Arbitrage
จากแนวคิดสู่การปฏิบัติ

Part V: Statistical Arbitrage (บท 18–21)

Pairs Trading, Mean Reversion, Factor Models, ML Overview

⚖️ อ่านก่อน: Stat Arb สำหรับรายย่อย — ทำอะไรได้จริง และต้องระวังอะไร

เนื้อหา Part นี้สอน **คณิตศาสตร์และเครื่องมือที่รายย่อยทำได้จริง** (cointegration test, OU, Kalman — เขียน Python ฟรีได้ทั้งหมด) แต่ต้องแยกให้ออกระหว่าง *"ทำได้"* กับ *"กำไรจริงหลังหักต้นทุน"*:

ทำได้ สร้าง/วิจัยโมเดล, เทรดคู่หุ้นใหญ่สภาพคล่องสูงด้วยทุนพอประมาณ, ใช้เป็นเครื่องมือ *เรียนรู้* market-neutral (พอร์ตที่ขึ้น/ลงตามตลาดไม่กระทบ)

ระวัง งานวิจัยเกือบทั้งหมดรายงานเป็นผลตอบแทน **ก่อนหักต้นทุน (gross)** และ **edge** (ความได้เปรียบที่ทำให้ชนะตลาด) เสื่อมลงต่อเนื่องหลังปี 2000

เกินกำลัง พอร์ต market-neutral หลายร้อยตัว, short หุ้นเล็ก/หา borrow (ยืมหุ้นมา short) ยาก, แข่ง execution กับ HFT

หลักฐานเรื่อง decay และต้นทุนอยู่ในกล่องของแต่ละบท + รายการอ้างอิงท้าย Part

บทที่ 18: Pairs Trading

Stat Arb เริ่มต้นจากแนวคิดง่ายๆ: **"หุ้น 2 ตัวที่เคลื่อนไหวด้วยกัน → เมื่อแยกจากกัน → เข้าเทรด"**

18.1 Correlation vs Cointegration

	Correlation	Cointegration
วัดอะไร	ทิศทางเหมือนกันไหม	**ระยะห่าง**คงที่ไหม
ปัญหา	2 หุ้นอาจ drift ออกจากกันตลอดกาล	Spread กลับหาค่าเฉลี่ย
ตัวอย่าง	BTC กับ ETH: correlated แต่ ratio เปลี่ยนเรื่อยๆ	Coke vs Pepsi: spread mean-reverts
ใช้กับ Arb?	ไม่พอ!	**ต้องใช้ตัวนี้!**

Engle-Granger Cointegration Test[13]

1. Regress A vs B: $A = \alpha + \beta \times B + \varepsilon$
2. ทดสอบ residuals (ε) ว่า stationary ไหม (ADF test)
3. ถ้า ε stationary → A กับ B cointegrated → **Spread = $\varepsilon = A - \alpha - \beta \times B$ จะ mean-revert!**

ข้อควรระวัง: ผล regress ไม่สมมาตร — regress A กับ B vs B กับ A ได้ β และผล ADF ต่างกัน ควรลองทั้งสองทิศ (หรือใช้ Johansen test[17] ซึ่งไม่ขึ้นกับทิศทางและรองรับมากกว่า 2 ขา)

📊 **เข้าใจ p-value ให้ถูก: "ระดับของหลักฐาน" ไม่ใช่สวิตช์ผ่าน/ตก**

เข้าใจผิดที่พบบ่อย: "ราคาหุ้นไม่เป็น normal → p-value ของ ADF/cointegration ใช้ไม่ได้" — **ไม่จริง** ADF ไม่ได้ต้องการให้ข้อมูลเป็น normal เลย: test statistic ของมันมี **distribution เฉพาะตัว** (ไม่ใช่ normal หรือ t) ที่สร้างจาก limit theory + simulation โดยตรง[14][18] ดังนั้นการทดสอบ normality (เช่น Shapiro–Wilk) **ไม่ใช่ด่านก่อนใช้ ADF** — สองอย่างตอบคนละคำถาม

กับดักที่คน DIY พลาดบ่อยสุด: ADF บน residual ที่*ประมาณ β มาแล้ว* ห้ามใช้ critical values ปกติ — OLS จะ "ดูด" ให้ residual ดู stationary เกินจริง ต้องใช้ critical values เฉพาะของ Engle–Granger/MacKinnon ที่ติดลบมากกว่า[18] → ในทางปฏิบัติ: **ใช้ `coint()` ของ statsmodels ซึ่งจัดการให้แล้ว อย่ารัน `adfuller()` บน residual เอง**

ถ้า p-value สูง แปลว่าอะไร: ADF มีปัญหา **power ต่ำ**เมื่อ spread กลับช้าหรือ sample สั้น (แยก φ=0.98 จาก φ=1 ไม่ออก) — p สูงจึงแปลว่า *"ยังสรุปไม่ได้"* ไม่ใช่ *"ไม่ mean-revert แน่นอน"* ทางแก้ที่ดีกว่าการลดเกณฑ์: **cross-check ด้วย KPSS**[16] ซึ่งมี null กลับด้าน (null = stationary) — ถ้า ADF reject *และ* KPSS ไม่ reject = หลักฐานแข็งสองทาง; ขัดกัน = ก้ำกึ่ง เก็บไว้ดูต่อ

สรุป: significance เป็น "ระดับของหลักฐาน" — จะ*อ้าง*ว่าคู่ cointegrated ต้องมีผล reject จาก test ที่ถูก spec แต่การไม่ reject = ยังอ้างไม่ได้ (เข้า Watchlist) ไม่ใช่หลักฐานว่าความสัมพันธ์ไม่มีอยู่ — *ใช้ p-value เป็นหลักฐาน อย่าใช้เป็น strategy*

18.2 Trading the Spread

```
Spread = Price_A − β × Price_B
Z-score = (Spread - Mean) / StdDev

Entry: Z > +2 → Short Spread (Short A, Long β×B)
Entry: Z < -2 → Long Spread (Long A, Short β×B)
Exit: Z ≈ 0 → Close ทั้งสองขา
Stop: |Z| > 4 → Cut loss (structural break?)
```

🪤 กับดักที่กฎ Z-score ข้างบนไม่ได้บอก: "3σ ที่ไม่มี headroom"

ช่วงที่ตลาดนิ่ง spread แคบ ค่า σ จะ**ยุบลงตาม** → เส้น mean กับ ±1σ ±2σ ±3σ **เบียดกันจนเกือบทับกัน** พอราคาขยับนิดเดียวก็แตะ 3σ แล้ว → ระบบเปิดออร์เดอร์ → **ค่าธรรมเนียมกับ spread กินกำไรหมดตั้งแต่ยังไม่เริ่ม**

ถ้ามองด้วยตา คุณจะไม่เข้าเทรดตอนนั้น เพราะเห็นว่า "ระยะทางมันสั้นเกินจะคุ้ม" — **แต่โมเดลมองไม่เห็น** เพราะ z-score เป็นค่าที่*ไม่มีหน่วย*: มันบอกว่า "ห่างกี่ σ" แต่ไม่บอกว่า "ห่างกี่บาท" การหารด้วย σ ได้**ทิ้งข้อมูลที่จำเป็นต่อการเทียบกับต้นทุนไปแล้ว**

นี่คือความผิดพลาดแบบเดียวกับที่บท 18 เตือนเรื่อง p-value เป๊ะๆ — z = 3 ในช่วงบีบตัวคือ "มีนัยสำคัญทางสถิติ แต่ไร้ความหมายทางเศรษฐกิจ" เวอร์ชันเทรด (แบบเดียวกับ r = 0.08 ที่ p < 0.001)

⚠️ **อันตรายซ้อน:** ช่วงที่แบนด์บีบตัวยัง*เพิ่มโอกาส breakout* ด้วย (ความผันผวนต่ำมักตามด้วยความผันผวนสูง) — การเข้า mean-reversion ตอนนั้นจึงแย่สองชั้น: headroom น้อย *และ* โอกาสที่ spread จะวิ่งหนีสูงกว่าปกติ

🔧 **ทางแก้: บังคับให้สัญญาณมี "หน่วยเงิน" ก่อนยิง**

อนุมานเกณฑ์ออกมาตรงๆ ได้: กำไรที่จะจับได้ = $(z_{เข้า} - z_{ออก}) \times \sigma$ และเราต้องการให้มันเป็น k เท่าของต้นทุนไป-กลับ

```
(z_เข้า − z_ออก) × σ ≥ k × ต้นทุนไป-กลับ แทนค่าจริง z_เข้า = 2, z_ออก = 0.5, k = 3 (อยากได้กำไร 3 เท่าของต้นทุน): 1.5 × σ ≥ 3 × cost → σ ≥ 2 × ต้นทุนไป-กลับ ← ถ้าต่ำกว่านี้ ไม่ต้องดู z เลย
```

วิธีที่ 1 — ตั้งพื้นให้ σ (แก้ที่ต้นเหตุ): บังคับ σ ไม่ให้ต่ำกว่าพื้น แล้ว z จะ*ไม่พองเอง*ตอนตลาดนิ่ง ไม่ต้องเขียนตัวกรองแยก

```
sigma_floor = 2 × ต้นทุนไป-กลับ sigma_eff = max(sigma_rolling, sigma_floor) z = (spread − mean) / sigma_eff
```

วิธีที่ 2 — เพิ่มด่านหน่วยเงินคู่กับ z (อ่านออกชัด):

```
เข้าเทรดเมื่อ: |z| ≥ 2 ← นัยสำคัญทางสถิติ และ |spread − mean| ≥ 3 × cost ← นัยสำคัญทางเศรษฐกิจ
```

แนะนำ ใช้ทั้งสองวิธีคู่กัน — วิธีที่ 1 กัน z พอง วิธีที่ 2 เป็นด่านสุดท้ายที่อ่านแล้วเข้าใจทันทีว่าทำไมถึงไม่เข้า

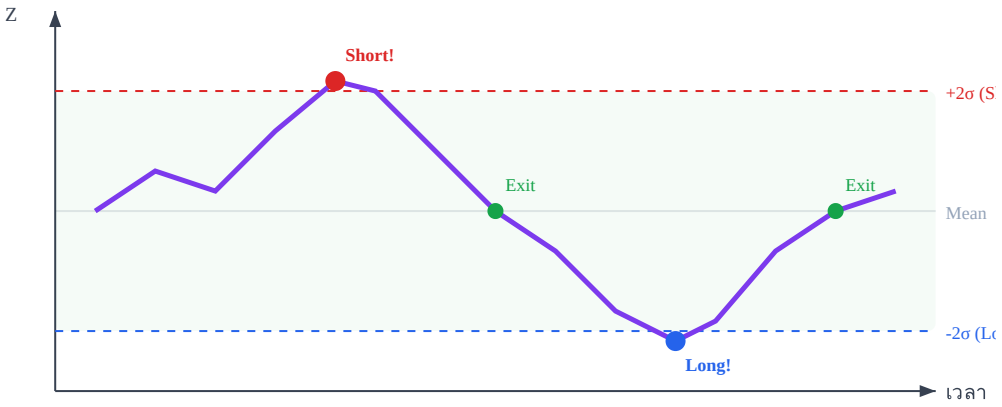
ทางเลือก ถ้าอยากให้สะอาดที่สุดเชิงแนวคิด: **เลิกใช้ z เป็นตัวตัดสิน** แล้วใช้ `edge_ratio = |spread − mean| / ต้นทุนไป-กลับ` เข้าเมื่อ ≥ 3 — วิธีนี้*ไม่มีทางเกิดปัญหานี้ตามนิยาม* เพราะหน่วยถูกโดยโครงสร้าง

โค้ดใน `code/pairs_trading_starter.py` ใส่ทั้งสองด่านให้แล้ว (พารามิเตอร์ `cost_bps` และ `edge_k`) — บรรทัด `[6]` สัญญาณวันนี้ จะบอกด้วยว่าถูกด่านไหนปฏิเสธ

⚠️ **ก่อน Short — ความเสี่ยงที่ไม่สมมาตรกับ Long**

ขา short ของ pairs trade **ไม่เหมือนการ Long**:

- **ขาดทุนได้ไม่จำกัด** (ราคาขึ้นได้ไม่มีเพดาน) + อาจโดน **margin call / forced buy-in / ถูกเรียกหุ้นคืน (recall)**
- เสีย **ค่า borrow ทุกวัน**ที่ถือ
- **stop ที่ |Z|>4 ไม่รับประกันราคา** — structural break ทำให้ spread พุ่งทะลุแล้วออกได้แพงกว่าที่คิด (นี่คือสิ่งที่ทำให้ LTCM เจ๊บ — ดูกล่องท้ายบท 21)



📊 หลักฐานวิจัย: Pairs Trading ได้ผลจริงไหม?

งานวิจัยต้นฉบับปี 2006[1] ใช้ distance method (formation 12 เดือน, เทรด 6 เดือน, เปิดที่ 2σ) บนหุ้นสหรัฐ ปี 1962–2002 พบผลตอบแทนส่วนเกิน **~11% ต่อปี (ก่อนหักต้นทุน)**

แต่การศึกษาต่อมาปี 2010[2] แสดงว่ากำไร **เสื่อมลงชัดเจน**: top-20 pairs ได้ 0.86%/เดือน (1962–1988) → 0.37% (1989–2002) → **0.24% (2003–2009)** และงานปี 2012[3] พบว่า **เมื่อหักค่าคอม + market impact + ค่า borrow แล้ว เหลือ ~0.30%/เดือน เฉพาะคู่ที่จับคู่ดีมากเท่านั้น** — คู่ทั่วไปต้นทุนกินกำไรเกือบหมด

ระวัง ค่าคอม $0 ช่วยได้ แต่ **spread + slippage + ค่า borrow ฝั่ง short** คือต้นทุนจริงที่ยังอยู่ และโอกาสที่เหลือมักอยู่ในหุ้นเล็กที่ short ยาก — บทวิจารณ์รวมปี 2017[4] สรุปว่าวิธี distance แบบง่ายเสื่อมไปนานแล้ว ส่วน cointegration มีพื้นฐานทฤษฎีแข็งกว่า

ดูคำเตือนเรื่อง data-snooping และ short-sale constraints ในบทวิจารณ์ [4] — รายการอ้างอิง [1]–[4] ท้าย Part

บทที่ 19: Mean Reversion

19.1 Ornstein-Uhlenbeck Process

เปรียบเทียบ: "ยางยืด"

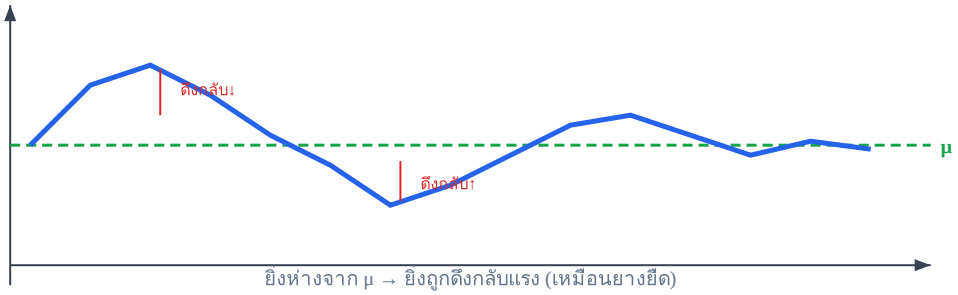
ลองนึกว่า spread ถูกผูกด้วยยางยืดกับค่าเฉลี่ย μ — ยิ่งดึงออกไกล ยิ่งถูกดึงกลับแรง

θ = ความแข็งของยาง (สูง = กลับเร็ว), σ = ลมที่พัดให้แกว่ง (noise)

```
dX = θ(μ - X)dt + σdW

θ = speed of mean reversion (ยิ่งสูง ยิ่งกลับเร็ว)
μ = long-run mean (ค่าที่ดึงดูดกลับ)
σ = volatility

Half-life = ln(2) / θ ← "นานแค่ไหนกว่า spread จะกลับครึ่งทาง"
```



ตัวอย่าง

ถ้า Half-life = 5 วัน → spread กลับครึ่งทางใน 5 วัน → ดีมาก trade ได้

ถ้า Half-life = 60 วัน → ช้าเกินไป → tie up capital นาน → อาจไม่คุ้ม

📊 **หลักฐานวิจัย: OU + Half-life ใช้กันจริง**

โมเดล "ยางยืด" นี้ไม่ใช่แค่ทฤษฎี — มืออาชีพใช้จริง งานวิจัยปี 2010[5] ใช้ OU จับ residual ของหุ้นหลังหักปัจจัย (PCA / sector ETF) บนตลาดสหรัฐ 1997–2007 ได้ **Sharpe ~1.44 (PCA)** แต่ **ลดเหลือ ~0.9 ช่วงปลาย (2003–2007)**

⚠️ ตัวเลขนี้เป็น **gross** (ก่อนต้นทุน) + สมมติเทรดทั้ง universe รายวันและ short ได้ทุกตัว; 1.44 เป็นค่าช่วงต้น ไม่ใช่ค่าคงที่ที่คาดหวังได้ — ตอกย้ำธีม edge เสื่อมเหมือนบท 18

ทำได้ ส่วนคำนวณ (fit OU/half-life ด้วย AR(1)/OLS) ทำได้ 100% บน `statsmodels` — ไม่ต้อง short ไม่ต้องใช้ทุน ใช้กรองคู่ที่ half-life สั้น (~5–30 วัน) ได้เลย แต่ **สั้นเกินไป (1–2 วัน) มักเป็น noise/bid-ask bounce** และยิ่งสั้นยิ่งเทรดถี่ = ต้นทุนสูง

19.2 Kalman Filter — Dynamic Hedge Ratio

ปัญหา: β (hedge ratio) เปลี่ยนตามเวลา → ใช้ rolling regression ไม่ดีพอ
Kalman Filter: อัปเดต β แบบ real-time ตามข้อมูลใหม่ → **adaptive pairs trading**

เปรียบเทียบง่ายๆ

Kalman = "ปรับ β ทีละนิดทุกวันตามข้อมูลใหม่" แทนที่จะคำนวณ β ใหม่หมดทั้งหน้าต่าง — เหมือนหมุนพวงมาลัยแก้ทีละนิด ไม่ใช่เลี้ยวกระชากทุกครั้งที่เจอโค้ง

📊 **หลักฐานวิจัย: Kalman ดีกว่า rolling regression อย่างไร**

ตำราวิชาการปี 2025 (Cambridge)[6] แสดงว่า hedge ratio จาก rolling regression **แกว่งแรงและ noisy** (เช่น 0.6–1.2) ขณะที่ Kalman ให้ค่าที่ **นิ่งกว่าแต่ยังปรับตามได้** (เช่น 0.55–0.65) และตัด parameter "ความยาว window" ที่ต้องเดาออกไป

ทำได้ มีไลบรารีพร้อม (`pykalman`, `statsmodels`) — ส่วนการเทรดจริงยังติดข้อจำกัด short เหมือนทุกกลยุทธ์ long/short

19.3 Regime Detection

เมื่อไร Mean Reversion ใช้ไม่ได้?

Trending market → spread ไม่ mean-revert → **ขาดทุนต่อเนื่อง**

Structural break → cointegration หาย (M&A, sector change)

ต้องมี **regime detection**: ถ้าตลาดเปลี่ยน → หยุดเทรด pairs

บทที่ 20: Factor-Based Statistical Arbitrage

20.1 จาก Pairs สู่ Portfolio

เทรด 1 คู่ → diversify ด้วยหลายคู่ → สร้าง **portfolio ของ pairs trades**
→ single-pair risk ลดลง → more consistent returns

20.2 Factor Models

```
Return_i = α_i + β₁×Market + β₂×Size + β₃×Value + ... + ε_i

α (alpha) = ส่วนที่ model อธิบายไม่ได้ = edge ที่เราหา!
β×Factor = systematic risk → hedge ออกได้
```

20.3 Market-Neutral Portfolio

```
Portfolio beta = 0 (ตลาดขึ้น/ลงไม่กระทบ)
Long stocks ที่มี positive alpha + Short stocks ที่มี negative alpha
→ กำไรจาก alpha เท่านั้น ไม่ขึ้นกับทิศตลาด [Lv.4 Structured]
```

PCA Factors (จากเล่มคณิตศาสตร์ บท 8)

ใช้ Eigenvalues ของ Covariance Matrix → หา principal components
PC1 มักคล้าย market factor; PC ถัดๆ ไป *อาจ* สะท้อน sector/style แต่ไม่เสมอ และตีความยาก/ไม่นิ่ง
→ Hedge ทีละ factor → เหลือแค่ alpha

⚠️ ความจริงสำหรับรายย่อย: Market-Neutral เต็มรูปแบบ = เกินกำลัง

การถือพอร์ต long-short หลายสิบ–หลายร้อยตัวเพื่อให้ beta = 0 ต้องใช้:

- **Short ได้ทุกตัว** — แต่หุ้นเล็กที่มี alpha สูงมักหา borrow ไม่ได้/ค่า borrow แพง (HTB 5–50%+/ปี)
- **ทุนมาก** — margin ทั้งสองขา + ต้องกระจายหลายตำแหน่งเพื่อลด single-pair risk
- **ข้อมูลคุณภาพสถาบัน** — point-in-time, survivorship-bias-free (CRSP/Compustat)

งานวิจัยปี 2010[5] เองก็จำกัดเฉพาะหุ้นใหญ่ที่ borrow ได้ และผลตอบแทนที่รายงานเป็น gross — งานปี 2015–2025 ชี้ว่า edge ถูกบีบจากการแข่งขัน + ค่า borrow ที่ *สูงขึ้น*

ทำได้ เริ่มจากพอร์ต pairs ไม่กี่คู่ในหุ้นใหญ่/ETF สภาพคล่องสูง — เก็บ market-neutral หลายร้อยตัวไว้เป็น "ของสถาบัน"

บทที่ 21: Machine Learning ใน Stat Arb

21.1 Feature Engineering

สร้าง features จากข้อมูลดิบ: momentum, mean-reversion signals, vol ratios, order flow

21.2 Walk-Forward Cross-Validation

Time Series ≠ Regular Data!

ห้ามใช้ random K-fold → data leakage (ใช้อนาคตทำนายอดีต)
ต้องใช้ **Walk-Forward**: train on past → test on next period → roll forward

21.3 Overfitting — ศัตรูตัวฉกาจ

```
Backtest Sharpe สูงผิดปกติ (เช่น 4–5) → สันนิษฐานไว้ก่อนว่า overfit
Backtest Sharpe ~1–2 → "พอเป็นไปได้" → ต้องทดสอบ out-of-sample จริง

(ตัวเลข 5 vs 1.5 เป็น rule of thumb ไม่ใช่เกณฑ์ที่พิสูจน์ในงานวิจัย — ดูหลักฐานด้านล่าง)
```

📊 **หลักฐานวิจัย: ทำไม Backtest Sharpe สวยๆ ถึงหลอก**

งานวิจัยปี 2014[7] พิสูจน์ว่า แค่มีข้อมูลรายวัน 5 ปี แล้วลองสัก **~45 กลยุทธ์** ก็**คาดได้โดยเฉลี่ย**ว่าจะเจออย่างน้อย 1 ตัวที่ in-sample Sharpe = 1.0 ทั้งที่ **out-of-sample ที่แท้จริง = 0** — Sharpe สูงสุดที่ "ฟลุก" ได้โตตาม **√(2·ln N)** ของจำนวนครั้งที่ลอง (N)

ทางแก้: **Deflated Sharpe Ratio**[8] ปรับ Sharpe ตามจำนวนครั้งที่ลอง + ความยาวข้อมูล + skew/kurtosis ส่วนงานสำรวจ "factor zoo" 316+ ปัจจัย ปี 2016[9] แนะนำว่าสัญญาณใหม่ควรผ่าน **t-stat > 3.0** ไม่ใช่ 2.0

ระวัง การจูน parameter หลายๆ ค่าแล้วเก็บตัวที่ดีสุด = กระบวนการ overfit ตรงเป๊ะ — และเครื่องมือ retail มักไม่นับ N ให้ ทำให้ Sharpe ที่เห็น "มองโลกในแง่ดีเกินจริง" เสมอ

📊 หลักฐานวิจัย: "Linear ชนะ Neural Net เสมอ" — จริงหรือไม่?

ไม่จริงตามที่เคยเชื่อ. งานวิจัยใหญ่ปี 2020[10] (หุ้นสหรัฐ 1957–2016) พบว่า **neural net ตื้นๆ และ tree ชนะ linear ใน gross** — long-short Sharpe ≈ **1.35 (NN) เทียบ 0.61 (linear)** [วัดแบบ value-weighted ทั้งคู่] เก่งจากการจับ **ปฏิสัมพันธ์ที่ไม่เป็นเส้นตรง**; ที่น่าสนใจคือ **NN ตื้น (~3 ชั้น) ชนะ NN ลึก** เพราะ finance noise สูง deep net จึง overfit

แต่หลังหักต้นทุน edge หด/หาย: งาน ML บน S&P 500 ปี 2017[11] gross >0.45%/วัน เหลือ ~0.25%/วันหลังต้นทุน และงานปี 2023[12] พบว่าความได้เปรียบของ ML หายไปเมื่อตัดหุ้นเล็ก/distressed และคิดต้นทุนจริง

แนวทางรายย่อย **เริ่มจากโมเดลง่ายก่อน** เทรดไม่ถี่ — ลดทั้ง overfitting และต้นทุน (สอดคล้องกับที่ NN ตื้นชนะ NN ลึก) รายย่อย *replicate งานวิจัยได้ แต่เก็บกำไร gross หลังต้นทุนแทบไม่ได้*; โมเดลซับซ้อน ≠ กำไรมากกว่าเสมอ

เรื่องจริง: Pairs Trading — Royal Dutch Shell A vs B (1907-2005)

Shell มีหุ้น 2 classes (Dutch=A, English=B) สัดส่วน 60:40 → ราคาควรเป็น 60:40 ตลอด
จริงๆ: ratio เบี่ยงเบน ±15% เป็นประจำ → pairs traders ทำกำไรหลายทศวรรษ
แต่ **LTCM ก็เทรดคู่นี้แล้วเจ็บ** เพราะ ratio กว้างขึ้นก่อน converge → margin call
บทเรียน: Pairs ที่ดีที่สุดก็มี timing risk — ต้อง size ให้รอดถึงวัน converge

ลองคิดดู (บท 18–21)

1. หุ้น A กับ B มี correlation = 0.95 → ใช้ pairs trading ได้ไหม? ทำไม?
2. Spread มี half-life = 3 วัน vs 90 วัน → อันไหนดีกว่าสำหรับ pairs trading?

เฉลย: 1. Correlation สูงไม่พอ! ต้องทดสอบ cointegration — ถ้าไม่ cointegrated spread อาจ drift ไปเรื่อยๆ 2. 3 วันดีกว่ามาก — กลับเร็วกว่า tie up capital น้อยกว่า

สรุป Part V (บท 18–21) — Part 6/9 ✓

✓ **Pairs Trading**: Cointegration > Correlation, Z-score entry/exit

✓ **p-value**: เป็น "ระดับของหลักฐาน" ไม่ใช่สวิตช์ — จัดคู่ลง Evidence Tiers (Reject→Trade)

✓ **OU Process**: θ = speed, Half-life กำหนดว่าคุ้มไหม

✓ **Kalman Filter**: Dynamic hedge ratio

✓ **Factor Models**: alpha = edge, beta = hedge ออกได้

✓ **Market-Neutral**: Long alpha + Short alpha, beta = 0

✓ **ML**: Walk-Forward CV (ห้าม K-fold), Overfitting = ศัตรู (Deflated Sharpe, t>3)

✓ **รายย่อย+ML**: เริ่มจากโมเดลง่าย (NN ตื้น>ลึก) — ของซับซ้อนชนะแค่ gross ต้นทุนกลืน edge

✓ **หลักฐาน decay**: pairs & stat-arb เสื่อมลงหลังปี 2000 — gross ≠ net

✓ **ทั้งหมดเป็น [Lv.4 Structured]** — ไม่ใช่ arb จริง มี statistical edge (และ edge บางลงเรื่อยๆ)

🛠️ ลงมือทำ: จากอ่าน → เทรดคู่แรก

ความรู้ในบท 18–21 จะเป็นแค่ทฤษฎีถ้าไม่ได้ลงมือ — ส่วนนี้คือ **playbook ทำตามได้จริง** ด้วย Python ฟรี โค้ดเต็มที่รันได้อยู่ใน repo: `code/pairs_trading_starter.py`

⚠️ ส่วนนี้เพื่อ**การศึกษา ไม่ใช่คำแนะนำการลงทุน** — backtest/ผลในอดีตไม่รับประกันอนาคต และมีความเสี่ยงขาดทุน (โดยเฉพาะขา short)

> 🎯 **เป้าหมายเดือนแรกไม่ใช่ "กำไร"**
>
> คือ **รันสกรีนแรกของคุณให้ได้** + เข้าใจว่าทำไมต้นทุนกินกำไร — แค่นี้คุณก็นำหน้าคนที่อ่านแล้วไม่ลงมือ 99% แล้ว ส่วนกำไรจริงเป็นเรื่องระยะยาวที่ต้องพิสูจน์ด้วย paper trade ก่อน

Workflow 6 ขั้น (จากศูนย์ → คู่แรก)

```
1) เลือก universe → หุ้นใหญ่/ETF กลุ่มเดียวกัน สภาพคล่องสูง (เช่น KO–PEP) 2) ทดสอบ cointegration → EG p < 0.05 + KPSS ไม่ขัด (ไม่ reject = Watchlist ไม่ใช่เทรด) 3) กรอง half-life → 5–30 วัน (สั้นไป = noise, ยาวไป = ทุนจม) 4) ตั้งกฎ Z-score → entry ±2, exit |Z|<0.5, stop ±4 5) Backtest (train→OOS) → ประเมินบน train วัดผลบน out-of-sample; ดู Sharpe "หลังหักต้นทุน" 6) Paper trade 2-3 เดือน → แล้วค่อย size ≤ 1-2%/คู่ ด้วยเงินที่เสียได้
```

Evidence Checklist — หลักฐานแน่นแค่ไหน?

เกณฑ์	หลักฐานแข็ง ✓	สัญญาณเตือน ✗
Cointegration (p-value)	< 0.05 บน train (+ KPSS ไม่ขัด)	≥ 0.05 → ยัง*อ้าง* cointegration ไม่ได้ (Watchlist ไม่ใช่เทรด)
Half-life	~5–30 วัน	< 5 วัน (noise) หรือ > 30 วัน (ทุนจม)
Short ได้ทั้งสองขา	หุ้นใหญ่ borrow ถูก/ง่าย	หุ้นเล็ก หา borrow ไม่ได้/แพง
สภาพคล่อง	bid-ask แคบ, วอลุ่มพอ	spread กว้าง → ต้นทุนกินกำไร
Sharpe **out-of-sample** (หลังต้นทุน)	> 1 และ in-sample ไม่ทิ้งห่าง OOS มาก	> 3 (สงสัย overfit) หรือพังเมื่อทดสอบ OOS
ขนาด (sizing)	≤ 1–2% ของพอร์ต/คู่	ไม่มี stop / size ใหญ่

⚠️ **กับดัก multiple testing:** ถ้าไล่สกรีนหลายสิบคู่ จะมีบางคู่ "ผ่าน p<0.05" ด้วยความบังเอิญ (เหตุผลเดียวกับ √(2·ln N) ในบท 21) — อย่าเชื่อคู่เดียวที่ผ่าน ให้ดูว่ามันยังผ่านในช่วง OOS/หลาย regime ด้วย

> 🗂️ **ตัดสินเป็น "ระดับ" ไม่ใช่ ผ่าน/ตก — Evidence Tiers**
>
> อย่าใช้ p-value ตัวเดียวเป็นสวิตช์ ให้จัดคู่ที่สกรีนได้ลง 5 ระดับตามหลักฐานที่สะสม:
>
> 🔴 **Reject** — ไม่มีเหตุผลเศรษฐกิจว่าทำไมต้องวิ่งด้วยกัน / EG ไม่ reject *และ* KPSS ชี้ไม่ stationary / short ไม่ได้ → ทิ้ง
> 🟡 **Watchlist** — เหตุผลเศรษฐกิจแน่น แต่หลักฐานสถิติก้ำกึ่ง (p ≈ 0.05–0.15, sample สั้น) → เก็บข้อมูลเพิ่ม/ดู rolling ห้ามเทรด
> 🔵 **Research candidate** — EG reject + half-life ดี แต่ OOS Sharpe < 1 → paper trade / ปรับโมเดล
> 🟢 **Trade candidate** — หลักฐานครบทุกชั้น + OOS Sharpe > 1 → paper trade 2–3 เดือน
> 🟢 **Trade (size เล็ก)** — paper ตรงกับคาด ≥20–30 สัญญาณ → เงินจริง ≤1–2%/คู่
>
> กฎเหล็ก: ถ้า test ไม่ผ่านเพราะสงสัยว่า spec ผิด (lag ไม่พอ, structural break, β เปลี่ยนตามเวลา) → **แก้โมเดล ไม่ใช่ลด threshold จนคู่ผ่าน**
> การจัดอันดับใน Watchlist: ใช้ **ขนาด + ความเสถียร** (half-life, ความถี่ที่ spread ตัดผ่านค่าเฉลี่ย, β นิ่งข้าม rolling windows) — **อย่าจัดอันดับด้วย p-value** เพราะมันผสม effect size กับความยาวข้อมูล: คู่ข้อมูลยาวได้ p ต่ำทั้งที่ effect อ่อน ส่วนคู่เศรษฐกิจแน่นแต่ข้อมูลสั้นได้ p สูงทั้งที่น่าค้นกว่า (และ p=0.04 กับ 0.06 แทบไม่ต่างกันจริง)

เครื่องมือสถิติแต่ละตัว — ต้อง "significant" ไหม?

เครื่องมือ	ใช้ significance อย่างไร
Correlation	อย่าใช้ p เป็นด่าน — ดู**ขนาด + ความเสถียร** (r=0.08 ที่ p<0.001 ก็ไร้ประโยชน์เชิงเทรด)
ADF / PP[14][15]	**ต้อง reject** ถ้าจะ*อ้าง*ว่า spread stationary; ไม่ reject = ยังสรุปไม่ได้ (power ต่ำ)
KPSS[16]	null กลับด้าน (null = stationary) — ใช้ **cross-check** กับ ADF ให้หลักฐานสองทาง
Engle–Granger[13][18]	**ต้อง reject** ด้วย critical values เฉพาะ (ห้ามใช้ ADF ปกติบน residual)
Johansen[17]	หลักฐานสำคัญเมื่อขา > 2 หรือกังวลเรื่องทิศทาง regression
Shapiro–Wilk (normality)	**ไม่ใช่ด่านก่อน ADF/cointegration** — ตอบคนละคำถาม
Backtest P&L	ดู OOS + ต้นทุน + selection bias — ไม่ใช่ p-value ตัวเดียว

โค้ด: ทดสอบคู่หุ้นใน ~10 บรรทัด

```
import yfinance as yf, statsmodels.api as sm from statsmodels.tsa.stattools import coint px = yf.download(["KO","PEP"], start="2018-01-01", auto_adjust=True, progress=False)["Close"][["KO","PEP"]].dropna() a, b = px["KO"], px["PEP"] t, pval, _ = coint(a, b) # 1) cointegration (ใช้ coint() เท่านั้น — print("p-value:", round(pval, 4)) # critical values ถูกชุด, อย่า adfuller residual เอง) beta = sm.OLS(a, sm.add_constant(b)).fit().params.iloc[1] spread = a - beta * b # 2) spread = A - beta*B
```

โค้ดเต็ม (half-life + Z-score + backtest train→OOS มีต้นทุน+borrow + **สัญญาณวันนี้**) อยู่ที่ `code/pairs_trading_starter.py` — รัน `python pairs_trading_starter.py KO PEP` บรรทัด `[6]` สัญญาณวันนี้ คือสิ่งที่เอาไปจดใน paper-trade log ทุกวัน (รันทุกเย็นหลังตลาดปิด)

ตัวอย่างการตัดสินใจ (เดินจริง 1 คู่)

สมมติรันคู่ A–B แล้วได้:

- p-value (train) = **0.03** → ผ่าน (cointegrated) ✓
- half-life = **12 วัน** → อยู่ในช่วง 5–30 ✓
- Sharpe: in-sample 2.1 → **out-of-sample 0.8** (หลังต้นทุน 20bps + borrow) → *ก้ำกึ่ง* ⚠️
- ทั้งสองตัวเป็นหุ้นใหญ่ borrow ง่าย ✓

คำตัดสิน: โครงสร้างดี แต่ OOS Sharpe เพียง 0.8 — จัดเป็น 🔵 **Research candidate** (ไม่ใช่ Trade candidate) → paper trade ≥ 2–3 เดือน (≥20–30 สัญญาณ) ถ้าผลยังตรงค่อยเลื่อนระดับเป็น Trade size เล็กมาก; ถ้า OOS Sharpe ติดลบ → ลดเป็น Reject ทันที

Action Plan — 3 เดือนแรก (ไม่เสี่ยงเงินก่อน)

```
วันนี้ : รันโค้ดกับ 1 คู่ที่สนใจ (ใช้ Google Colab ก็ได้ ไม่ต้องลงอะไร) สัปดาห์ 1 : คัด
3–5 คู่ที่ผ่าน Checklist ครบ เดือน 1–3 : paper trade — รันโค้ดทุกเย็น จดสัญญาณ
[6] ลงชีต (ไม่ใช้เงินจริง) ทุกสัปดาห์ : re-test cointegration — ถ้า p-value โผล่
> 0.10 หรือ β เลื่อนเยอะ = ถอดคู่นั้น จบ 3 เดือน : มี ≥20–30 สัญญาณแล้วเทียบกับที่คาด
→ ตรง: size ≤1–2%/คู่ | ไม่ตรง: ทิ้ง/แก้
```

📝 **Paper trade ทำยังไง? (สำหรับมือใหม่)**

Paper trade = เทรดสมมติ ไม่ส่งคำสั่งจริง ไม่เสียเงิน — แค่จดว่า "ถ้าเทรดจริงจะได้/เสียเท่าไร"

ทำชีต (Google Sheets/Excel) คอลัมน์: วันที่ | คู่ | Z-score | Action (Long/Short/Exit) | ราคา A | ราคา B | P&L สมมติ
ลูปประจำวัน: รันสคริปต์หลังตลาดปิด → อ่านบรรทัด [6] สัญญาณวันนี้ → จดลงชีต → สิ้นเดือน รวม P&L ดูว่าตรงกับ backtest ไหม

หมายเหตุ: คู่ตัวอย่างเป็นหุ้นสหรัฐ การ short จริงต้องใช้โบรกเกอร์ที่รองรับ margin/short หุ้นสหรัฐ (บัญชีไทย long-only ทำไม่ได้) — แต่ช่วง paper trade **ไม่ต้องมีบัญชี** แค่จดสัญญาณ

Pitfalls — ทำ / ไม่ทำ

✓ ทำ	✗ ไม่ทำ
ทดสอบ cointegration ก่อนเสมอ	เชื่อ correlation สูงอย่างเดียว
ดู Sharpe "หลังหักต้นทุน"	ตื่นเต้นกับ Sharpe สูงๆ จาก backtest
คิดค่า borrow + spread + slippage	ลืมต้นทุนฝั่ง short
walk-forward / out-of-sample	random K-fold (data leakage)
paper trade ก่อน, size เล็ก	ลงเงินก้อนใหญ่ตั้งแต่คู่แรก
เริ่มหุ้นใหญ่ borrow ง่าย	ไล่หุ้นเล็กที่ "ส่วนต่างเยอะ" แต่ short ไม่ได้
re-test cointegration เป็นรอบ + ถอดคู่ที่พัง	พึ่ง stop ที่ |Z|>4 อย่างเดียวกัน structural break
มอง p-value เป็น "ระดับหลักฐาน" (Evidence Tiers)	ลด threshold จนคู่ผ่าน — แก้เกณฑ์แทนที่จะแก้โมเดล
กระจายหลายคู่ที่ไม่ซ้ำ sector/factor	ถือ 5 คู่ที่จริงๆ เป็น bet เดียวกัน

🧰 Starter Kit

ไม่อยากลงอะไร? เปิด `colab.research.google.com` → เซลล์แรกพิมพ์ `!pip install yfinance statsmodels` → วางโค้ดแล้วกด Run (ฟรี ไม่ต้องติดตั้ง Python ในเครื่อง)
ลงในเครื่อง: `pip install yfinance statsmodels pandas numpy`
ข้อมูลฟรี: Yahoo Finance (yfinance), Stooq — พอสำหรับงานรายวัน (ระวัง survivorship bias: ไม่เห็นหุ้นที่ถูกถอด/เจ๊ง → ผลดูดีเกินจริง)
อ่านต่อ: บทวิจารณ์รวม [4] (ภาพรวมวิธีทั้งหมด) และวิธี OU/PCA [5] ในรายการอ้างอิงด้านล่าง

📚 อ้างอิง (References) — Part V

งานวิจัย peer-reviewed ที่ใช้สนับสนุนกล่องหลักฐานในบท 18–21 (citation/ปี/วารสาร ตรวจสอบแล้ว) — ใช้เป็นจุดเริ่มอ่านต่อ ไม่ใช่คำแนะนำการลงทุน

1. Gatev, E., Goetzmann, W. N., & Rouwenhorst, K. G. (2006). "Pairs Trading: Performance of a Relative-Value Arbitrage Rule." *The Review of Financial Studies*, 19(3), 797–827. link
2. Do, B., & Faff, R. (2010). "Does Simple Pairs Trading Still Work?" *Financial Analysts Journal*, 66(4), 83–95. link
3. Do, B., & Faff, R. (2012). "Are Pairs Trading Profits Robust to Trading Costs?" *Journal of Financial Research*, 35(2), 261–287. link
4. Krauss, C. (2017). "Statistical Arbitrage Pairs Trading Strategies: Review and Outlook." *Journal of Economic Surveys*, 31(2), 513–545. link
5. Avellaneda, M., & Lee, J.-H. (2010). "Statistical Arbitrage in the US Equities Market." *Quantitative Finance*, 10(7), 761–782. link
6. Palomar, D. P. (2025). *Portfolio Optimization: Theory and Application*, Ch. 15 (Pairs Trading / Kalman). Cambridge University Press. link
7. Bailey, D., Borwein, J., López de Prado, M., & Zhu, Q. (2014). "Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-of-Sample Performance." *Notices of the AMS*, 61(5), 458–471. link
8. Bailey, D., & López de Prado, M. (2014). "The Deflated Sharpe Ratio: Correcting for Selection Bias, Backtest Overfitting, and Non-Normality." *The Journal of Portfolio Management*, 40(5), 94–107. link
9. Harvey, C. R., Liu, Y., & Zhu, H. (2016). "…and the Cross-Section of Expected Returns." *The Review of Financial Studies*, 29(1), 5–68. link
10. Gu, S., Kelly, B., & Xiu, D. (2020). "Empirical Asset Pricing via Machine Learning." *The Review of Financial Studies*, 33(5), 2223–2273. link
11. Krauss, C., Do, X. A., & Huck, N. (2017). "Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500." *European Journal of Operational Research*, 259(2), 689–702. link
12. Avramov, D., Cheng, S., & Metzker, L. (2023). "Machine Learning vs. Economic Restrictions: Evidence from Stock Return Predictability." *Management Science*, 69(5), 2587–2619. link
13. Engle, R. F., & Granger, C. W. J. (1987). "Co-integration and Error Correction: Representation, Estimation, and Testing." *Econometrica*, 55(2), 251–276. link
14. Dickey, D. A., & Fuller, W. A. (1979). "Distribution of the Estimators for Autoregressive Time Series with a Unit Root." *Journal of the American Statistical Association*, 74(366), 427–431. link

15. Phillips, P. C. B., & Perron, P. (1988). "Testing for a Unit Root in Time Series Regression." *Biometrika*, 75(2), 335–346. link

16. Kwiatkowski, D., Phillips, P. C. B., Schmidt, P., & Shin, Y. (1992). "Testing the Null Hypothesis of Stationarity Against the Alternative of a Unit Root." *Journal of Econometrics*, 54(1–3), 159–178. link

17. Johansen, S. (1991). "Estimation and Hypothesis Testing of Cointegration Vectors in Gaussian Vector Autoregressive Models." *Econometrica*, 59(6), 1551–1580. link

18. MacKinnon, J. G. (2010). "Critical Values for Cointegration Tests." *Queen's Economics Department Working Paper* No. 1227. link

วิธีอ่านตัวเลขในงานวิจัยเหล่านี้: ผลตอบแทน ~11%/ปี ของ [1] และ Sharpe จาก [5], [10] เป็นค่า *ก่อนหักต้นทุน (gross)* และมักเป็น out-of-sample บนทั้ง universe — **ผลจริงหลังต้นทุนของรายย่อยจะต่ำกว่านี้มาก** ให้ใช้เป็น "เพดานบน" ไม่ใช่ค่าที่คาดหวังได้จริง; ส่วนเกณฑ์ "Sharpe 5 vs 1.5" เป็น rule of thumb ไม่ใช่ผลพิสูจน์ใน [7]–[8]

จบ Part V

Part VI: Merger & Event Arbitrage