```
// This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 4.0
International
// https://creativecommons.org/licenses/by-nc-sa/4.0/

// © Zeiierman {
//@version=6
indicator('Dynamic Swing Anchored VWAP (Zeiierman)', overlay = true, max_bars_back = 5000,
max_labels_count = 500, max_polylines_count = 100)
//~~}

// ~~ Tooltips {
var string t1 = "Number of bars used to detect swing highs and lows. Larger values identify
bigger, more significant swings but react slower. Smaller values detect more frequent swings but
may produce more noise."
var string t2 = "Controls how quickly the VWAP adjusts to new price action. Lower values make
the VWAP react faster (tighter to price), higher values make it smoother and slower to change."
var string t3 = "When enabled, the VWAP reaction speed changes automatically based on market
volatility. High volatility shortens the tracking period (more responsive), low volatility lengthens it
(smoother)."
var string t4 = "Controls how strongly volatility influences the VWAP reaction speed. Values
above 1 increase the effect of volatility changes; values below 1 make it less sensitive to
volatility."
var string t5 = "Color used for swing high/low labels drawn on the chart to indicate pivot points."
var string t6 = "Color used for swing low labels when marking pivot points."
var string t7 = "Color used for VWAP lines when in an uptrend."
var string t8 = "Color used for VWAP lines when in a downtrend."
var string t9 = "Width of the VWAP lines drawn on the chart. Larger values make the lines thicker
and more visible."
//
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~}

// ~~ Inputs {
prd = input.int(50, title='Swing Period', minval=2, group='Swing Points', tooltip=t1)
baseAPT = input.float(20, 'Adaptive Price Tracking', minval=1, step=1, group='Swing Points',
tooltip=t2)
useAdapt = input.bool(false, 'Adapt APT by ATR ratio', group='Swing Points', tooltip=t3)
volBias = input.float(10.0, 'Volatility Bias', minval=0.1, step=0.1, group='Swing Points', tooltip=t4)

highS = input.color(color.lime, title="Swing Labels", group="Style", inline="Swing", tooltip=t5)
lowS = input.color(color.red, title="", group="Style", inline="Swing", tooltip=t6)
S = input.color(color.lime, title="VWAP Lines", group="Style", inline="VWAP", tooltip=t7)
R = input.color(color.red, title="", group="Style", inline="VWAP", tooltip=t8)
xx = input.int(2, minval=1, title="", group="Style", inline="VWAP", tooltip=t9)
//
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~}

// ~~ Global Variable {
b = bar_index
//
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~}

// ~~ PIVOTS Variables {
var ph = float(na)
```

```
var pl = float(na)
var phL = b
var plL = b
var lab = label(na)
var prev = float(na)

ph := ta.highestbars(high, prd) == 0 ? high : ph
pl := ta.lowestbars(low, prd) == 0 ? low : pl
phL := ta.highestbars(high, prd) == 0 ? b : phL
plL := ta.lowestbars(low, prd) == 0 ? b : plL
dir = phL > plL ? 1 : -1
//
~~~
```

```
barsback = b - x
p := y * volume[barsback]
vol := volume[barsback]
vap = p / vol

vwap.poly.delete()
polyline.new(vwap.points, false, false, line_color = dir < 0 ? R : S, line_width = xx)
vwap.points.clear()

for i = barsback to 0 by 1
apt_i = aptSeries[i]
alpha = alphaFromAPT(apt_i)

pxv = hlc3[i] * volume[i]
v_i = volume[i]

p := (1.0 - alpha) * p + alpha * pxv
vol := (1.0 - alpha) * vol + alpha * v_i
vappe = vol > 0 ? p / vol : na

vwap.points.push(chart.point.from_index(b - i, vappe))

vwap.poly := polyline.new(vwap.points, false, false, line_color = dir < 0 ? R : S, line_width = xx)

else
apt_0 = aptSeries
alpha = alphaFromAPT(apt_0)

pxv = hlc3 * volume
v0 = volume

p := (1.0 - alpha) * p + alpha * pxv
vol := (1.0 - alpha) * vol + alpha * v0
vap = vol > 0 ? p / vol : na

vwap.poly.delete()
vwap.points.push(chart.point.from_index(b, vap))
vwap.poly := polyline.new(vwap.points, false, false, line_color = dir > 0 ? R : S, line_width = xx)
//~~ }
```