Ch. 03

Implementing Mean Reversion Strategies

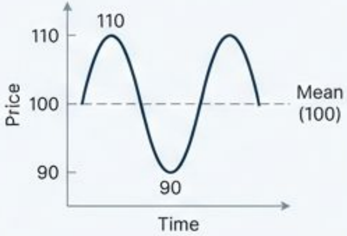
- 다양한 사전 학습 모델과 데이터셋을 제공하는 온라인 플랫폼

Ch. 02에서는…

평균회귀(Mean Reversion)와 공적분(Cointegration) 전략

1. 핵심 개념: 정상성 (Stationarity)

- 시간이 흘러도 데이터의 **평균과 변동 폭**이 일정하게 유지되는 성질
- **직관:** 가격이 100원을 중심으로 110원(고평가)이 되면 내려오고, 90원(저평가)이 되면 다시 올라오는 '복원력'
- **전략:** '너무 높으면 매도, 너무 낮으면 매수' (데이터가 정상성을 가질 때만 유효)



2. 단일 자산의 한계 vs 포트폴리오 해법

- **문제점:** 삼성전자와 같은 개별 주식은 장기적으로 우상향 (추세 존재 = **비정상성**)
- **해결책 (공적분):** 성격이 비슷한 두 자산(예: 코카콜라-펩시)을 엮어 분석
- 개별 자산은 제멋대로 움직여도, 두 자산 간의 **'비율'**이나 **'차이'**는 일정 범위를 유지함

3. 롱-숏(Long-Short) 페어 트레이딩 모델

- 두 자산의 가격 차이(Spread)가 정상성을 띤다면, 그 괴리를 이용해 수익 창출
- **실행 전략:** 스프레드 확대 → 고평가 자산 매도(Short) + 저평가 자산 매수(Long)

```
Spread = Price_A - (Hedge_Ratio * Price_B)
```

> **해석:** A와 B의 가격 차이(Spread)가 특정 범위 내에서 움직일 때, 통계적으로 유의미한 차익거래 기회 발생

4. 이론과 현실의 괴리 (Reality Check)

- **이론:** 가격은 언젠가 반드시 장기 평균으로 회귀하며 공적분 관계는 안정적임
- **현실:** 시장의 구조적 변화, 뉴스, 산업 트렌드 이동으로 인해 **'영구적인 평균'은 존재하지 않음**

결론: 맹목적인 믿음보다는 스프레드의 정상성이 깨지는 시점을 지속적으로 모니터링해야 함

실제로는...

실전 평균회귀 (Practical Mean Reversion)

완벽한 이론보다 빠른 대응: 공적분에서 수익실현까지

1. 핵심 철학 (Philosophy)

- **이론 (Theory):** 시계열의 완벽한 **정상성(Stationarity)** 요구. 장기 평균 회귀 필수.
- **실전 (Practice):** 진정한 공적분 불필요. **일시적 가격 왜곡(Distortion)** 포착.

단기적이거나 Seaonal 평균회귀 포착

"바닥에 멈추길 기다리지 않고,
***튀어 오르는 짧은 구간**만 취한다."*

2. 수학적 정의 (Definition)

$$Spread\_t = \log(Price\_t^{A}) - \beta * \log(Price\_t^{B})$$

해석: 두 자산의 스프레드가 발산하지 않고 평균으로 회귀하는 성질 이용

3. 실전 수익 모델 (Execution Examples)

Card A: 단기 과매도 (Short-term)

상황: 악재 없는 수급 꼬임 급락 (예: 삼성전자 -4%)

대응: 당일 기술적 반등 **+1~2%** 발생 시 **즉시 청산**

장기 균형 무시

Card B: 페어 트레이딩 (Pairs)

상황: 동종 업계 (삼성 vs 하이닉스) 괴리율 확대

대응: 며칠 내 **스프레드 축소** 가능성 확인 시 진입

완벽한 공적분 검정 불필요

Card C: 계절적 패턴 (Seasonal)

상황: 장 마감, 월말 리밸런싱 등 **특정 시간/조건**

대응: 전체 기간이 아닌, **특정 조건** 하에서만 발생하는 회귀 공략

4. 핵심 제약 조건: 반감기 (Half-life)

⚠ 아무리 완벽한 정상 시계열이라도 **회귀 속도가 느리면 무용지물.**
평균 회귀에 10년이 걸린다면 **트레이딩 전략으로서 가치 없음 (Not Profitable).**

선형 평균회귀의 한계

Theory & Translation

ORIGINAL TEXT

We also described a simple linear mean reversion strategy that simply 'scales' into an asset... It is not a very practical strategy due to the demand of unlimited buying power.

TRANSLATION 자산을 점진적으로 매수/매도하는 단순 선형회귀 전략

우리는 가격이 평균에서 벗어난 정도에 비례하여 진입 규모를 '조절(scales)'하는 단순 선형 전략을 설명했다. 이는 무한한 매수 여력을 요구하므로 비실용적이다.

The Trap & The Solution

The Logic (Code Block)

```
Position_Size = -k * (Current_Price - Mean_Price)
```

가격이 평균보다 낮을수록 매수량(Position)을 선형적으로 계속 늘림 (일명 '물타기')

지속적인 극소 단위 리밸런싱

Practical Example (Samsung Electronics)

예시: 삼성전자 (평균 100,000원)

↘ 98,000원 (-2%) ⟶ 10주 매수

↘ 95,000원 (-5%) ⟶ 30주 매수

↘ 90,000원 (-10%) ⟶ 60주 매수

- **Critique (Problem)**

  ❌ **치명적 단점:** 시장이 구조적으로 하락하면 무한한 자본이 필요하며 파산 위험.

NotebookLM

이번 장에서는 실용적인 대안인

1. 단순한 평균회귀 전략 볼린저 밴드
2. Scaling in 방식의 장단점
3. 헤지 비율과 평균 가격 추정을 위한 Kalman filter
4. 평균회귀 전략에서 데이터 오류가 초래할 수 있는 위험

을 다룰 예정

본론으로 들어가기 앞서 반드시 알아야할 사항

프로토타입 백테스트의 함정과 실전 보정

⚠ 1. 전방주시 편향 (Look-ahead Bias)

- 학습 목적의 단순화를 위해 **거래비용(Transaction Costs)** 및 **현실적 제약** 의도적 **배제**
- 변수 최적화(Optimization)와 백테스트에 **동일 데이터 사용** 오류 범함
- 미래 정보를 미리 알고 투자하는 것과 같아 성과가 **과대평가**됨

2. 평균회귀 전략의 치명적 취약점

- 평균회귀는 본질적으로 **이상치(Outlier)**에 베팅하는 구조
- 단순 **데이터 오류**를 **이례적 수익 기회**로 착각할 위험 극대화
- 실전에서는 **데이터 정제**(Cleaning) **실패**가 거대한 **손실**로 직결

3. 이론과 실전의 괴리 (Logic Flow)

```
# 잘못된 접근 (책의 예제 단순화)
Best_Parameter = Optimize(Data_A)
Backtest_Result = Run_Strategy(Best_Parameter, Data_A) # 동일 데이터 사용
```

↓

```
# 올바른 접근 (실전 구현)
Best_Parameter = Optimize(Data_Train)
Real_Result = Run_Strategy(Best_Parameter, Data_Test) - Transaction_Costs
```

해석: 검증 데이터 분리(Walk-forward)와 비용 차감이 없다면 백테스트 결과는 허상에 불과함

4. 결론 및 행동 지침

책의 소스 코드는 이해를 돕기 위한 **프로토타입**. 실제 트레이딩 적용 시 모든 **통계적 편향**과 **오류**를 직접 수정하는 작업 필수.

Trading Pairs Using Price Spreads, Log Price Spreads, or Ratios

가격 스프레드, 로그 가격 스프레드, 또는 비율을 이용한 페어 트레이딩

평균회귀 거래를 위한
포트폴리오 구성을 위해...

목표: 혼돈 속에서 질서 찾기

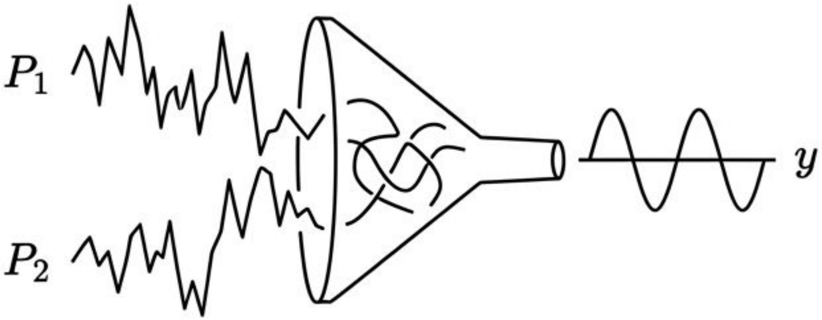
우리의 목표는 불규칙하게 움직이는 두 자산을 결합하여 **평균으로 회귀하는(Mean-Reverting)** 신호를 만드는 것입니다.

핵심 과제

비정상 시계열(Non-stationary) $P_1, P_2 \to y$
→ 정상 시계열(Stationary) y

세 가지 도구

1. Price Spread: 가격의 선형 결합
2. Log Price Spread: 수익률(Log)의 결합
3. Price Ratio: 단순 상대 가치 비율



1. Price Spread (가격 스프레드) 가격 차이가 평균으로 회귀

가장 직관적인 표준 접근 방식 Pair Trading의 경우

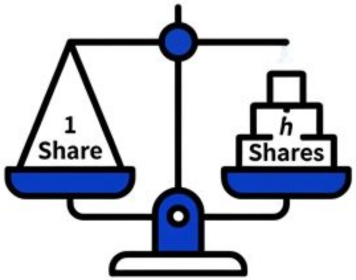
정상 시계열

$$\boxed{y = P_1 - h\ P_2}$$

선형회귀나 요한센으로 가중치인 hedge ration를 구함

의미: 자산 1을 1주 매수하고, 자산 2를 h주 매도함

- **직관적:** '주식 수(Shares)'를 기준으로 포트폴리오 구성
- **구성:** Long P1 + Short h P2
- **단점:** 자산 가격 레벨이 크게 변하면 스프레드 폭도 같이 커짐 (비정상성 위험)



2. Log Price Spread (로그 스프레드)

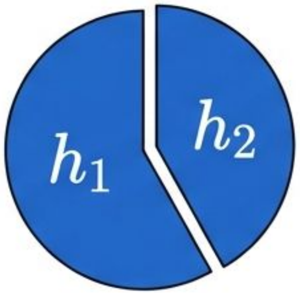
수익률(Returns) 관점의 접근 가격 시계열 대신 로그 가격이 공적분 관계에 있다고 가정

$$\log(q) = h_1 \log(y_1) + h_2 \log(y_2)$$

*의미: h는 주식 수가 아닌 **자본 비중(Capital Weights)**을 의미함

이론적 배경

- 로그 차분(Δ log)은 **수익률**($\Delta P/P$)과 유사
- **Johansen 공적분 테스트**와 이론적으로 가장 잘 부합함
- **해석**: 자산 간의 '비율' 관계가 일정하게 유지될 때 유리함



교재에서 로그 가격의 특성을 알아보기 위해 1차 차분을 취함

$$\log(q) = h_1 \log(y_1) + h_2 \log(y_2)$$

$$\Delta \log(q) = h_1 \Delta \log(y_1) + h_2 \Delta \log(y_2)$$

$$\Delta \log(x) \equiv \log(x(t)) - \log(x(t-1)) = \log(x(t)/x(t-1)) \approx \Delta x / x$$

x의 변화가 작을 때 성립되는 것을 참고하여

우변은 $h_1 \Delta y_1 / y_1 + h_2 \Delta y_2 / y_2$

즉, 주식 수익률 x 가중치 = 포트폴리오 수익률

“포트폴리오 수익률 = 각 종목 수익률의 가중합”

로그 차분은 거의 수익률과 같기 때문에, 그 식은 결국 각 종목 수익률의 가중합, 즉 포트폴리오 수익률을 의미한다.

🎯 예: 삼성전자 & SK하이닉스

총 1억 원 투자한다고 하자.

비중을 이렇게 정함:

- 삼성전자 50%
- SK하이닉스 -50% (공매도)

즉,

- $h_1 = 0.5$
- $h_2 = -0.5$

🔵 q는 무엇?

q는 이 포트폴리오의 총 가치야.

예를 들어 어느 날:

- 삼성전자 수익률 +2%
- SK하이닉스 수익률 +1%

포트폴리오 수익률:

$$0.5 \times 2\% + (-0.5) \times 1\% = 1\% - 0.5\% = 0.5\%$$

이게 q의 변화야.

🟢 처음 상태

- 삼성전자 5천만 원 매수
- SK하이닉스 5천만 원 공매도

순 투자금은 0이지만

실제로는:

- 공매도로 5천만 원 현금이 생김
- 삼성전자 매수에 5천만 원 사용

즉, 현금이 이미 포함돼 있음.

🔵 하루 후 가격 변화

- 삼성전자 +2%
- SK하이닉스 +1%

계산해보자:

삼성전자:

5천만 × 1.02 = 5,100만

SK하이닉스 숏 손실:

5천만 × 1.01 = 5,050만

순이익:

+50만 원

이 50만 원은 어디에 있지?

👉 현금으로 계좌에 쌓임.

Log Spread의 숨겨진 비용: 리밸런싱

로그 기반 모델은:

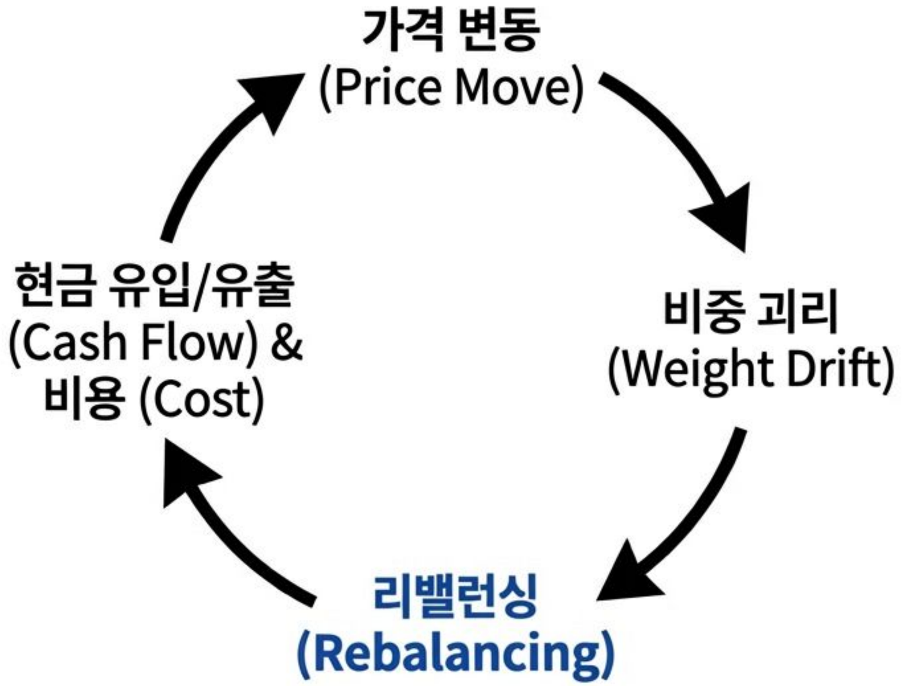
- “고정 비율(capital weight)”을 가정함.

하지만 실제 시장에서는:

- 가격이 매일 변함
- 비율이 자동으로 변함

그래서

👉 로그 기반으로 stationary를 유지하려면
👉 트레이더가 계속 비율을 맞춰줘야 함.



현금(Cash) 컴포넌트의 필수성

- 가격 변동 시 비중이 틀어짐
  → 지속적인 매매 필요
- 리밸런싱 과정에서 현금의 유입/유출 발생
- **결론:** 이론적 완벽함 vs 높은 **운용 비용**(Transaction Costs)

3. The Ratio (가격 비율)
실용적 대안

$$\$Ratio = P_1 / P_2$$

*의미: 자산 1의 가격이 **자산 2 대비 몇 배인**가 **(상대 가치)***

사용 조건
- 두 자산이 진정한 **공적분**(Cointegration) 관계가 아닐 때
- **구조적 변화**(Structural Break)가 존재할 때

특징
- 복잡한 **헤지 비율**(h) 산출 불필요
- 고정된 h 대신 **상대적 고평가/저평가**에 집중

Ratio의 강점: 스케일 불변성 (Scale Invariance)

시나리오: 자산 A ($10 → 100), 자산 B (5 → 50)

	구분	초기 상태	가격 급등 후	판정
1	Price Spread ($A - 2B$)	$10 - 10 = 0	$100 - 100 = 0	유지 (이상적)
2	Price Spread ($A - B$)	$10 - 5 = 5	$100 - 50 = 50	실패 (비정상 시계열)
3	Ratio (A / B)	$10 / 5 = 2	$100 / 50 = 2	성공 (정상 시계열)

인사이트: 성장주나 인플레이션 환경에서는 **Ratio**가 더 강력한 정상성을 가짐.

두 자산이 실제로는 공적분 관계가 아니지만 단기적으로 스프레드가 평균회귀한다고 믿는다면, 가격 스프레드나 로그 가격 스프레드보다 비율을 지표로 사용하는 것이 더 잘 작동

NotebookLM

특수 사례: 통화

1️⃣ 통화 페어는 본질적으로 "ratio"다

예: **EUR.GBP**

이건 "유로를 파운드로 나눈 값"이야.
수학적으로:

$$\mathrm{EUR.GBP} = \frac{\mathrm{EUR.USD}}{\mathrm{GBP.USD}}$$

즉, 통화 페어는 **가격 비율**이야. 그래서 환율 평균회귀 전략은 자연스럽게 **ratio 기반**이 된다.

📌 예시

- EUR.USD = 1.10
- GBP.USD = 1.25

$$\mathrm{EUR.GBP} = 1.10/1.25 = 0.88$$

이 값이 평균에서 많이 벗어나면 되돌림을 노리는 게 평균회귀 전략이야.

2️⃣ 그런데 어떤 통화쌍은 직접 거래가 안 된다

예: **MXN.NOK** (멕시코 페소 / 노르웨이 크로네)

많은 브로커는 이 "직접 환율"을 제공하지 않을 수 있어.
그 대신 우리는 이렇게 만들어야 해:

$$\mathrm{MXN.NOK} = \frac{\mathrm{USD.NOK}}{\mathrm{USD.MXN}}$$

즉, 두 개의 USD 기반 환율을 조합해서 "비율"로 만들어.

5️⃣ 그런데 두 방법은 완전히 같지 않다

중요한 차이 👇

- **MXN.NOK를 직접 거래**하면
  → 손익(P&L)은 NOK로만 계산됨.
- **USD.NOK와 USD.MXN을 조합해서 거래**하면
  → 손익이 NOK와 MXN 두 통화로 나뉘어 발생함.

즉, 구조는 비슷하지만
실제 손익 통화와 리스크 노출이 달라.

📊 예시 숫자

초기:

- USD.NOK = 10
- USD.MXN = 20

그러면:

$$10/20 = 0.50$$

📈 환율 변화

- USD.NOK → 11
- USD.MXN → 20 (변화 없음)

그러면:

$$11/20 = 0.55$$

Case Study: Gold (GLD) vs. Oil (USO)

GLD	USO

가설 (Hypothesis)
유가 상승 → 인플레이션 상승 → 금 가격 상승 (양의 상관관계)

현실 (Reality)
- 장기적 공적분(Cointegration) 관계 **미약함**
- 하지만 단기적 평균 회귀(Mean Reversion) 경향은 존재 가능

실험 질문
동일한 전략을 **Spread**, **Log Spread**, **Ratio**로 각각 실행 시 성과는?

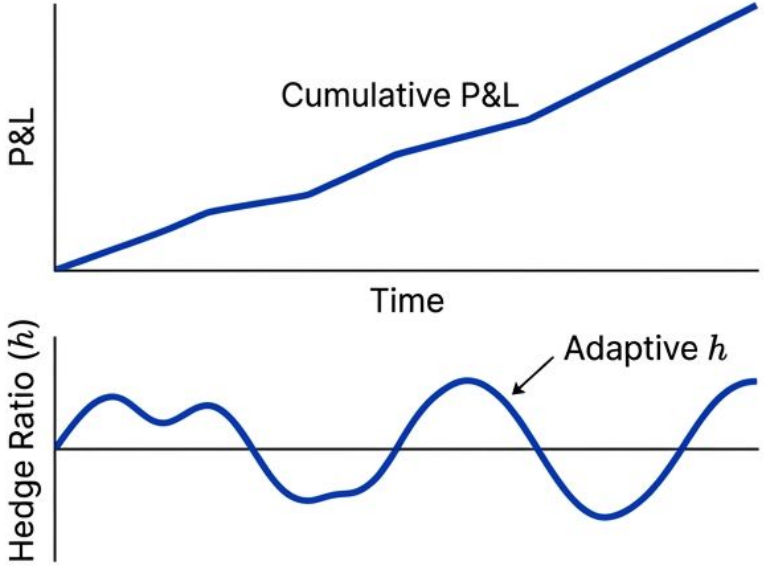
전략 1: Dynamic Price Spread (적응형 스프레드)

실행 방법

- 매일 최근 20일 데이터로 헤지 비율(h) **재계산(Rolling Regression)**
- 신호: $Z-Score 역매매

결과

- **APR (연수익률): 10.9%**
- **Sharpe Ratio: 0.59**



평가: Best Performance. 공적분이 약한 관계에서는 h를 지속적으로 수정해주는 것이 유리함.

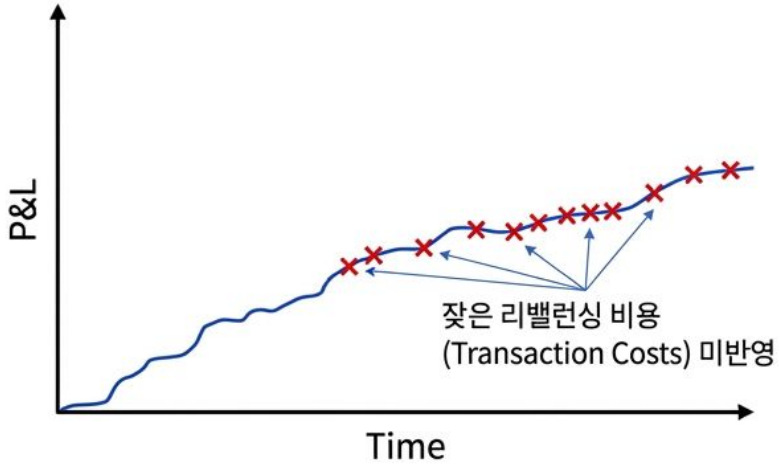
전략 2: Log Price Spread (로그 스프레드)

실행 방법

- 로그 가격 회귀분석 (Lookback 20일)
- 자본 비중(Weights) 기반 트레이딩

결과

- **APR**: **9%** (Spread 대비 저조)
- **Sharpe Ratio**: **0.5**



치명적 단점: 이론적 엄밀함이 높은 운용 비용을 초래.

전략 3: The Ratio (단순 비율)

- **실행 방법**
  - 단순 비율 (GLD/USO)의 볼린저 밴드 매매
  - 자금 배분: 롱/숏 동일 금액
- **결과**
  - **APR: 음수 (Negative)**

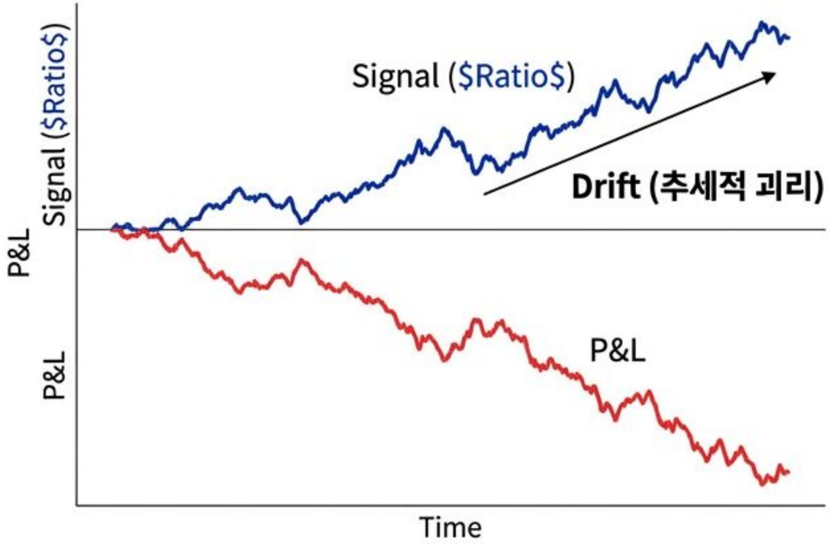
> “이 비율은 평균으로 돌아올 것이다”

그래서:

- 비율이 높아지면 숏
- 비율이 낮아지면 롱

그런데 문제가 생김 👇

비율이 계속 추세적으로 상승하면
👉 숏 포지션이 계속 손실



실패 원인: GLD와 USO는 구조적 관계가 불안정함. 단순 비율은 이를 보정하지 못함.

NotebookLM

종합 비교 분석

구분	Price Spread	Log Spread	Ratio
기본 개념	주식 수 ($Shares$)	자본 비중 ($Weights$)	상대 가치 ($Relative$)
정상성 유지	가격 급등 시 취약	스케일 변화에 강함	스케일 변화에 강함
운용 비용	낮음	높음 (리밸런싱)	낮음
적합 대상	일반적 주식/ETF	이론적 엄밀성 필요 시	성장주, FX, 이종 자산
GLD/USO 성과	**Best (Dynamic)**	Medium	**Fail**

결론: 언제 무엇을 써야 하는가?

1. 적응형 Price Spread
(Dynamic Hedge Ratio)

- 관계가 느슨한 원자재/ETF
- 시간이 따라 변하는 관계
- 예: **GLD vs USO**

2. Ratio
(단순 비율)

- 장기적으로 함께 성장하는 종목
- 가격 레벨 차이가 커지는 경우
- 예: **통화(FX), 성장주**

3. Log Spread
(로그 스프레드)

- 엄격한 이론적 공적분 검증 완료 시
- 비용 감내 가능 시

Bollinger Bands

선형 평균회귀의 한계와 볼린저 밴드

1. 선형 전략(Linear Strategy)의 치명적 결함

- 가격 이탈(Deviation)에 비례하여 투자 단위를 계속 늘리는 방식.
- 이론적으로는 우수하나 현실적으로는 파산 위험 존재.
- 가격이 끝없이 하락할 경우, 평균으로 회귀하기 전까지 무한대의 자본이 필요함.
- 소위 '물타기'가 무제한으로 지속되는 구조 (예: -10%, -30%, -60% 하락 시 계속 매수).

2. 현실적 대안: 볼린저 밴드(Bollinger Bands)

- 평균에서 '조금' 벗어난 노이즈는 무시.
- 사전에 정의된 임계값(Threshold) 이상 크게 벗어났을 때만 진입.
- 최대 투자 규모를 사전에 확정할 수 있어 리스크 통제 가능.
- 데이터 스누핑(Data-snooping) 편향이 적고 직관적임.

3. 전략의 핵심 차이

구분	선형 전략 (Linear)	볼린저 밴드 (Bollinger)
진입 시점	모든 이탈 구간	특정 표준편차(σ) 돌파 시
포지션 크기	이탈폭에 비례 (가변)	**0 또는 1 유닛 (고정)**
자본 요구	예측 불가능 (무한대)	**예측 가능 (제한적)**

트레이딩 로직과 진입·청산 메커니즘

1. 핵심 지표: Z-Score 산출

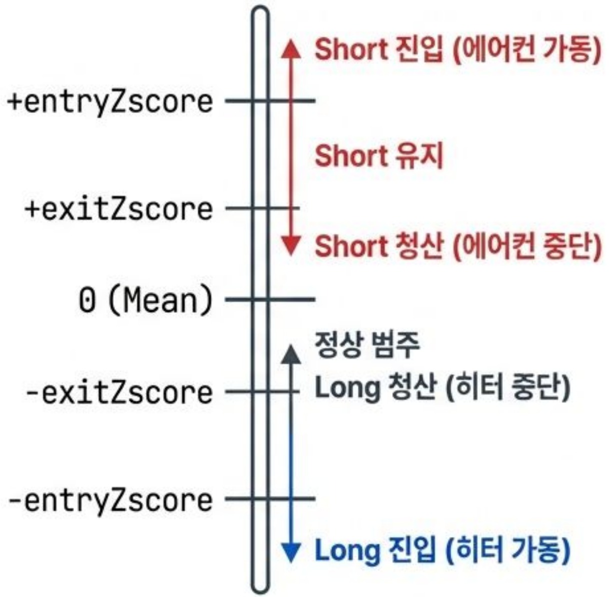
```
zScore = (Price - movingAvg(Price, lookback)) ./
   movingStd(Price, lookback);
% 현재 가격이 이동평균으로부터 몇 표준편차만큼 떨어져 있는지 계산
```

- `lookback`: 이동평균과 표준편차를 계산하는 과거 기간 (자유 파라미터).
- `zScore`: 진입과 청산을 결정하는 정규화된 시그널.

2. 트레이딩 규칙 (Rules)

- **Long 진입**: `zScore < -entryZscore` (가격이 평균보다 과도하게 하락, 예: -2σ).
- **Short 진입**: `zScore > entryZscore` (가격이 평균보다 과도하게 상승, 예: $+2\sigma$).
- **청산 (Exit)**: `zScore`가 `exitZscore` 범위 내로 복귀할 때.
- **조건**: 반드시 `|exitZscore| < |entryZscore|` 여야 함.

3. 직관적 이해: 온도계 모델



핵심: **'충분히 멀어졌을 때'** 진입하고,
'정상 범주로 돌아왔을 때' 청산한다.

자금 관리와 파라미터 최적화

1. 이진(Binary) 포지션 관리의 이점

- 복잡한 스케일링 없이 오직 0(대기), 1(매수), -1(매도) 유닛만 보유.
- 최대 손실폭과 필요 자본을 명확히 계산 가능 (자본 배분 효율성 극대화).
- 신호가 없는 날은 전일 포지션을 유지 (fillMissingData 함수 활용).

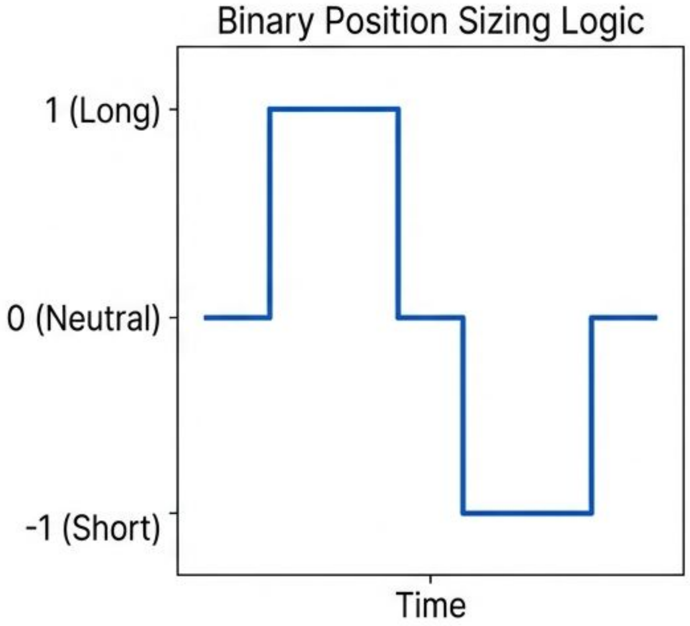
2. 파라미터 민감도와 트레이드 오프

- **공격적 설정** (짧은 Lookback, 낮은 Z-score):
  - 보유 기간 단축, 거래 빈도 증가.
  - 수익 기회 확대 vs 거래 비용/슬리피지 증가.
- **보수적 설정** (긴 Lookback, 높은 Z-score):
  - 거래 횟수 감소, 신뢰도 높은 구간만 진입.
  - 기회비용 발생 가능성.

3. 성과 검증 (GLD-USO 예제)

```
numUnitsLong(longsEntry) = 1;   % 진입 신호 시 1 유닛
numUnitsLong(longsExit) = 0;    % 청산 신호 시 0 유닛
```

- **결과**: 연환산 수익률 17.8%, 샤프 비율 0.96
- 단순한 로직 변경(선형 → 밴드)만으로도 위험 조정 수익률(Sharpe Ratio) 대폭 개선.



Does Scaling-in Work?

스케일링 인(Scaling-in)의 딜레마: 수학적 최적화 vs 현실적 생존

STRATEGY BRIEF: Mean Reversion Tactics

The Theoretical World

1. 직관과 정의 (Intuition)

- **스케일링 인(Averaging-in):** 평균회귀 전략에서 가격이 이격될수록 포지션을 점진적으로 확대
- **목적:** 평균 단가 개선 및 대규모 진입 시 **시장 충격(Market Impact) 완화**
- **가정:** '가격은 결국 평균으로 회귀한다'는 믿음

2. 이론적 검증: Mathematical Reality

연구 결과 (Schoenberg & Corwin, 2010): 단일 진입(All-in)이 분할 진입보다 기대 수익(EV) 면에서 항상 우월함.

```
# 가정: 자금 2계약, 목표가 F (F > L1 > L2)
# 전략 I (L1 All-in): 현재가 진입 매수
E[Profit_I] = 2(F - L1)

# 전략 II (L2 All-in): 바닥 대기 후 전량 매수
E[Profit_II] = 2p(F - L2)

# 전략 III (Average-in): 타협안 (분할 매수)
E[Profit_III] = (F - L1) + p(F - L2)
```

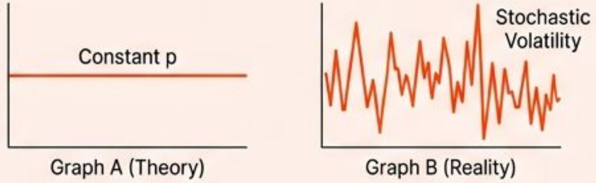
결론: 확률 p가 임계값보다 낮으면 **전략 I**이, 높으면 **전략 II**가 항상 최적. 분할 매수는 수학적으로 **'이도 저도 아닌'** 차선책.

The Real World

3. 현실적 반론: 왜 여전히 사용하는가? (Practical Robustness)

이른적 열위에도 불구하고 **불확실성(Uncertainty)** 때문에 실전에서는 필수적임.

- **변동성의 변화:** 수학적 모델은 확률 p를 상수로 가정하나, 현실의 변동성은 끊임없이 변함.



- **샤프 지수(Sharpe Ratio) 개선:** 단순 수익금은 낮을 수 있으나, 분산 효과로 낮을수 있으나, 분산 효과로 **변동성(Volatility)을 축소하여** 위험 조정 수익률 상승.
- **표본 외(Out-of-Sample) 강건성:**
  - In-Sample (과거): All-in 유리 (최적점 확인 가능)
  - Out-of-Sample (미래): 구조적 변화 대응을 위해 분할 진입이 생존에 유리

Summary: 백테스트(In-sample)에서는 'All-in'이 최고의 수익을 내지만, 불확실한 미래(Out-of-sample)의 생존과 샤프 지수 방어를 위해 'Scaling-in'은 여전히 필수적인 전술이다.

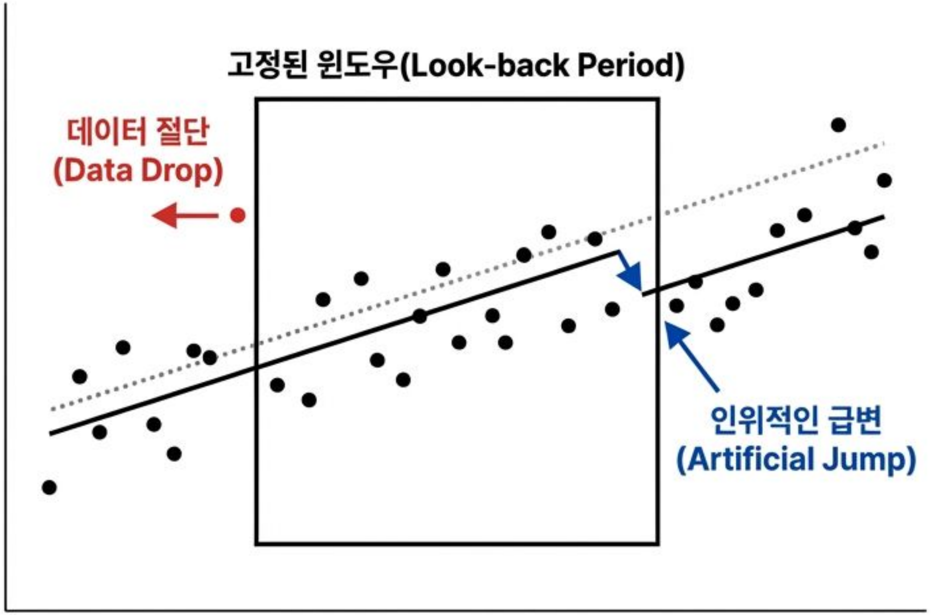
NotebookLM

Kalman Filter

Dynamic Linear Regression
(칼만 필터: 동적 선형 회귀)

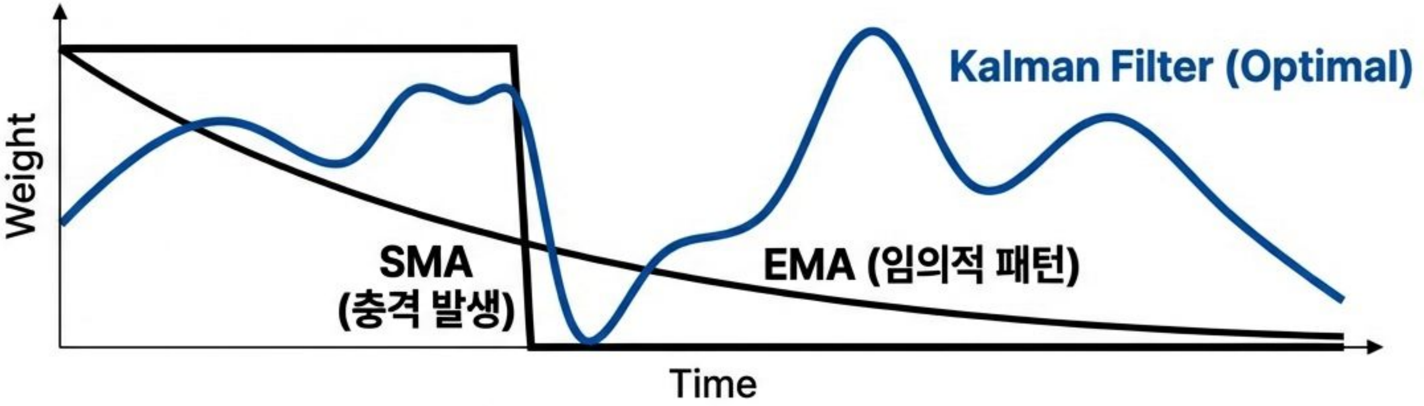
정적 모델의 한계를 넘어서는
퀀트 트레이딩 전략

정적 회귀분석(OLS)의 치명적 한계



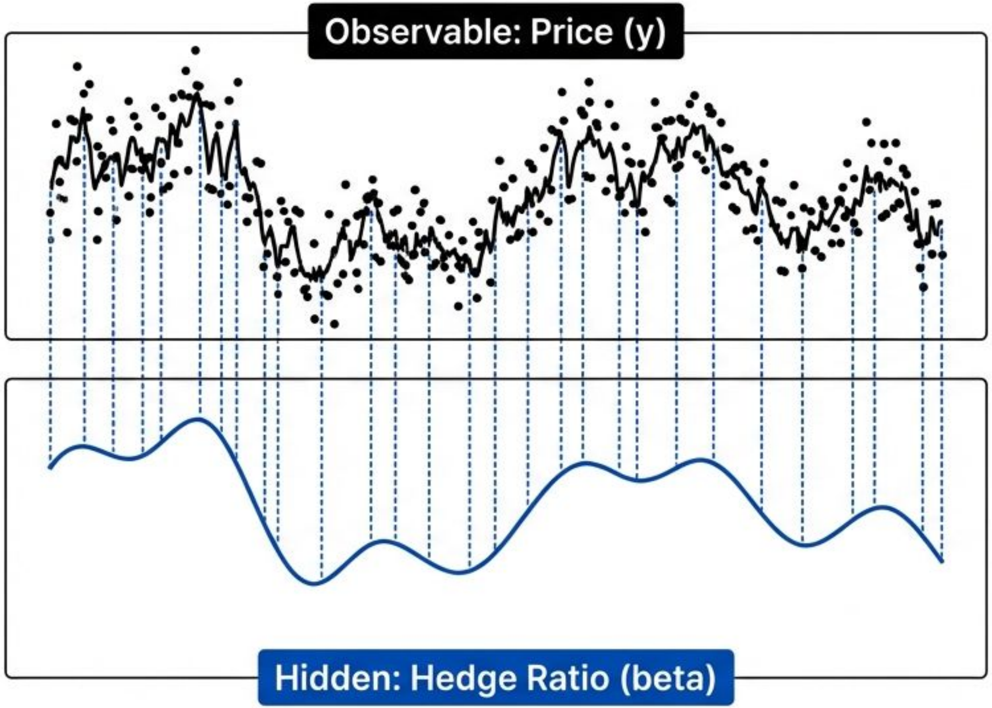
- 공적분(Cointegration)은 이상적 개념. 실제 데이터는 끊임없이 변함.
- **고정된 윈도우(Look-back Period)**의 문제점.
- 오래된 데이터의 삭제가 헤지 비율(Hedge Ratio)에 **인위적인 급변(Jump)**을 초래.
- 시장 변화가 아닌, 데이터 윈도우 이동에 의한 왜곡.

가중치(Weighting)의 딜레마와 해결책



- **단순 이동평균(MA):** 데이터 이탈 시 충격 발생.
- **지수 이동평균(EMA):** 가중치 감소 패턴이 임의적(Arbitrary)임.
- **Kalman Filter**: 데이터 자체의 노이즈를 기반으로 **최적의 가중치를 동적으로 산출**.
- 임의의 절단점(Cutoff Point) 없이 모든 데이터를 효율적으로 활용.

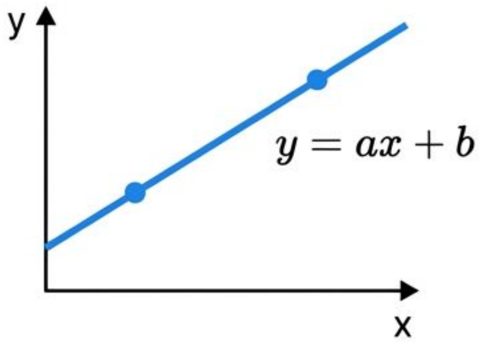
핵심 직관: 관측 변수 vs 숨겨진 변수



- 우리는 가격을 보지만, 관계를 거래한다.
- **관측 변수 (Observable):** 눈에 보이는 가격 데이터 (예: GLD, USO).
- **숨겨진 변수 (Hidden):*** 알고 싶은 관계값 (예: Hedge Ratio beta).
- **목표:** 관측된 가격을 통해 숨겨진 beta를 매일 최적으로 역추적.

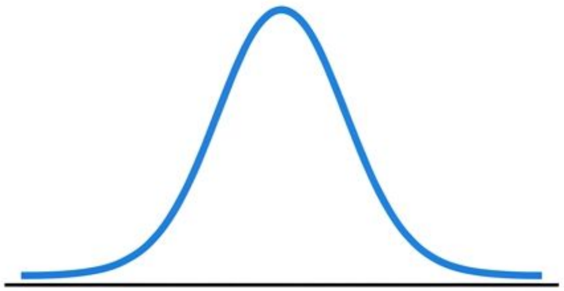
두 가지 기본 가정 (Linearity)

1. 선형성 (Linearity)



1. **관측 방정식:** 가격 = (헤지 비율 x 다른 자산 가격) + 노이즈.
2. **상태 방정식:** 오늘의 헤지 비율 = 어제의 헤지 비율 + 노이즈.

2. 정규분포 (Gaussian)



가정: 모든 노이즈는 **가우시안 분포(정규분포)**를 따름.

평균과 분산만으로 전체 분포 설명 가능.

수학적 프레임워크 (The Equations)

Measurement Equation

$$y(t) = x(t)\,\beta(t) + \varepsilon(t)$$

해석: 가격 y는 x와의 선형 관계(beta)에 오차를 더한 값.

State Transition Equation

$$\beta(t) = \beta(t-1) + \omega(t)$$

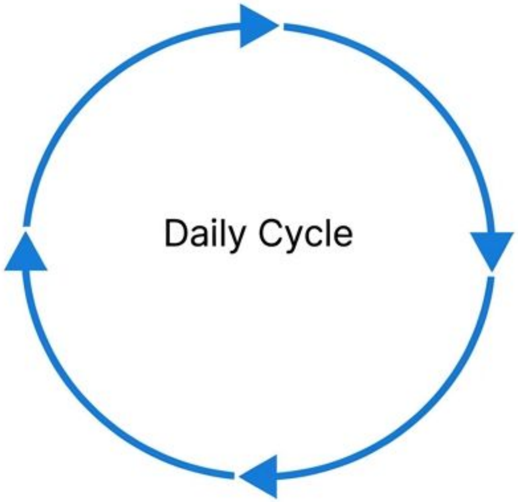
해석: 오늘의 관계(beta)는 어제와 비슷하나 미세하게 변함.

Note: beta는 벡터로, 기울기(Slope)와 절편(Intercept)을 모두 포함.

재귀적 루프 (The Recursive Loop)

1. 예측 (Predict)

어제 정보를 바탕으로 오늘 상태(β) 추정



2. 관측 (Observe)

실제 새로운 가격 데이터(y) 확인

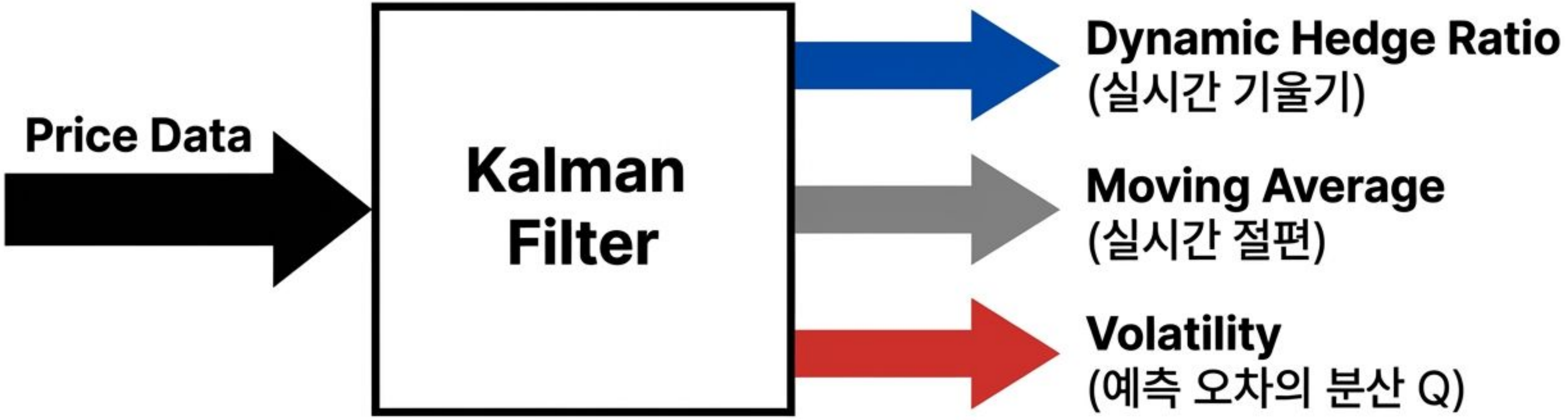
3. 업데이트 (Update)

예측과 실제의 차이(오차)를 반영해 β 수정

이 과정을 매일 반복하며 **평균제곱오차(MSE)를 최소화**.

NotebookLM

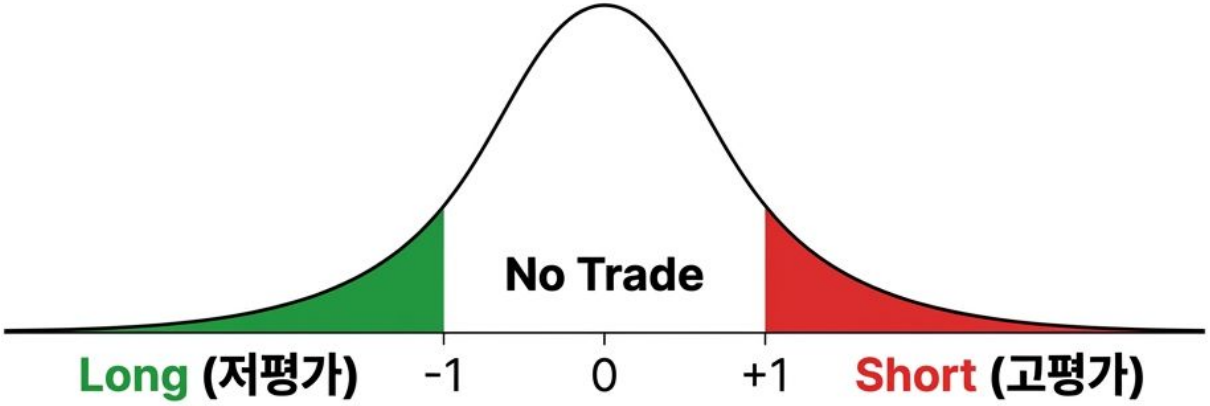
단 하나의 알고리즘, 세 가지 출력



- 별도의 이동평균선이나 볼린저 밴드 계산이 **불필요함**.
- 동적 헤지 비율뿐만 아니라 스프레드의 평균과 표준편차를 동시에 얻는다.

NotebookLM

신호 생성: Z-Score 접근법



예측 오차 $e(t)$: 실제 가격 - 예측 가격 (스프레드 편차).

예측 분산 $Q(t)$: 오차의 허용 범위 (표준편차 σ의 제곱).

진입 신호: $e(t) < -\sqrt{Q(t)}$ (**저평가 매수**).

청산 신호: $e(t) > -\sqrt{Q(t)}$ (**평균 회귀**).

트레이딩 로직 구현 (Implementation)

```
# Long Entry (너무 싸다)
if e(t) < -sqrt(Q(t)):
    Buy EWC, Sell EWA (Ratio: beta)

# Long Exit (평균 회귀)
if e(t) > -sqrt(Q(t)):
    Close Position

# Short Entry (너무 비싸다)
if e(t) > sqrt(Q(t)):
    Sell EWC, Buy EWA (Ratio: beta)
```

결과: EWA-EWC 백테스트 결과, **APR 26.2%, Sharpe 2.4** 달성.

칼만 필터 전략의 우위

No Look-back

데이터 절단으로 인한 인위적 충격 제거.

All-in-One

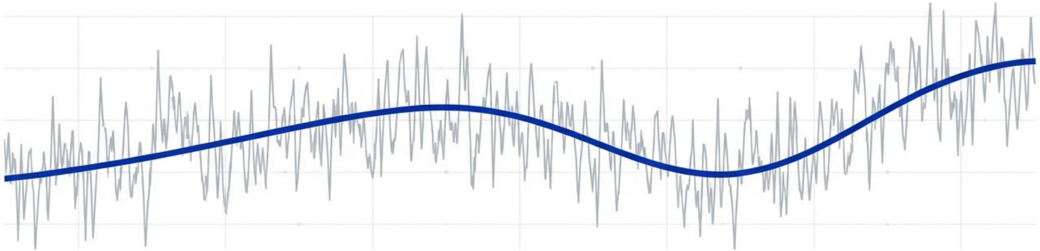
헤지 비율, 평균, 변동성 동시 추정으로 모델 단순화.

Adaptive

시장 구조 변화에 빠르게, 그러나 부드럽게 적응.

가우시안 가정 하에서 오차를 최소화하는 **최적 추정량(Optimal)**.

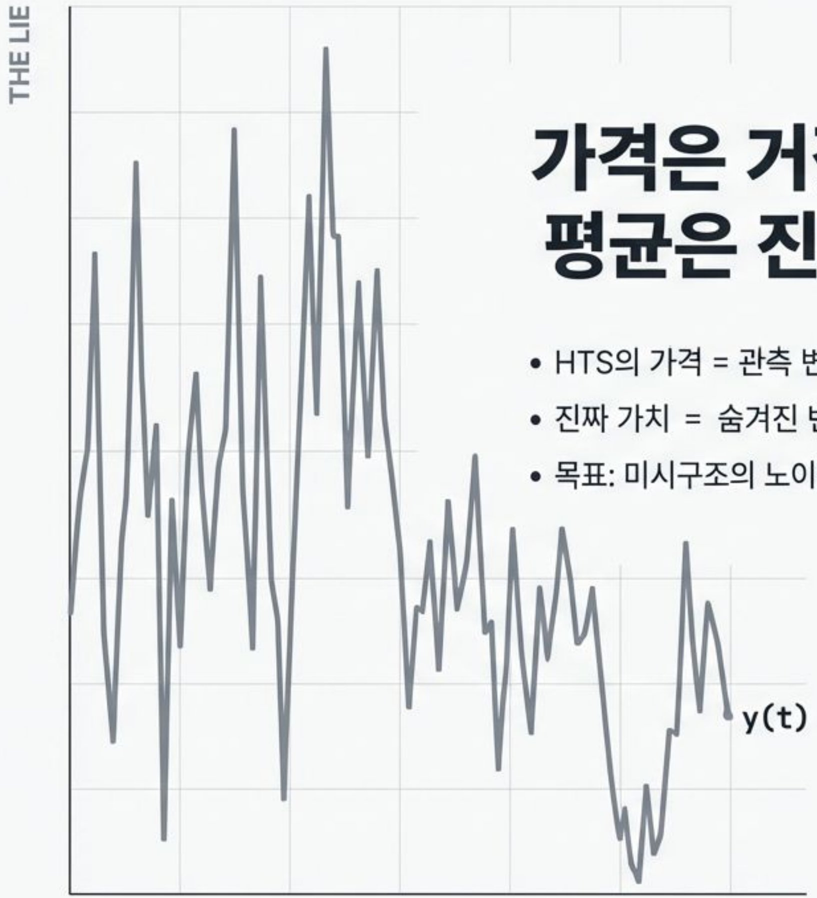
마켓메이킹을 위한 칼만 필터



Kalman Filter for Market Making

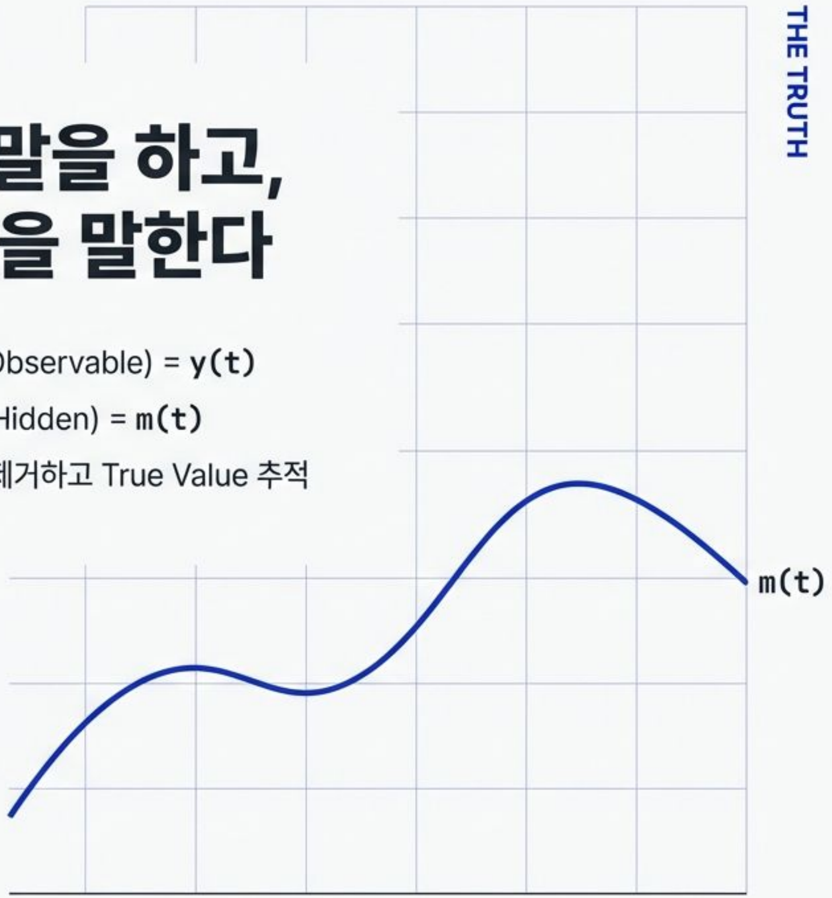
단일 자산 평균회귀 전략의 핵심 엔진

PRESENTER: [PRESENTER NAME]



가격은 거짓말을 하고, 평균은 진실을 말한다

- HTS의 가격 = 관측 변수 (Observable) = **y(t)**
- 진짜 가치 = 숨겨진 변수 (Hidden) = **m(t)**
- 목표: 미시구조의 노이즈를 제거하고 True Value 추적



세상을 바라보는 두 개의 방정식

MEASUREMENT (관측)

$$y(t) = m(t) + \varepsilon(t)$$

오늘의 가격 = 진짜 평균 + 일시적 노이즈

STATE TRANSITION (전이)

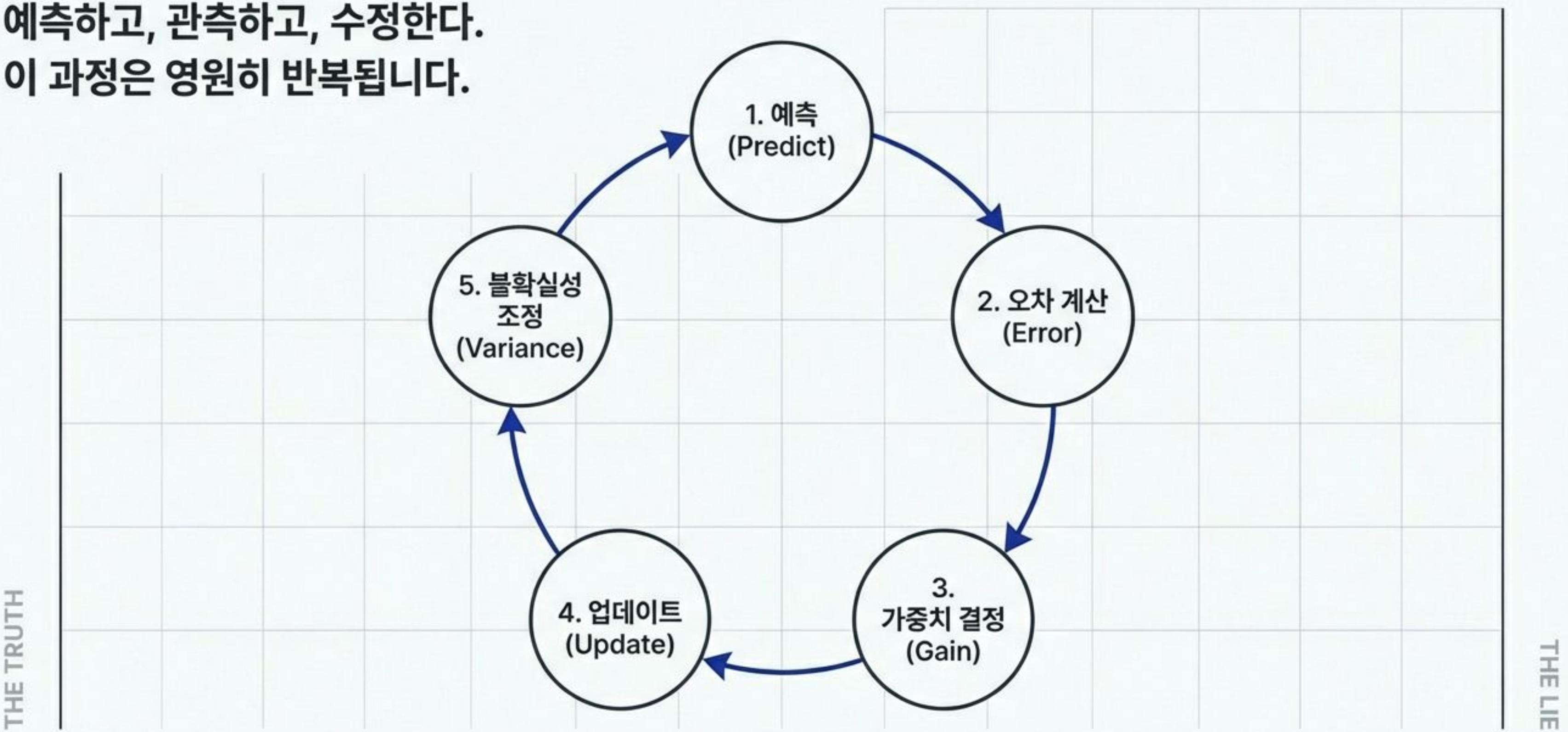
$$m(t) = m(t-1) + \omega(t-1)$$

진짜 평균 = 어제 평균 + 구조적 변화

평균은 상수가 아니라 동적인 변수(Time-varying Variable)입니다.

끊임없이 순환하는 추정 엔진

예측하고, 관측하고, 수정한다.
이 과정은 영원히 반복됩니다.



Step 1: 예측 (Prediction)

$$m(t|t-1) = m(t-1|t-1)$$

시장이 열리기 전, 우리는 어제의 평균을 오늘로 투영합니다.
시간이 흐름에 따라 상태 전이 오차만큼 분산은 커집니다.

Step 2: 괴리 (Innovation)

$$e(t) = y(t) - m(t|t-1)$$

관측된 가격과 예측된 평균의 충돌.
이 '오차'는 모델을 수정하기 위한
가장 귀중한 연료입니다.

Step 3: 신뢰의 균형 (Kalman Gain)

$$K(t) = \frac{R(t|t-1)}{R(t|t-1) + Ve}$$

K의 의미: 오늘 들어온 데이터를 얼마나 믿을 것인가?

- Noise (Ve) High -> Trust Model (K Low)
- Noise (Ve) Low -> Trust Data (K High)

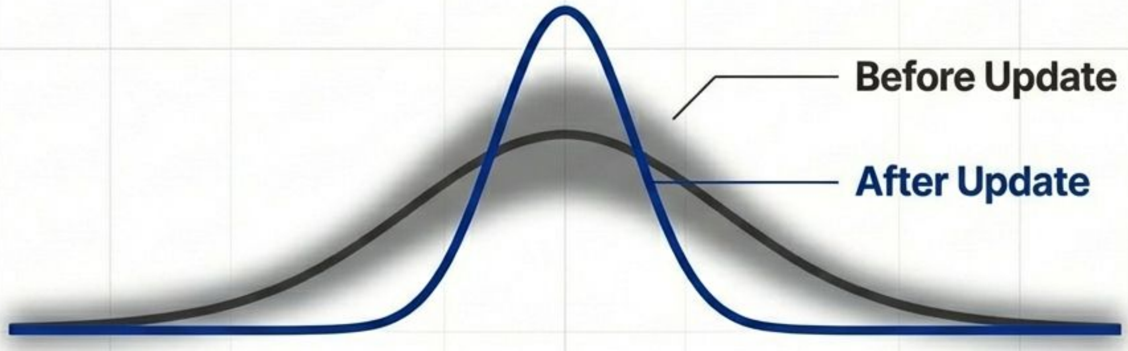
Step 4: 수정 (Update)

$$m(t|t) = m(t|t-1) + K(t) * e(t)$$

새로운 평균 = 기존 예측 + (신뢰도 K × 오차)
급격한 가격 변동에도 평균은
부드럽게(Smooth) 따라갑니다.

Step 5: 불확실성의 조정 (Variance Update)

$$R(t|t) = (1 - K(t)) * R(t|t-1)$$



데이터를 관측했으므로 불확실성(R)은 줄어듭니다.
시스템은 자신의 추정값이 얼마나 정확한지 스스로 인지합니다.

모든 틱(Tick)이 동등하지 않다

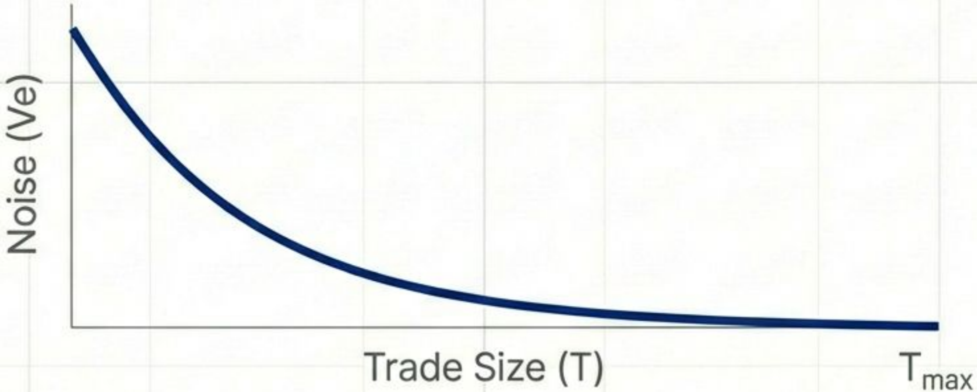
1 Share Trade
Noise (Ve) High

1,000,000 Share Trade
Noise (Ve) Low

Market Maker의 딜레마: 이것은 노이즈인가, 정보인가?
1주 거래는 무시하고, 대량 거래는 신뢰해야 합니다.
결론: 측정 오차 Ve는 상수가 아니라 시간 t 에 따른 함수여야 합니다.

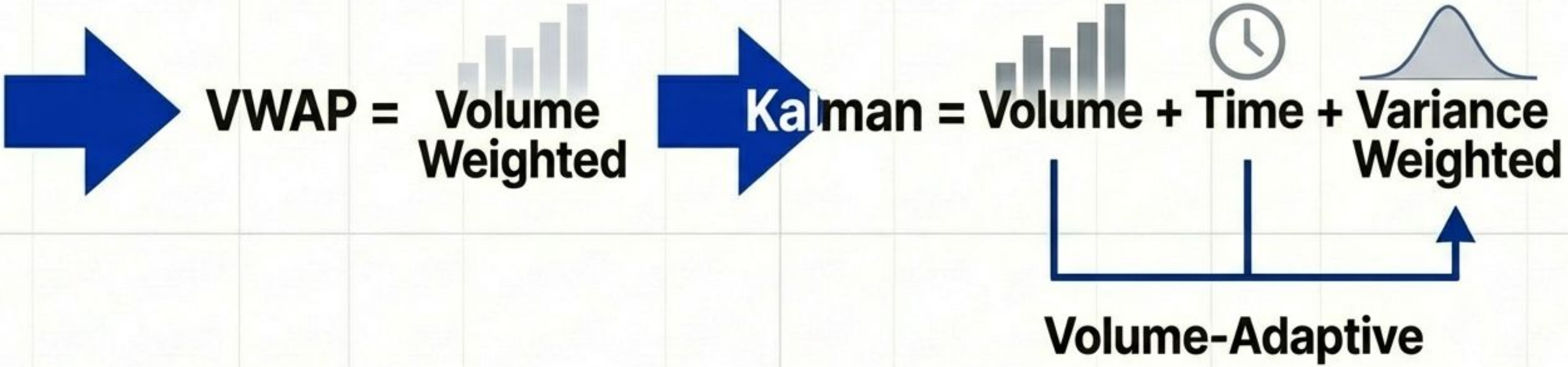
거래량이 곧 신뢰도다 (Volume-Adaptive Noise)

$$Ve(t) = R(t|t-1) - \left(\frac{T}{T_{max}}\right)^2 * R(t|t-1)$$



- $T \approx T_{max}$ (대량 거래): $Ve \rightarrow 0$, $K \rightarrow 1$ (가격 즉시 반영)
- $T \ll T_{max}$ (소량 거래): $Ve \rightarrow$ High, $K \rightarrow 0$ (노이즈 무시)

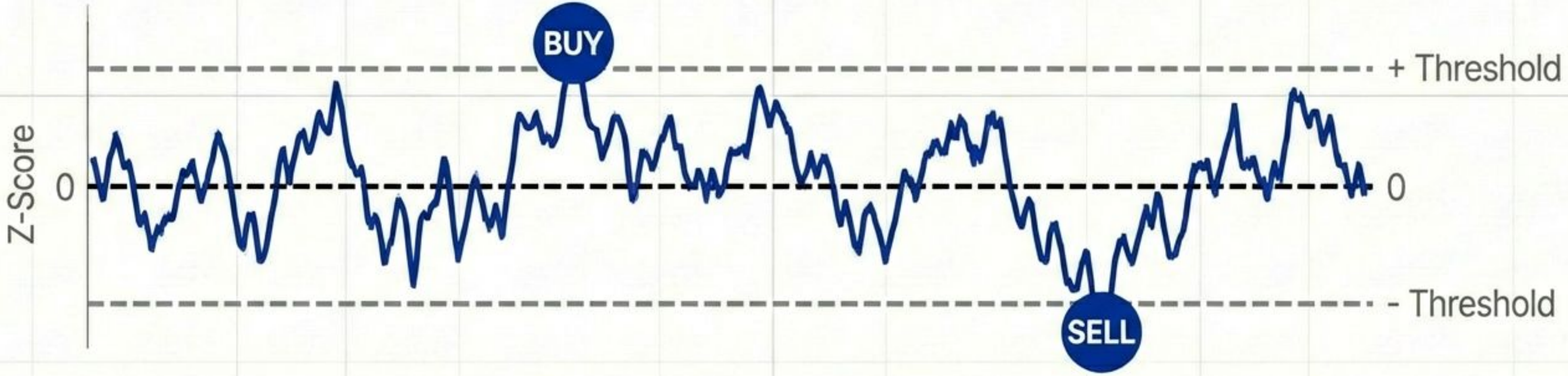
칼만 필터와 VWAP의 연결 고리



- 거래량에 따라 Ve를 조정하면, 칼만 필터는 **'동적 VWAP'**처럼 작동합니다.
- 단순한 이동평균보다 **시장 미시구조(Microstructure)** 정보를 훨씬 더 정교하게 반영합니다.

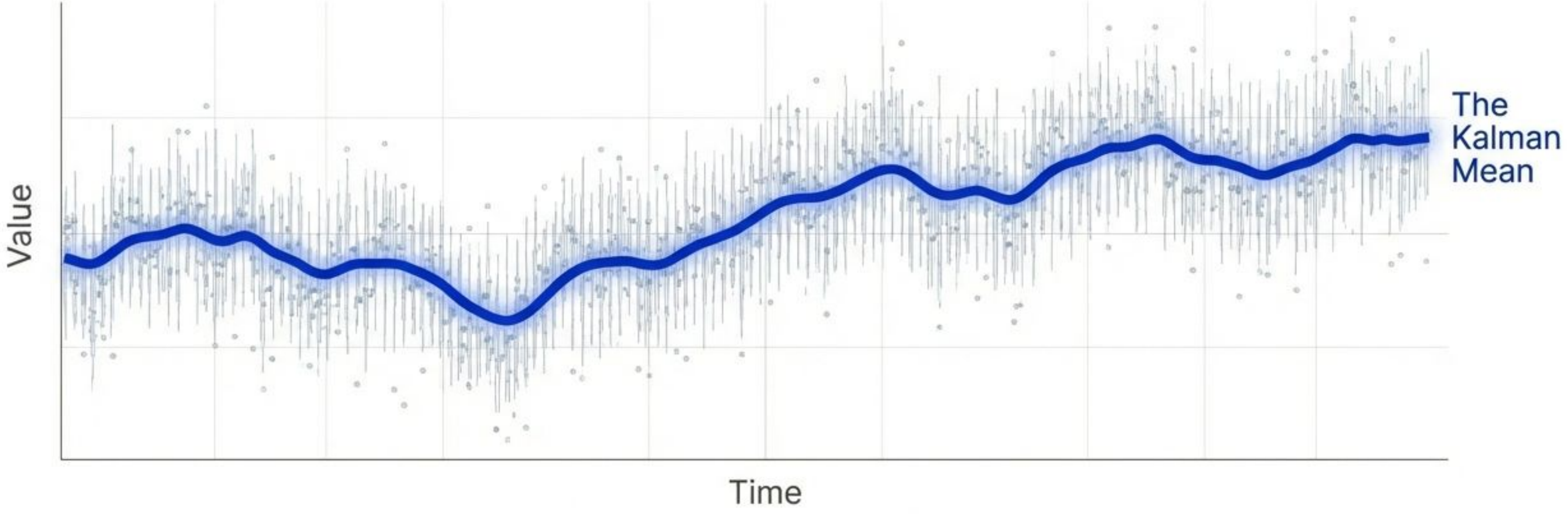
진입 시점 포착 (Generating the Alpha)

$$Z_t = \frac{e(t)}{\sqrt{Q(t)}}$$



- 단순한 가격 차이(e)가 아니라, 불확실성(Q)으로 정규화된 Z-Score를 사용합니다.
- 변동성이 큰 장에서는 진입 장벽이 높아지고, 조용한 장에서는 민감하게 반응합니다.

정적인 평균에서 동적인 신호로



- 1. 칼만 필터는 매 틱(Tick)마다 진정한 가치를 재평가합니다.
- 2. 거래량 정보를 결합하여 노이즈에 속지 않습니다.
- 3. 가장 강력한 마켓메이킹 도구는 '예측'과 '수정'의 끊임없는 반복입니다.