

# How to Calculate Logarithmic Retracements and Extensions Using EWT

## A Quick Note on Scaling

Elliott Wave Theory (EWT) can be used in both Linear and Logarithmic Scaling. Most analysis for Medium or Higher Time Frame (MTF and HTF respectively) is conducted using Logarithmic Scaling. As such, I will be going over Logarithmic Scaling and ignoring Linear.<sup>1</sup>

## Retracements

Logarithmic Retracements are a bit of a weird process. To explain, imagine you have a stock that you believe started a Market Cycle at  $W_0$ , impulsively climbed to  $W_1$ , and retraced to  $W_2$ . The equation to solve for the logarithmic retracement (denoted as  $r$ ) is:

$$r = \frac{\ln |W_1 \div W_2|}{\ln |W_1 \div W_0|} \quad (1)$$

## Extensions

Logarithmic Extensions, in my opinion, are far more intuitively understandable. To explain, similarly imagine you have graphically obvious Wave 3 Extension, with defined Wave Origin ( $W_0$ ), 1 ( $W_1$ ), 2 ( $W_2$ ), and 3 ( $W_3$ ) price points. The equation to find the extension (denoted as  $\varepsilon$ ) is:

$$\varepsilon = \frac{\ln |W_3 \div W_2|}{\ln |W_1 \div W_0|} \quad (2)$$

## Example Using Polkadot (\$DOT)

Given the crypto-winter seems to be finally over since the peak around November 2021 I have examined various stocks and cryptocurrencies. \$DOT has one of the most typical EWT structures of those I looked at and thus is perfect for this example. For a high resolution picture of my EWT charting see: <https://www.tradingview.com/x/bzHxeU8N>.

\$DOT began a Cycle Wave Degree on 19 Aug 2020 at \$2.71 ( $W_0$ ) and peaked on 4 Nov 2021 at \$55 ( $W_1$ ). It retraced until 30 Dec 2022, rebounding off \$4.2246 ( $W_2$ ). The Primary Wave Degree breakdown within that Cycle Wave 1 is as follows:  $W_1 = \$6.8674$ ,  $W_2 = \$3.6035$ ,  $W_3 = \$49.7522$ , and  $W_4 = \$10.3785$ .

Therefore, \$DOT's Cycle Wave 2 retraced 85.3% and Cycle Wave 1 Primary Wave 3 was a 282.3% extension.

---

<sup>1</sup> In Linear terms, figuring these out is also fairly self-explanatory.