

SETTING STOP-LOSSES USING PRICE VOLATILITY

by Cynthia A. Kase



By accounting for volatility, the variance in volatility, and volatility skew, Kase DevStops provide a statistically sound basis upon which to place stop-loss and stop-and-reverse orders.

The idea of using volatility-based stops first occurred to me after I read Welles Wilder's *New Concepts in Technical Trading*, published in 1978. In this book, Wilder sets out a stop and reverse system centred around a measure called "True Range". It was my reading of that book along with the burst in volatility that took place in the oil market in 1990 that led me to investigate the use of True Range in setting the size of stops, and ultimately to the development of the Kase DevStop system in 1991.

The DevStop

A number of trading techniques use average True Range (ATR) values with fixed multipliers to determine where to place stop and reverse orders - for example, stops might be set at 3 x ATR (see Box for definition of TR and ATR). Yet while True Range is proportional to volatility and using such an approach is an improvement over a fixed-value stop, the use of the average (ATR) is insufficient to truly capture market behaviour.

Think of two populations, Population A and Population B. Population A is comprised of chorus girls and has an average height of 5'10" - the shortest is 5'9" and the tallest 5'10½". Population

True Range

True Range (TR) is the full price range of a period and is defined as the largest value of the following three calculations:

1. TR = Highest price of current period("H") - Lowest price of current period("L")
2. TR = Highest price of current period("H") - Closing price of previous period("C")
3. TR = Closing price of previous period("C") - Lowest price of current period("L")

Average True Range (ATR) is the simple average of True Range over the past n periods or an exponential moving average.

Source: The Encyclopedia of Technical Market Indicators, Colby, 2003

B is comprised of professional basketball players and their elementary school aged children, and has an average height of 5'10" - the shortest is 3'2" and the tallest 6'11". Obviously, even though both populations have the same average height, a higher value is needed to be equivalent to, say, the 95th percentile.

The same is true for a market. A stop that is going to perform optimally must consider the variability of range, not just the average. So I developed the DevStop, where "Dev" stands for standard deviation around the mean.

The value of the DevStop is calculated as follows:

$$\text{DevStop Amount} = \text{average}(2 \times \text{TR}, n) + \text{stddev}(2 \times \text{TR}, n)$$

For the simple reason that True Range values based on single bars are too small, notice the DevStop employs the True Range of two bars (2 x TR),

which we call the True Range Double (TRD).

We can use this calculation to determine the points at which the stops should be set. For a normal bell curve, for example, a stop placed at the one standard deviation level has an 85% chance of being hit and a stop placed at the 1.65 standard deviation level has a 95% chance of being hit.

Volatility skew

The problem with using a normal bell curve is that it doesn't accurately reflect how range is really distributed. When we look at range as a proxy for volatility (which is a logarithmic term) the range is more or less log normally distributed as opposed to normally distributed. Thus it has a right hand skew.

Figure 1 shows the distribution of the daily TRD values for the Live Cattle contract from August 1969 to August 2004. It can clearly be seen that there is a severe right hand skew. Indeed a full two-thirds of the data are below →

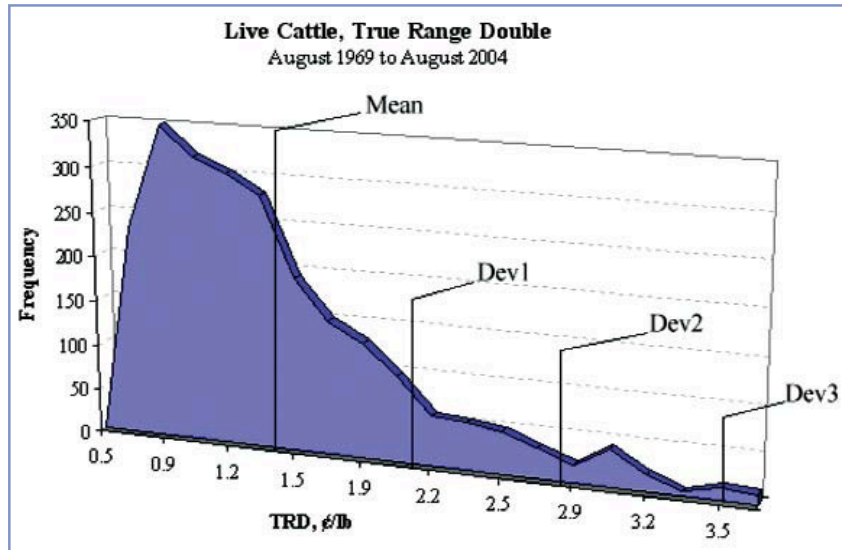


Figure 1. Distribution of TRD

the average TRD of 1.4 cents. Therefore a correction of the normal standard deviation levels must be made to account for the skew.

Back in the early 1990's when we first developed the DevStop we performed research on 100 bar subsets of a total of 10,000 bars of 10 minute British Pound data to estimate the skew corrections that would be needed to equate to 1.0, 2.0 and 3.0 standard deviations. We found that the one standard deviation level, on average, did not require a correction, and that the two and three standard deviation levels needed to be corrected by about 10 percent and 20 percent respectively, that is, to 2.2 and 3.6 standard deviations over the mean. Figure 2 shows an example of the DevStops on the daily chart for the October 2004 Lean Hogs contract, where each line has been calculated as follows:

$$\text{Warning Line} = \text{average}(\text{TRD}, n) + 0\text{stddev}(\text{TRD}, n)$$

$$\text{Dev 1} = \text{average}(\text{TRD}, n) + 1\text{stddev}(\text{TRD}, n)$$

$$\text{Dev 2} = \text{average}(\text{TRD}, n) + 2.2\text{stddev}(\text{TRD}, n)$$

$$\text{Dev 3} = \text{average}(\text{TRD}, n) + 3.6\text{stddev}(\text{TRD}, n)$$

The computer does not know, of course, if one is long or short, so a moving average crossover system is embedded into the front end of the code. If the fast moving average, normally defaulted to five (5) is above the slow, defaulted to 21, then the computer assumes you are long and trails the stop from the highest high since the moving averages crossed. If the fast is below the slow, the DevStops trail from the lowest low. The stops are flipped from below the data to above the data when the averages flip from long to short and vice versa. But the front end could be programmed for any entry system.

Adjusting for skew

Recently I decided that it was time to update the original study. Rather than using 10-minute bars on one contract, we decided to test daily data over a range of physical commodity and financial futures from contract inception to September 2004, amounting to more than 50,000 data points. In addition, we also evaluated the data in 250 bar sets which give a more representative population than the 100 bar sets, stepped backward in 100 bar increments. Table 1 lists the study data.

The first question I addressed as I looked at the results is how the data

compares with our prior adjustments, i.e.: no correction at the one standard deviation level, a 10 percent correction at the two standard deviation level and a 20 percent correction at the three standard deviation level, resulting in one, 2.2 and 3.6 standard deviations over the mean for the DevStops. Table 2 shows the percentile rankings and number of standard deviations for a normal bell curve versus the average and median values for the actual data.

I found that the warning line, which is set at the average of the data, or zero (0.0) standard deviations over the mean, actually captures about 60 percent of the observations. This can be seen by looking across the row marked "60" in bold under the percentile column. Where a normal bell curve would require a setting of 0.3 standard deviations over the mean to be at this point, the actual data only requires a setting of zero.

At the 85th percentile the one standard deviation setting for normal bell curve corresponds to about 0.95 for the actual data, which means that a set-

Contract	Symbol	Start Date	Data Points
Crude Oil	CL	03/30/83	5373
Corn	C	03/28/68	9190
US Dollar Index	DX	11/20/85	4752
Copper	HG	08/22/72	8034
Natural Gas	NG	04/03/90	3608
S&P 500 Index	SP	04/21/82	5649
Soybeans	S	02/02/68	9210
10 yr T-Notes	TY	07/09/82	5626

Table 1. Data used to September 13, 2004

Percentile	Normal Bell		Actual Data	
	Average	Average	Median	
50	0.0	-0.21	-0.21	
55	0.1	-0.11	-0.10	
60	0.3	0.0	0.0	
65	0.4	0.15	0.15	
70	0.5	0.30	0.30	
75	0.7	0.47	0.48	
80	0.8	0.67	0.69	
85	1.0	0.94	0.95	
90	1.28	1.28	1.29	
95	1.6	1.88	1.89	
97.5	2.0	2.42	2.40	
99.86	3.0	3.57	3.48	

Table 2. Normal Bell Curve vs. Actual Data

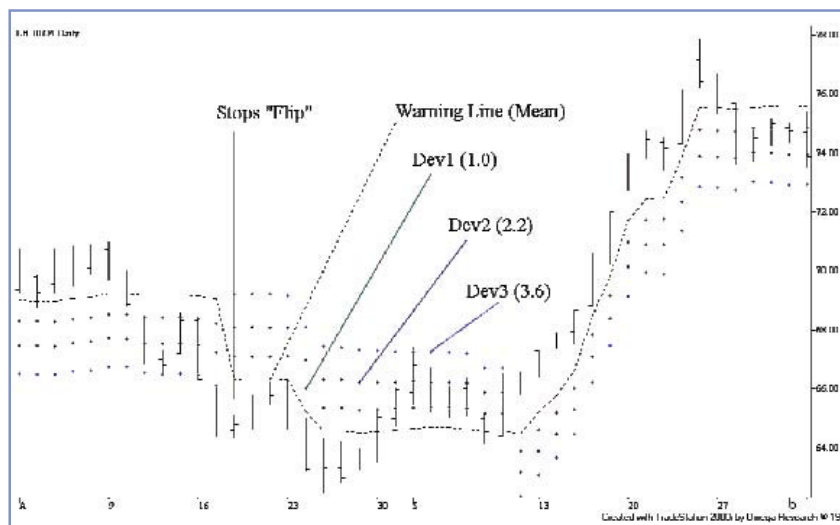


Figure 2. DevStops on October 2004 Lean Hogs Daily Chart

ting of 1.0 for DevStops is roughly equivalent to a very slightly higher percentile reading at about the 86th percentile. At the 90th percentile, marked in italics, the curves cross from the actual data being at a standard deviation level slightly less than that generated by the bell curve to slightly more. So using 1.28 for a DevStop setting would on average generate a stop at the 90th percentile, the same as a bell curve.

At two (2.0) standard deviations, a correction of 20 percent would be needed to correspond with the actual data, the same as has been used all these years for the three (3.0) standard deviation stop. Rather than correspon-

ding to 2.0 standard deviations over the mean at the 97.5 percentile, a setting of 2.2, on average, corresponds with about a percent lower, that is, with the 96.5 percentile. So the use of 2.2 rather than 2.4 has been accurate within one percent of its original intent. As noted, the use of 3.6 - a 20 percent correction to 3.0 - is almost exactly correct.

Finally, I looked at the degree of variation among the futures contracts studied (Table 3). While a fairly wide range of variation between each data set was found (each data set being 250 bars), there is little variation among futures contract. The average standard deviation for the results by data set is 0.13

“A STOP THAT IS GOING TO PERFORM OPTIMALLY MUST CONSIDER THE VARIABILITY OF RANGE.”

CYNTHIA KASE

(results not presented here), but the average variation by commodity is only 0.03, less than 25% that of the variation by data set. This means that the range of variation by data set is about the same no matter what is traded.

In summary, the methodologies that were used a dozen years ago to develop the Kase DevStop are still valid. The one standard deviation level used for Dev1 and 2.2 standard deviation level used for Dev2 are within a percentile of the settings that were the original intent, specifically the 86th vs. 85th percentile for Dev1 and 96.5 versus 97.5 for Dev2. For Dev3, the original setting at 3.6 is right on the mark. There is some variability around the mean values found, but there is insignificant variation by commodity. That it, regardless of the commodity studied, the results come out about the same, so no adjustment must be made to the DevStops based on the futures contract traded.

Percentile	CL	C	DX	HG	NG	SP	S	TY	Std Dev
50	-0.22	-0.22	-0.18	-0.19	-0.24	-0.21	-0.23	-0.19	0.02
55	-0.12	-0.12	-0.09	-0.09	-0.12	-0.11	-0.12	-0.08	0.02
60	-0.01	0.00	0.03	0.03	0.00	0.02	0.00	0.04	0.02
65	0.11	0.12	0.17	0.17	0.14	0.15	0.13	0.18	0.02
70	0.26	0.26	0.32	0.31	0.29	0.31	0.29	0.32	0.02
75	0.43	0.44	0.49	0.48	0.47	0.49	0.47	0.47	0.02
80	0.65	0.63	0.68	0.69	0.68	0.69	0.67	0.67	0.02
85	0.92	0.89	0.94	0.96	0.95	0.95	0.94	0.94	0.02
90	1.26	1.27	1.27	1.29	1.30	1.28	1.29	1.30	0.01
95	1.86	1.92	1.84	1.91	1.89	1.84	1.90	1.90	0.03
97.5	2.43	2.48	2.33	2.37	2.39	2.44	2.50	2.44	0.06
99.86	3.69	3.69	3.58	3.50	3.56	3.57	3.53	3.43	0.09

Table 3. Average Standard Deviation Results by Commodity vs. Percentile of Data

Cynthia Kase is president of Kase and Company, Inc. (www.kaseco.com). Kase DevStop is available on CQG, TradeStation, eSignal, Aspen Graphics and DTN ProphetX.