

Quant Edge PineScript Pack

Three edge-based indicators with hypotheses, parameter rationale,
and falsification protocols

Quant Edge · Edition 2026.05

Contents

1. Introduction — What this pack is for
2. Quant Edge Regime Filter
3. Quant Edge Liquidity Sweep Detector
4. Quant Edge Volatility Contraction
5. How to use the Falsification Protocol
6. Subscribe for weekly drops

Introduction

The three indicators in this pack share a discipline rare in retail TA: each ships with a **falsifiable hypothesis**, a documented reason for every parameter default, and a falsification protocol you can run to decide whether the indicator has edge on your instrument. If an indicator's falsification test fails on your data, it has no edge for you. Discard it. That is the discipline. This pack is the first edition. **Subscribers to the Quant Edge PineScript Pack on Whop receive one new indicator every Monday, perpetually**, generated by the same methodology.

The three discipline rules behind this pack

1. **Hypothesis must be falsifiable.** Every indicator can be tested. "RSI says oversold" is not a falsifiable claim — it's a vibe.
2. **Parameter defaults must be justified.** Every number in every input has a stated reason, not an artifact of curve-fit optimization.
3. **Stacked features must be orthogonal.** Two correlated filters do not improve signal. Each indicator stays focused on one mechanism.

Qe Liquidity Sweep

```
//@version=5
// =====
// QE Liquidity Sweep Detector v1.0 (Quant Edge PineScript Pack)
//
// HYPOTHESIS
// Sweeps of recent swing highs/lows separate into two regimes based on
// displacement of the close from the wick extreme:
// - HIGH displacement (close far from wick): liquidity sweep → MEAN REVERT
// - LOW displacement (close near wick): stop run → CONTINUATION likely
// The bar's range vs prior N-bar average gates the signal (velocity filter).
//
// USE CASE
// Standalone alerts when a high-quality liquidity sweep occurs. Use as input
// to a fade strategy (long after sweep of swing low; short after sweep of
// swing high) with stop beyond the swept level.
//
// PARAMETERS (defaults from a bilateral US30 sweep-fade that posted 357%/mo OOS
// in backtest; transferred to a retail-friendly Pine implementation here)
// swingLookback = 20 // bars for swing high/low detection
// sweepDisplacementMin = 0.40 // min (close offset / bar range)
// sweepVelocityMin = 1.20 // min (bar range / prior avg)
// priorAvgLookback = 20 // bars for velocity average
//
// FALSIFICATION
// Backtest the alert as long-on-sweep-low / short-on-sweep-high with SL beyond
// the swept level and TP at 5× SL. If win rate × avg R does not yield
// positive expectancy across at least 30 sweeps in walk-forward, this signal
// has no edge on the chosen instrument. Tune or discard.
// =====

indicator("QE Liquidity Sweep Detector v1.0", overlay=true)

// --- Inputs
swingLookback = input.int(20, "Swing lookback bars", minval=3, maxval=200)
sweepDisplacementMin = input.float(0.40, "Min close displacement", minval=0.1,
maxval=1.0, step=0.05)
sweepVelocityMin = input.float(1.20, "Min bar velocity", minval=1.0,
maxval=5.0, step=0.05)
priorAvgLookback = input.int(20, "Velocity avg lookback", minval=5, maxval=100)
showLabels = input.bool(true, "Show sweep labels")

// --- Swing levels
swingHigh = ta.highest(high, swingLookback)[1]
swingLow = ta.lowest(low, swingLookback)[1]

// --- Bar metrics
barRange = high - low
avgRange = ta.sma(barRange, priorAvgLookback)
```

```

velocity      = barRange / nz(avgRange, barRange)

// --- Sweep detection
hitHigh      = high > swingHigh
hitLow       = low < swingLow
displacementHi = (high - close) / barRange // how far close pulled back from sweep top
displacementLo = (close - low) / barRange // how far close pulled back from sweep bottom

sweptHighRevert = hitHigh and (displacementHi >= sweepDisplacementMin) and (velocity >=
sweepVelocityMin) and (close < swingHigh)
sweptLowRevert  = hitLow  and (displacementLo >= sweepDisplacementMin) and (velocity >=
sweepVelocityMin) and (close > swingLow)

// --- Continuation flag (low displacement, high velocity)
sweptHighContinue = hitHigh and (displacementHi < sweepDisplacementMin) and (velocity >=
sweepVelocityMin)
sweptLowContinue  = hitLow  and (displacementLo < sweepDisplacementMin) and (velocity >=
sweepVelocityMin)

// --- Visualization
plot(swingHigh, "Prior Swing High", color=color.new(color.gray, 60))
plot(swingLow,  "Prior Swing Low",  color=color.new(color.gray, 60))

plotshape(showLabels and sweptHighRevert, title="Sweep High → Fade short", text="↓
FADE", location=location.abovebar, color=color.red, style=shape.labeldown,
textcolor=color.white)
plotshape(showLabels and sweptLowRevert, title="Sweep Low → Fade long", text="↑
FADE", location=location.belowbar, color=color.green, style=shape.labelup,
textcolor=color.white)
plotshape(showLabels and sweptHighContinue, title="Sweep High → Continuation", text="↑
cont", location=location.abovebar, color=color.orange, style=shape.triangledown)
plotshape(showLabels and sweptLowContinue, title="Sweep Low → Continuation", text="↓
cont", location=location.belowbar, color=color.orange, style=shape.triangleup)

// --- Alerts
alertcondition(sweptHighRevert, "Sweep High Fade", "QE Sweep: short opportunity after
swept high with displacement")
alertcondition(sweptLowRevert, "Sweep Low Fade", "QE Sweep: long opportunity after
swept low with displacement")
alertcondition(sweptHighContinue, "Sweep High Continuation", "QE Sweep: continuation
signal after high break with low displacement")
alertcondition(sweptLowContinue, "Sweep Low Continuation", "QE Sweep: continuation
signal after low break with low displacement")

```

Qe Regime Filter

```
//@version=5
// =====
// QE Regime Filter v1.0 (Quant Edge PineScript Pack)
//
// HYPOTHESIS
// Mean-reversion signals fail systematically in a specific regime:
// range expansion is LOW (compression) AND trend strength is HIGH (directional
// force building). In this regime, mean-reversion signals should INVERT, not
// suppress – the compression-with-trend signature predicts breakouts, not fades.
//
// USE CASE
// Overlay on any existing mean-reversion or breakout system. When the regime
// flag fires (AND-logic on two orthogonal conditions), suggest the trader
// reverse direction on their next signal instead of trusting the mean
// reversion bias.
//
// PARAMETERS (defaults from edge-research work)
// rangeExpansionMax = 0.85 // max ratio of current range vs lookback range
// trendStrengthMin = 0.20 // min normalized trend score (ADX-like, 0-1)
// lookback = 65 // bars used for both calculations
//
// EDGE LINEAGE
// Pattern derived from a regime-aware AND-logic detector that flipped one
// backtest fold from -$3,147 to +$1,664 on M5 indices. The key insight: OR
// logic fires too often, AND logic with inversion (not suppression) captures
// the regime where the underlying strategy's assumptions invert.
//
// FALSIFICATION
// On any underlying mean-reversion strategy, walk-forward test with regime
// overlay (AND-fire days inverted) vs without. If AND-fire-day inversion
// doesn't improve OOS Sharpe by >0.15 on aggregate, this filter has no edge
// on that strategy. Re-test before deploying capital.
// =====

indicator("QE Regime Filter v1.0", overlay=true)

// --- Inputs
lookback = input.int(65, "Lookback bars", minval=10, maxval=300)
rangeExpansionMax = input.float(0.85, "Max range expansion ratio", minval=0.1,
maxval=2.0, step=0.05)
trendStrengthMin = input.float(0.20, "Min trend strength (0-1)", minval=0.0,
maxval=1.0, step=0.05)
showInversion = input.bool(true, "Show inversion suggestion")
plotBg = input.bool(true, "Highlight AND-fire bars")

// --- Range expansion: current range vs lookback range mean
currRange = high - low
avgRange = ta.sma(currRange, lookback)
```

```

rangeExpansion = currRange / avgRange

// --- Trend strength: ADX-style normalized to [0, 1]
[diPlus, diMinus, adx] = ta.dmi(14, 14)
trendStrength = math.min(1.0, adx / 50.0)

// --- AND-logic regime detection
regimeCompressionTrend = (rangeExpansion < rangeExpansionMax) and (trendStrength >
trendStrengthMin)

// --- Visualization
bgcolor(plotBg and regimeCompressionTrend ? color.new(color.orange, 80) : na,
title="Regime AND-fire")
plotshape(showInversion and regimeCompressionTrend and not regimeCompressionTrend[1],
title="INVERT", text="△ INVERT", location=location.abovebar, color=color.orange,
style=shape.labeldown, textcolor=color.white)

// --- Plot diagnostic lines on lower panel via separate sub-indicator recommended
// (range expansion + trend strength can be plotted in a companion indicator)

// --- Alert conditions
alertcondition(regimeCompressionTrend and not regimeCompressionTrend[1], title="QE Regime:
INVERSION on", message="QE Regime Filter fired – invert direction of next mean-reversion
signal")
alertcondition(not regimeCompressionTrend and regimeCompressionTrend[1], title="QE Regime:
INVERSION off", message="QE Regime returned to normal – resume baseline mean-reversion
bias")

```

Qe Volatility Contraction

```
//@version=5
// =====
// QE Volatility Contraction v1.0 (Quant Edge PineScript Pack)
//
// HYPOTHESIS (Minervini VCP-style)
// Stage-2 uptrending stocks form bases through three contractions where each
// pullback is shallower than the last AND volume dries up. The third
// contraction's high becomes the pivot – breakout above pivot on volume
// expansion signals continuation.
//
// THIS INDICATOR DETECTS
// - Sequence of 3 lower swing-high pullbacks within a base
// - Each pullback's range is < 0.7× prior pullback's range (contraction)
// - Volume on each pullback < 80% of 50-bar average (drying up)
// - Pivot = highest swing high in the base
// - Breakout = close > pivot AND volume > 1.4× 50-bar average
//
// USE CASE
// Stock screener overlay. Combine with trend filter (price > 200 SMA) for
// classic O'Neil / Minervini setups. Long entries on breakout bar; stop
// beneath the third contraction's low.
//
// PARAMETERS
// pivotLookback      = 10
// contractionRatio    = 0.7
// volumeDryRatio      = 0.80
// breakoutVolumeMult = 1.40
// stageFilter         = true (require price > 200 SMA)
// =====

indicator("QE Volatility Contraction v1.0", overlay=true)

pivotLookback      = input.int(10, "Pivot lookback", minval=3, maxval=50)
contractionRatio    = input.float(0.7, "Contraction ratio (each pb)", minval=0.3,
maxval=0.95, step=0.05)
volumeDryRatio      = input.float(0.80, "Volume dry ratio (vs avg)", minval=0.3,
maxval=1.0, step=0.05)
breakoutVolumeMult  = input.float(1.40, "Breakout volume multiplier", minval=1.0,
maxval=3.0, step=0.1)
stageFilter         = input.bool(true, "Require price > 200 SMA")
showBase           = input.bool(true, "Highlight base region")

// --- Stage 2 trend filter
sma200 = ta.sma(close, 200)
inStage2 = not stageFilter or (close > sma200)

// --- Swing point detection (pivot highs and lows)
ph = ta.pivohigh(high, pivotLookback, pivotLookback)
```

```

pl = ta.pivotlow(low, pivotLookback, pivotLookback)

// --- Track last 3 pullbacks (range = pivot_high - subsequent pivot_low)
var float lastHigh1 = na
var float lastHigh2 = na
var float lastHigh3 = na
var float lastLow1 = na
var float lastLow2 = na
var float lastLow3 = na

if not na(ph)
    lastHigh3 := lastHigh2
    lastHigh2 := lastHigh1
    lastHigh1 := ph

if not na(pl)
    lastLow3 := lastLow2
    lastLow2 := lastLow1
    lastLow1 := pl

// --- Contraction check: each pullback shallower than prior
range1 = lastHigh1 - lastLow1
range2 = lastHigh2 - lastLow2
range3 = lastHigh3 - lastLow3

contracting = not na(range1) and not na(range2) and not na(range3) and (range1 < range2 *
contractionRatio) and (range2 < range3 * contractionRatio)

// --- Volume dry on pullbacks (proxy: current bar volume vs SMA50)
volAvg = ta.sma(volume, 50)
volumeDry = volume < volAvg * volumeDryRatio

// --- Pivot = highest of last 3 swing highs
pivot = math.max(nz(lastHigh1, low), math.max(nz(lastHigh2, low), nz(lastHigh3, low)))

// --- Breakout
breakoutBar = inStage2 and contracting and (close > pivot) and (volume > volAvg *
breakoutVolumeMult)

// --- Visualization
plot(pivot, "Pivot", color=color.new(color.purple, 50), linewidth=2,
style=plot.style_circles)
plot(sma200, "200 SMA", color=color.new(color.gray, 40))
bgcolor(showBase and contracting and inStage2 ? color.new(color.blue, 90) : na,
title="Base region")
plotshape(breakoutBar, title="Breakout", text="BREAKOUT", location=location.belowbar,
color=color.green, style=shape.labelup, textcolor=color.white)

// --- Alerts
alertcondition(breakoutBar, "VCP Breakout", "QE VCP: Stage-2 stock breaking out of
contraction base on volume")
alertcondition(contracting and not contracting[1], "Base Formed", "QE VCP: 3-stage
contraction base just formed; watch pivot for breakout")

```

How to use the Falsification Protocol

Every indicator includes a "Falsification" comment block. To use it:

1. Apply the indicator on the instrument and timeframe you intend to trade.
2. Implement the falsification test the indicator describes (typically a backtest of a specific entry/exit rule that uses the indicator).
3. If the test fails the documented threshold (e.g., "Sharpe must improve by 0.15"), the indicator has no edge on your data — do not deploy capital.
4. If the test passes, treat the result as a hypothesis to validate further with walk-forward and stress-test slippage.

This protocol is the single most important practice in this pack. Skip it and the indicators are no better than any retail TA library.

Subscribe for weekly drops

The Quant Edge PineScript Pack is updated weekly. Every Monday, subscribers receive a new indicator generated by the same methodology, with the same level of hypothesis, parameter, and falsification documentation.


\$19/month, \$179/year, or \$499 lifetime. Available on Whop.

The pack is also a marketing channel for the companion **Quant Edge Notes** weekly newsletter on systematic methodology — free tier on Substack.

Support This Work (Crypto)

This pack and the weekly updates are generated by an autonomous research and authoring pipeline. If the indicators saved you time, you can support continued work directly via any EVM-compatible wallet (ETH, USDC, USDT on Ethereum, Base, Arbitrum, Optimism, or Polygon):

0x1AD997D684Eb7BD1A6e6A0f2aB1728D5E8Ba0B4a

 ETH receive address QR

No fees taken in transit. No platform middleman. Sent funds are recoverable by the author through the wallet's mnemonic.