ 3 จะเจาะลึก OU process ทั้งหมด

ตรวจสอบความเข้าใจ 2.1

กำหนด $dX = 2X\, dt + 0.3\, dW$ และ $f(X) = X^2$

ถาม: หา df โดยใช้ Itô's Lemma

▶ ดูคำตอบ

ตรวจสอบความเข้าใจ 2.2

ทำไมนักเทรดต้อง log price ก่อน แทนที่จะใช้ price โดยตรงในการสร้าง spread ε?

▶ ดูคำตอบ

บทที่ 3: Ornstein-Uhlenbeck Process — หัวใจของ Stat Arb

OU Process คือโมเดลหลักที่ใช้อธิบายพฤติกรรม mean-reverting ของ spread ε บทนี้จะครอบคลุม: นิยาม → solution → parameters → การ calibrate จากข้อมูลจริง → การอ่านค่า half-life

3.1 SDE ของ OU Process

ORNSTEIN-UHLENBECK (OU) SDE

$$d\varepsilon_t = \kappa(\theta - \varepsilon_t)\, dt + \sigma\, dW_t$$

Parameter	ชื่อ	ความหมาย	ตัวอย่างค่า
$\kappa > 0$	Mean-reversion speed	ความเร็วดึงกลับ — ยิ่งมาก ยิ่งเร็ว	0.1–5.0 (ต่อวัน)
θ	Long-run mean	ค่ากลางที่ ε วิ่งเข้าหา	0 หรือ funding rate spread
$\sigma > 0$	Diffusion	ความผันผวนของ spread	0.01–0.5%

3.2 Solution ของ OU Process

SDE นี้ solve ได้แบบ closed-form โดยใช้ Itô's Lemma กับ $f(\varepsilon,t) = e^{\kappa t}\varepsilon$:

$$\varepsilon_t = \theta + (\varepsilon_0 - \theta)\, e^{-\kappa t} + \sigma \int_0^t e^{-\kappa(t-s)}\, dW_s$$

อ่าน solution นี้อย่างไร

- **Term แรก θ:** ค่ากลางระยะยาว
- **Term สอง $(\varepsilon_0-\theta)e^{-\kappa t}$:** ระยะห่างจากค่ากลาง ณ เวลา 0 ที่ค่อยๆ decay เข้าหา 0 ด้วยอัตรา $e^{-\kappa t}$
- **Term สาม σ∫...dW:** noise สะสมตลอดเส้นทาง — แม้จะ noisy แต่ variance bounded

Conditional distribution ณ เวลา t ถ้ารู้ ε_0:

```
E[ε_t | ε_0] = θ + (ε_0 − θ) e^{−κt} Var[ε_t | ε_0] = (σ²/2κ)(1 − e^{−2κt})
เมื่อ t → ∞: E[ε_∞] = θ (mean ระยะยาว) Var[ε_∞] = σ²/(2κ) (variance ระยะยาว — bounded!)
```

3.3 Half-Life: ตัวชี้วัดที่สำคัญที่สุด

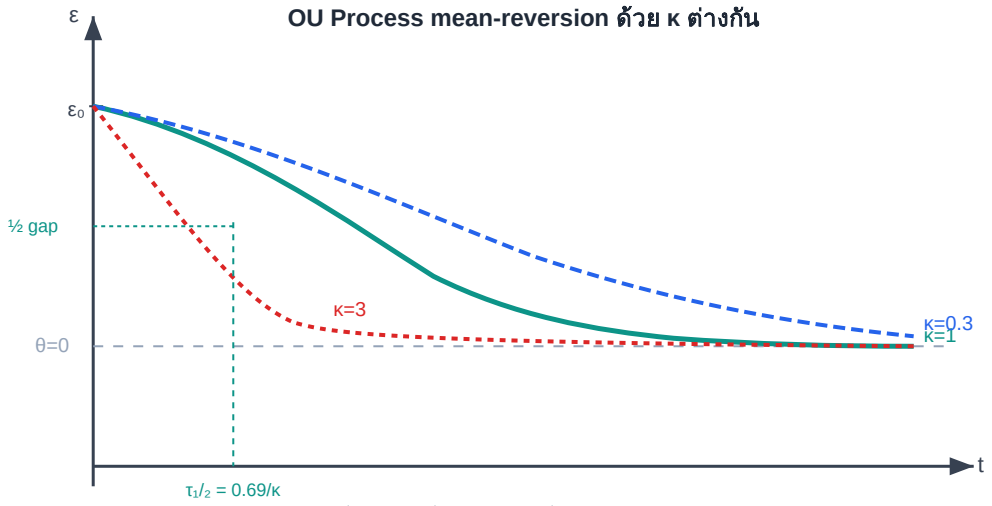
HALF-LIFE ของ OU PROCESS

$$\tau_{1/2} = \ln(2) / \kappa \approx 0.693 / \kappa$$

Half-life คือเวลาที่ใช้สำหรับ gap ระหว่าง ε และ θ ลดลงครึ่งหนึ่ง:

```
E[ε_{τ₁/₂} | ε_0] = θ + (ε_0 − θ) · ½
```

κ (ต่อวัน)	Half-life (วัน)	ความหมายสำหรับ trader
0.1	6.9 วัน	Spread ใช้เวลา ~1 สัปดาห์กว่าจะครึ่งทาง — ถือนาน
0.5	1.4 วัน	~1.5 วัน — swing trade ระยะสั้น
1.0	16.6 ชั่วโมง	ภายในวัน — intraday strategy
2.0	8.3 ชั่วโมง	ครึ่งวัน — fast mean-reversion
5.0	3.3 ชั่วโมง	Spread กลับเร็วมาก — ต้องมี execution เร็ว
10.0	1.7 ชั่วโมง	เกือบ HFT — spread เกือบไม่มีโอกาส



ยิ่ง κ มาก ยิ่งกลับเร็ว — ยิ่งต้อง execute เร็ว

3.4 Stationary Distribution

เมื่อ $t \to \infty$ OU process เข้าสู่ stationary distribution (steady-state) ซึ่งเป็น Normal:

```
ε_∞ ~ N(θ, σ²/(2κ)) Stationary std dev: σ_ε = σ/√(2κ)
```

นี่คือ distribution ที่เราจะใช้ตั้ง entry/exit threshold ใน Chapter 10 เช่น เข้า trade เมื่อ $\varepsilon > \theta + 2\sigma_\varepsilon$ หรือ $\varepsilon < \theta - 2\sigma_\varepsilon$

Relationship ระหว่าง parameters กับ การ trade

ถ้าต้องการ...	ต้องการ parameter
Signal บ่อยขึ้น (threshold hit มากขึ้น)	σ_ε สูง หรือ threshold ต่ำลง
Holding time สั้นลง	κ สูง (half-life สั้น)
Entry คาดเดาได้ (θ stable)	θ ไม่เปลี่ยนมาก (non-drifting)
Risk ต่อ trade ต่ำ	σ ต่ำ (spread ไม่กระโดด)

3.5 AR(1): OU Process ในโลกข้อมูล Discrete

ข้อมูล price ที่มีเป็น discrete (รายนาที รายชั่วโมง) ไม่ใช่ continuous เมื่อ discretize OU process ด้วย time step Δt:

```
ε_{t+Δt} = a + b · ε_t + η_t โดยที่: a = θ(1 − e^{−κΔt}) ≈ κθΔt (ถ้า κΔt เล็ก) b
= e^{−κΔt} ≈ 1 − κΔt Var(η_t) = σ²(1−e^{−2κΔt})/(2κ) ≈ σ²Δt
```

นี่คือ AR(1) — autoregressive model order 1 ที่คุ้นเคย ตัวแปรสำคัญคือ:

ความสัมพันธ์ AR(1) ↔ OU PARAMETERS

$$\kappa = -\ln(b) / \Delta t \qquad \theta = a / (1-b)$$

ทำไม AR(1) ต้องมี |b| < 1 สำหรับ mean-reversion

$b = e^{-\kappa\Delta t}$ เมื่อ $\kappa > 0 \rightarrow b = e^{(\text{ลบ})} \rightarrow 0 < b < 1$ ✓

ถ้า $b \geq 1 \rightarrow \kappa \leq 0 \rightarrow$ ไม่ mean-reverting (random walk หรือ explosive)

ถ้า $b = 0 \rightarrow \kappa = \infty \rightarrow$ กลับทันทีในทุก time step (white noise)

การทดสอบ: ถ้า b ใกล้ 1 มาก (เช่น 0.99) → κ เล็กมาก → half-life ยาว → spread กลับช้า

3.6 Calibrating OU Parameters จากข้อมูลจริง

วิธีหา κ, θ, σ จากข้อมูล spread $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T$:

1. **Run OLS regression AR(1):**
   Regress ε_t บน ε_{t-1}: $\varepsilon_t = a + b \cdot \varepsilon_{t-1} + \eta_t$

2. **Extract parameters:**
   $\hat{\theta} = a / (1-b)$, $\hat{\kappa} = -\ln(b) / \Delta t$

3. **Estimate σ:**
   $\hat{\sigma}^2 = \mathrm{Var}(\hat{\eta}) \times 2\hat{\kappa} / (1 - e^{-2\hat{\kappa}\Delta t}) \approx \mathrm{Var}(\hat{\eta}) / \Delta t$

4. **Compute Stationary σ_ε:**
   $\hat{\sigma}_\varepsilon = \hat{\sigma} / \sqrt{(2\hat{\kappa})}$ หรือ ≈ std(ε residuals จาก mean $\hat{\theta}$)

5. **Compute Half-life:**
   $\hat{\tau}_{1/2} = \ln(2) / \hat{\kappa} = -\ln(2) \times \Delta t / \ln(b)$

```
# Pseudo-code: Calibrate OU from spread series
function calibrate_ou(eps, delta_t):
  n = len(eps)
  eps_lag = eps[0:n-1] # ε_{t-1}
  eps_curr = eps[1:n] # ε_t

  # OLS: eps_curr = a + b * eps_lag + residuals
  a, b = ols(eps_lag, eps_curr)
  residuals = eps_curr - (a + b * eps_lag)
```

```
# Convert to OU params
theta = a / (1 - b)
kappa = -log(b) / delta_t
sigma_sq_dt = var(residuals)
sigma = sqrt(sigma_sq_dt * 2 * kappa / (1 - exp(-2*kappa*delta_t)))

half_life = log(2) / kappa
sigma_stat = sigma / sqrt(2 * kappa) # stationary std dev

return {theta, kappa, sigma, half_life, sigma_stat}
```

ตัวอย่าง: Calibrate OU บน Bybit↔Lighter BTC Spread

สมมติมีข้อมูล 30 วัน รายนาที (Δt = 1/1440 วัน = 1 นาที) ของ log spread:

```
ε_t = log(P_Bybit_t) − β·log(P_Lighter_t) , β = 1.003 OLS AR(1)
result: a = 0.000002, b = 0.9985 Residual std = 0.00012 κ =
−ln(0.9985) / (1/1440) = 0.0015 × 1440 = 2.16 ต่อวัน θ = 0.000002 /
(1−0.9985) = 0.00133 (≈ 0.133% basis) σ_stat = std(ε) ≈ 0.00080 (0.08%
stationary std) Half-life = ln(2) / 2.16 ≈ 0.32 วัน ≈ 7.7 ชั่วโมง
```

สรุป: Spread ระหว่าง Bybit↔Lighter กลับหาค่ากลางภายใน ~8 ชั่วโมง เป็นโอกาส intraday stat arb

Entry threshold ที่ 2σ_stat ≈ 0.16% จาก θ — ถ้า spread เบี่ยงเบินมากกว่านี้ถือว่า "ห่างเกินปกติ"

3.7 ข้อจำกัดของ OU Process เดี่ยว

OU process ธรรมดาสมมติหลายอย่างที่อาจไม่เป็นจริง:

สมมติฐาน OU	ปัญหาในตลาดจริง	โมเดลที่แก้
κ, θ, σ คงที่ตลอดเวลา	Regime เปลี่ยน — spread อาจ "หลุด" จากค่ากลางนานๆ	Regime-Switching OU (Ch.9)
Noise เป็น Gaussian	ราคา jump ได้ (news, liquidation)	Jump-Diffusion OU (Ch.8)
σ คงที่	ช่วง volatile spread ผันผวนมากขึ้น	GARCH on residuals (Ch.7)
β คงที่	Hedge ratio เปลี่ยนตามสภาวะตลาด	Kalman Filter (Ch.4)

บทถัดๆ ไปจะ relax สมมติฐานเหล่านี้ทีละข้อ แต่ OU Process ยังเป็น baseline ที่ดีที่สุด

ตรวจสอบความเข้าใจ 3.1

ข้อมูล spread ε รายชั่วโมง (Δt = 1/24 วัน) ของ Bybit↔Lighter BTC ให้ผล OLS AR(1): b = 0.94, a = 0.00008

ถาม: (a) คำนวณ κ ต่อวัน (b) half-life เป็นชั่วโมง (c) ถ้า ε_0 = 0.15% (เบี่ยงเบิน 0.07% จาก θ) ε จะเหลือครึ่งหนึ่งของการเบี่ยงเบินหลังกี่ชั่วโมง?

▶ ดูคำตอบ

ตรวจสอบความเข้าใจ 3.2 (ออกแบบ strategy)

OU parameters จาก calibration: κ = 3.0 ต่อวัน, θ = 0, σ_stat = 0.05%

ต้นทุนการ trade รวม (fee + spread) ≈ 0.04% per side ต่อ trade

ถาม: ควรตั้ง entry threshold ที่ระยะห่างจาก θ เท่าไร เพื่อให้มั่นใจว่า expected profit > ต้นทุน?

▶ ดูคำตอบ

สรุป Part I

สิ่งที่ได้จาก Part I

1. **Synthetic Process:** $\varepsilon = F(A) - \beta \cdot F(B)$ สร้าง process ใหม่ที่มี mean-reversion จาก 2 assets ที่เดิน random walk
2. **β = Entry Ratio:** ไม่ใช่ 1:1 เสมอ — β กำหนดว่า long A 1 หน่วยต้อง short B กี่หน่วยเพื่อให้ ε stationary
3. **Stochastic Calculus:** $dW \sim N(0,dt)$, $(dW)^2=dt$, Itô's Lemma เป็นเครื่องมืออ่าน SDE
4. **OU Process:** $d\varepsilon = \kappa(\theta-\varepsilon)dt + \sigma dW$ — โมเดล mean-reversion ที่ solve ได้แบบ closed-form
5. **Half-life = ln(2)/κ:** วัดว่า spread ใช้เวลากลับครึ่งทางนานแค่ไหน — กำหนด holding period และ strategy type
6. **Calibration:** OLS AR(1) บน discrete spread data → $b = e^{-\kappa\Delta t}$ → recover κ, θ, σ

Part II: Hedge Ratio β → **Part III: Cointegration Tests** → **Part IV: Spread Dynamics** → **Part V: Signal & Sizing**

Statistical Arbitrage — Part I | Bybit↔Lighter BTC Perpetual Running Example | v1.0