

GP Trend Line DLL

The trend line DLL plots trend lines along a price chart. Starting at the right-hand end, it alternates backwards between rising and falling trend lines, using peaks and troughs as the turning points. Peaks and troughs are determined by a percentage change parameter, which determines how far the price must move against the trend behind a peak or trough to signal a change. The direction of the first (most recent) trend line is passed by the caller, perhaps determined by the slope of a moving average. Only a single trend line is drawn for each direction. If, for example, an uptrend sharply changes slope such that two separate trend lines would normally be drawn, the DLL will just attempt to fit the best single trend line to the whole region.

The plotting routine also checks for price breaks above down-sloping trend lines. These occur when the closing price rises above the initial cross price by a specified percentage, where the initial cross price is the trend line price value on the day the closing price first crossed above the trend line. An array is returned indicating the days where this occurred for each plotted downward-sloping trend line (if it happened at all). For a price break on the most-recent trend line, values are also returned indicating the bar index of the day it happened, the price value of the trend line on the initial cross day, and the bar index of the most recent peak. These latter values are mainly of use in explorations and backtests.

Installation

To install the plugin, copy the file "GP_Trendlines.dll" into the AmiBroker "Plugins" folder. If AmiBroker is currently running, select "Tools->Plug-ins" from the menu, otherwise just start AmiBroker. A warning will probably appear about a new plugin being found.

To use with the indicator AFL code, create a new indicator and call it "GP_Trendlines". Then either copy the file "GP_Trendlines.afl" into the appropriate folder (exactly which one depends on the version of AmiBroker) or open that AFL file in notepad then edit the new indicator and copy and paste the code into it.

To use in an exploration, open the file "GP_Trendline Explore.afl" in the Automatic Analysis window, select the appropriate parameters and stocks to explore, set the range to "n last days = 1", then Explore.

To use in a backtest, open the file "GP_Trendline Backtest.afl" in the Automatic Analysis window, select the appropriate parameters and stocks to test, set the range to the desired From-To dates, then Back Test. Note that this can be very slow, taking some minutes to do one pass of the ASX 200 stocks on a 1.8GHz Centrino.

Trend Line Function And Variables

The trend line function definition is:

fit = **GP_DrawTrendLines**(*bottom*, *top*, *lineCnt*, *startIx*, *earliestIx*, *trendChange*, *rightOvershoot*, *closeOver*, *overAgo*, *startUp*, *isLog*);

Where:

"*fit*" is a line fit value for the most recent falling trend line.

"*bottom*" is an array of bottom prices.

"*top*" is an array of top prices.

"*lineCnt*" is the number of lines to draw (zero means fill chart).

"*startIx*" is the right-most bar index to start from.

"*earliestIx*" is the left-most bar index to draw back to.

"*trendChange*" is the percentage change for peak\trough detection.

"*rightOvershoot*" is the number of bars to extend a line past the start of the next one.

"*closeOver*" is the percentage the price must close above a falling line to signal a crossover.

"*overAgo*" is the number of days ago to accept a crossover signal from (explorations only).

"*startUp*" is one if start with upward trend line or zero if start with downward trend line.

"*isLog*" is one for logarithmic display or zero for linear display.

The DLL also requires the AFL code to have the following global variables defined:

```
GP_tlineUp1[0] = 0;
GP_tlineUp2[0] = 0;
GP_tlineDown1[0] = 0;
GP_tlineDown2[0] = 0;
GP_closeAbove[0] = 0;
GP_aboveIx = -1;
GP_yover = -1;
GP_recentPeak = -1;
GP_changeAtr[0] = 0;
```

Where:

"*GP_tlineUp1*" is the first up trend line array.

"*GP_tlineUp2*" is the second up trend line array.

"*GP_tlineDown1*" is the first down trend line array.

"*GP_tlineDown2*" is the second down trend line array.

"*GP_closeAbove*" is the close above signal array.

"*GP_aboveIx*" is the bar index of the most recent close above day.

"*GP_yover*" is the downward trend line price value on the first crossover day (for explorations and backtests only).

"*GP_recentPeak*" is the bar index of the most recent peak.

"*GP_changeAtr*" is an array to scale the peak\trough change parameter by to compensate for price volatility.

Each function parameter and global variable is explained below.

Top And Bottom Price Arrays

Upward-sloping trend lines are fitted to the bottom array while downward-sloping trend lines are fitted to the top array. These arrays are price arrays. Typically they would be either the High and Low arrays or arrays of the highest and lowest Open and Close values. Currently the code directly accesses the Close array when looking for price breakouts, so the function can't be used for drawing trend lines on volume, RSI, or other charts.

Line Count

The line count parameter is the maximum number of lines to draw, where zero means fill the price chart. For explorations and backtests, this value should always be two.

Start And Earliest Indices

The start index is the right-most bar index to start drawing from (remember lines are drawn from right to left). To start at the most recent day, set this to BarCount-1.

The earliest index is the left-most bar index to include in the trend line calculations and drawing. For the start of the chart, set this to zero.

By using the BarIndex, BeginValue, and EndValue AFL functions to set these values, trend lines can be limited to within a selected chart region.

Trend Change Percentage

This is the percentage move against the trend behind a peak or trough for that peak or trough to be taken as a change point for trend line direction. The higher the value, the longer-term the trend lines will become.

The change percentage is an actual percentage value (ie. a value of two means two percent). However, to compensate for price volatility, the value is multiplied by the contents of the *"GP_changeAtr"* array for the current bar before being used with the top or bottom price. This array can be filled with anything, and should be all ones to use the change percentage unmodified. When testing for top or bottom price beyond the change percentage range, the formula used is: $1 + (change \backslash 100) * GP_changeAtr$. This value is then multiplied by the top or bottom price of the peak or trough and compared to the current value of the top or bottom array.

My AFL routines fill this array with the formula: $1 + 5 * (ATR(3) / Close)$.

Right Overshoot

This is simply the number of bars past the change point of a trend line to continue drawing the line. A value of zero makes the trend line end exactly at the change point.

Note though that when a price breakout is signalled, the downward sloping trend line leading to it will automatically be extended to either the index of the signalled day or to a small amount below the lowest prices, whichever comes first.

Close Over Percentage

This is the percentage by which the closing price must rise above the initial cross price of a downward-sloping trend line to signal a price breakout. The initial cross price is the price value of the downward-sloping trend line on the day that the closing price first crossed above the trend line. This value is used as a base for the crossover percentage as the trend line continues to fall over time, and a price breakout might otherwise ultimately be signalled due to the fall in the trend line rather than an increase in closing price.

Over Ago Period

This value is only used in explorations to specify how many days back a price breakout above a downward-sloping trend line can have occurred to still signal a match. A value of zero means it must have happened on the most recent bar.

The only effect this variable has is to force the DLL function to return -1 as the fit value if the most recent price break out was not within this time period. All other variables are returned as usual.

Start Up Flag

This flag can be either zero or one to indicate whether the most recent trend line should be downward or upward-sloping respectively. This can be determined by any desired means, but might typically be by the slope of a suitable-period moving average.

Is Log Flag

This flag can be either zero or one to indicate whether the indicator display is linear or logarithmic respectively. This makes a difference to how the trend lines are drawn, where the incorrect setting will make the lines appear curved on a plot. Even where lines are not plotted (ie. in explorations and backtests), it will still affect the positioning of the lines and the price break out signals.

Fit Return Value

The return value is a number (not an array) that gives an indication of how well the last downward-sloping trend line fitted to the top price array. It is calculated over the length of the trend line by summing the percentage difference between the trend line and the top array price and then dividing by the number of bars, effectively giving an average percentage price differential. The differences used are actually fractions, not percentages, meaning simply that they haven't been multiplied by 100. The larger the number, the worse the fit.

To turn the value into a more readable form, with larger numbers being the better fit, the following AFL code can be used:

$$fit = 100/(1+100*fit);$$

This will force the value into the range 0-100, with 100 being a perfect fit.

If the fit value is returned as -1, then there was no price breakout above the most recent downward-sloping trend line.

Trend Line Arrays

Four array variables are used for the trend line plots: two for upward lines and two for downward lines. Upward and downward lines are separated to allow them to be plotted in different colours if desired. Each direction also uses two arrays, alternating between arrays as the lines are added, to help minimise the chance of two lines in the same direction overlapping. If they do overlap, the AFL plot routine will treat them as a single line and draw a line between the two ends. Even with two arrays, it is still possible to make lines overlap and appear joined in the plot (eg. if a large right overshoot value is specified).

To show all trend lines, all four arrays should be plotted. The two most-recent trend lines (one up, one down) will be in the first arrays for each direction.

Close Above Array

This array is returned with a one on any price breakout day and a zero on every other day.

Above Index

This returns the bar index of the most recent price breakout day. It is mainly of use in explorations and backtests.

Initial Close Over Price

This returns the price value of the most recent downward-sloping trend line on the day when the closing price first moved above the trend line, or -1 if it never did. It is mainly of use in explorations and backtests.

The closing price on the Above Index day minus this price gives the amount by which the closing price broke out on that day. Expressing it as a percentage of the initial close over price allows the strength of the price breakout on that day to be compared (eg. for use with the PositionScore backtest variable).

Most-Recent Peak Index

This returns the bar index of the most recent peak (ie. the first day of the most recent downward-sloping trend line). It is mainly of use in backtests to determine the right overshoot amount to specify when drawing upward-sloping trend lines while searching for sell signals (ie. where the closing price during a down-trend has dropped below the previous upward-sloping trend line).

Change ATR Array

This array is used by the DLL to compensate the peak\trough change percentage for price volatility. See the description under Trend Change Percentage above for more details. It must be filled with appropriate values before the DLL function is called. While it's called an ATR array, and I personally use the ATR function as part of the formula to fill it, it can be filled with any values.

I also use this array to compensate the initial trend direction EMA for price volatility, by using it to increase the EMA period with more volatile stocks.