THE TRADERS' MAGAZINE SINCE 1982 www.traders.com JANUARY 2026

TECHNICAL ANALYSIS OF STOCKS & COMMODITIES™

JANUARY 2026

$9.99US $9.99CAN

01>

0 71435 01384 2

Which Methods Work Best?

Smoothing The Data

Of the various smoothing methods commonly available to traders and technical analysts, which are best at finding the trend in price data?

We smooth price data in order to find the trend. It's an attempt to separate the "signal" from the "noise." In some markets, there is a lot of noise. But finding the signal, or the trend, allows us to take advantage of a price move by ignoring the unnecessary noise. But I'm sure you all know that.

Most of us, including me, start by using a simple moving average. It tends to work well. But everyone using the same method can lead to sudden jumps in price and poor fills. There are other smoothing methods that offer better returns and some diversification. We'll compare the following ten alternatives:

1. Simple moving average
2. Average-off method
3. Weighted moving average
4. Triangular moving average
5. Pivot point moving average
6. Standard deviation moving
7. Moving median
8. Geometric moving average
9. Exponential smoothing
10. Linear regression

Some of these have variations, but we'll try to use the simplest approach so that we can have a reasonable way to compare them. We'll apply them to the SPDR S&P 500 ETF (SPY) and the Euro-Bund 10-year note futures (FGBL) from 2000 (about 25 years). That way, we have a relatively noisy price series (SPY) and a fairly smooth one (FGBL).

You'll find an EasyLanguage program in the sidebar, "Smoothing The Data, In EasyLanguage" (cleverly named!), that will allow you to try these methods on other markets.

In the following, I'll use programming notation rather than mathematical symbols. Hopefully, it will be easier to read.

N will be the number of calculation days, and *t* is "today." For our stock tests, we'll use 110 and 20 days to see if a longer or shorter period makes a difference.

1. Simple moving average

SMA(*t*) = Average(close,*N*), the standard protocol, where *N* represents the calculation period and *t* is today.

2. Average-off method

For the average-off method, we subract one day from the previous average, AvgOff(*N*–1), and add today's price, then average.

$$\text{AvgOff}(t) = [(N-1)*\text{AvgOff}(t-1) + \text{close}(t)] / N$$

3. Weighted moving average

Each price can have its own weight, then we divide by the sum of the weights. If we have *N* days and give the nearest day a weighting of *N* and the farthest a weighting of 1, we give more value to the most recent price. Alternatively, we can group them so the *N* goes to the nearest two days, *N*–1 to the next two days, and so on. Or we can make the middle value the largest, in which case we have triangular weighting (our next option). For simplicity, we will start with *N* and reduce the weighting by 1 as prices get older. In the example below, we use *N*=10.

$$\text{WMA}(t) = \text{sum}[10*p(t) + 9*p(t-1) + 8*p(t-2) + \ldots + 1*p(t-9)] / 55$$

by Perry J. Kaufman

In the code in the "Smoothing The Data" sidebar, I use a "Do loop" to create the weighted moving average.

4. Triangular moving average

This is a weighted moving average with the middle value as *N* and values to the left and right of the middle value have *N* decreasing by 1. This will be easier to program than showing the series. In the following example calculation, we use *N*=5. The weights increase from 1 to 3 and then back to 1.

$$TMA(t) = sum[1*p(t) + 2*p(t–1) + 3*p(t–2) + 2*p(t–3) + 1*p(t–4)] / 9$$

Again, for simplicity, we will increase and decrease *N* by 1. You might note that a triangular moving average of 5 or 7 days has been used to identify a price cycle. We'll try those as well.

5. Pivot-point weighting

Pivot-point weighting is similar to triangular weighting in that it starts at 1, increases, then drops again. For a 10-day calculation, it starts at *N*=–3 on day 10 (the oldest data) and increases by 1 each day, ending at 7 for the newest data. We then divide by the total of the weights, which is 22. *N*=0 is at about 1/3 of the way through the calculation. For a 10-day pivot-point weighting, we would have:

$$PPW(t) = [–3*p(t–10) + –2*p(t-9) + –1*p(t–8) + 0 + 1*p(t–6) + … + 7*p(t)] / 22$$

Remember to divide by the sum of the weights. Negative weights are subtracted.

6. Standard deviation average

Modifying the moving average with the standard deviation of prices adds a touch of volatility. I tend to use 20 days for the standard deviation. You'll need to change the program in the sidebar if you want another value.

- Find the standard deviation, SD = STDDEV(close,20)
- Convert to a percentage, SD% = (SD(*t*) – SD(*t*–1)) / SD
- Create the SDavg = AVERAGE(close,10) + 0.05*SD%

I tend to use the standard deviation of the price returns rather than the price and divide by the previous value. Using the price includes a potential trend. In the code given in the sidebar, we'll stick to the original rules. This rule uses 5% of the standard deviation, which is pretty small. You might want to try something bigger.

7. Moving median

Finding the median can be a slow process if you're using a lot of data. It's the middle price, which means you first need to sort the data, preferably highest or lowest. If you use an odd number of prices, then the middle is easy to find. If you use an even number, you can average the two middle prices. For a 10-day median, we will use the more recent of the two middle values for simplicity. It should not make much difference.

I have always liked the median. It tends to be more stable than an average because it doesn't get distorted by an odd outlier.

8. Geometric moving average

The geometric moving average is a growth function. It is the product of *N* prices (we'll use *N*=10) divided by the root of 1/*N*. We get:

$$GMA(t) = [p(t) × p(t–1) × … × p(t–9)]^(1/10)$$

Using a large calculation period can cause the product to be very large. You'll need to check that your computer can handle that.

9. Exponential smoothing

Exponential smoothing is the same as percentage smoothing. Each new day takes on the maximum smoothing value and reduces all the old data by the same percentage. It's the same as a company issuing 10% new stock. All the old stock is reduced by 10%. The problem I have with exponential smoothing is that the trend always turns when the price goes through the trendline.

$$EXP(t) = EXP(t–1) + s*[p(t) – EXP(t–1)]$$

where *s* is the percentage smoothing constant.

In most programs, you can enter the number of days to smooth, so that *N*=10 days becomes *s*=2/(*N*+1) or *s*=0.18. But it is not necessary to use whole days. Any fraction

Finding the signal, or the trend, allows us to take advantage of a price move by ignoring the unnecessary noise.

between 0 and 1 will work. One of the advantages of exponential smoothing is that you can change the smoothing factor each day to create an adaptive moving average. The program in the sidebar uses whole days but you can change that easily.

10. Linear regression

For most analysts, a linear regression means using the function SLOPE. On a chart, the linear regression may look like a straight line but we are only interested in the angle of the line at the current point. This may be more difficult on a spreadsheet, but we'll just use the function SLOPE. We are long when it's positive and short when it's negative.

You can use the trend of the slope—that is, when today's slope is greater than yesterday's slope, we are long. The opposite for shorts. It increases the number of trades and may work for shorter trends. To use it that way, you'll need to make a small change to the program.

Comment

I've indicated some options in most of the smoothing choices, but for our test, I've kept it simple. I think the results will still prove interesting.

	SPY Using 110 Days (Long Only)	Net Profit	Profit Factor	Number of Trades	% Profitable Trades	Maximum Drawdown	Ratio Profit to Drawdown
1	Moving Average	$16,010	2.70	108	38.0%	($3,252)	4.92
2	Average-Off	$11,750	1.92	109	31.2%	($3,534)	3.32
3	**Weighted**	**$17,622**	**2.61**	**141**	**58.9%**	**($3,883)**	**4.54**
4	Triangular	$11,140	1.50	229	52.8%	($5,021)	2.22
5	Pivot-Point	$4,410	1.20	227	39.2%	($6,381)	0.69
6	Standard Deviation	$16,632	2.87	108	38.9%	($2,867)	5.80
7	**Median**	**$19,470**	**5.15**	**29**	**51.7%**	**($4,946)**	**3.94**
8	Geometric	$16,010	2.70	108	38.0%	($3,252)	4.92
9	Exponential	$11,814	1.78	168	32.1%	($4,944)	2.39
10	Linear Regression	$11,431	2.67	27	40.7%	($3,998)	2.86

DATA: YAHOO FINANCE

FIGURE 1: RESULTS OF TESTING SPY WITH A 110-DAY CALCULATION PERIOD. The 110-day calculation period was chosen as the best trend length to use for the S&P 500 market, so this trend length was used to test the 10 smoothing methods. Data from 2000 through July 2025 was used for the tests. The standard deviation smoothing method (the row in red text) gave the best result based on returns and drawdown. The weighted and median methods (the rows in bold text) also showed good results.

	SPY Using 20 Days (Long Only)	Net Profit	Profit Factor	Number of Trades	% Profitable Trades	Maximum Drawdown	Ratio Profit to Drawdown
1	Moving Average	$10,196	1.41	305	44.6%	($5,343)	1.91
2	Average-Off	$9,837	1.44	297	36.0%	($3,837)	2.56
3	**Weighted**	**$11,925**	**1.38**	**374**	**52.9%**	**($4,540)**	**2.63**
4	Triangular	$10,768	1.27	525	55.1%	($2,705)	3.98
5	Pivot-Point	$9,440	1.25	526	44.5%	($3,819)	2.47
6	Standard Deviation	$10,252	1.41	307	45.0%	($5,343)	1.92
7	**Median**	**$11,924**	**1.60**	**139**	**43.9%**	**($4,847)**	**2.46**
8	Geometric	$10,196	1.41	305	44.6%	($5,343)	1.91
9	Exponential	$6,723	1.20	416	35.8%	($5,256)	1.28
10	Linear Regression	$9,828	1.52	145	49.0%	($3,576)	2.75

FIGURE 2: RESULTS OF TESTING SPY WITH A 20-DAY CALCULATION PERIOD. What happens when a 20-day calculation period is used instead of a 110-day trend length? Does it improve results, particularly for the triangular and pivot-point smoothing approaches, which look for a turning point in price? A shorter calculation period seems to help the triangular and to some degree the pivot-point method.

Comparing the 10 methods

Comparing these methods will be tricky, because each may have a different "best window." If you like one of these methods, try testing it on a number of stocks or futures markets using the program in the sidebar. Or create a spreadsheet, which can be done for most of these methods.

I've put all 10 of these methods into one program. Select the strategy using the numbers assigned at the beginning of the program.

Because we're only testing the SPY and the Euro-Bund, we are only interested in long positions. I've had experience testing and trading both, and the short sales simply don't work over time. In the following tests, there was no market in which the short sales were profitable. Yes, 2008 would have been a good short sale for the SPY, but it has given it all back since then.

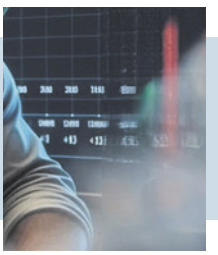
These tests start at 2000 and go through July 2025.

Test results

In Figure 1, the results of the 110-day trend, which was chosen as the best trend for the S&P, is applied to the 10 strategies. The line in red (the standard deviation smoothing method) was my choice based on the returns and drawdown. The lines in bold (weighted and median) are two other good choices.

What happens when we test a 20-day calculation period? I thought it would improve the triangular and pivot-point approaches, which look for a turning point in the prices. The results are in Figure 2.

Everyone using the same method can lead to sudden jumps in price and poor fills.



SMOOTHING THE DATA, IN EASYLANGUAGE

The following code gives you a means to calculate and test the 10 smoothing methods discussed in this article. Options are provided for some of the smoothing methods. Note that you can modify the code to try out variations and other markets, as well as to optimize, allow short sales, or change the data interval.

```
{ TSM Moving Average : Moving average system
  Copyright 2025 P J Kaufman. All rights reserved. }

{ strategy (1 - 10) as follows:
      1. Moving average
      2. Average-off
      3. Weighted moving average
      4. Triangular moving average
      5. Pivot point moving average
      6. Standard deviation moving average
      7. Moving median
      8. Geometric moving average
      9. Exponential moving average
      10. Linear regression
}
      input:    strategy(1), period(20), SDvalue(0.05),
usefutures(false), longonly(false);
      vars:     signal(0), psignal(0), trend(0), size(1), ix(0), INT
middle(0),
                value(0), SD(0), SDpercent(0), prod(0), root(0),
                smooth(0), newprice(0), sumofprices(0), sumof-
weights(0),
                INT start(0), stockinvestment(10000), futuresin-
vestment(25000),
                ATRper(20), equity(0), adate(" "), totalPL(0),
NAV(100),
                returns(0), cumreturn(0), todayPL(0), total-
Long(0), totalShort(0);
      array:    pricearray[150](0), sortprices[150](0), sortedar-
ray[150](0);

      if usefutures then
                size = futuresinvestment/(Avgtruerange(ATRper
)*bigpointvalue)
          else
                size = stockinvestment/close;

// 1. Moving Average
      if strategy = 1 then begin
          trend = average(close,period);
          end;

// 2. Average off
      if strategy = 2 then begin
          If Currentbar = 1 then
                trend = average(close,period)
          else
                trend = ((period - 1)*trend[1] + close)/period;
          end;

// 3. Weighted moving average
      if strategy = 3 then begin
          sumofprices = 0;
          sumofweights = 0;
          for ix = 0 to period - 1 begin
                sumofweights = sumofweights + ix + 1;
                sumofprices = sumofprices + (ix+1)*close[ix];
                end;
          trend = sumofprices/sumofweights;
          end;

// 4. Triangulation
      if strategy = 4 then begin
          middle = period/2;
// increase weights towards middle
          sumofprices = 0;
          sumofweights = 0;
// prices increase towards middle
          for ix = 0 to middle begin
                sumofweights = sumofweights + ix + 1;
                sumofprices = sumofprices + (ix+1)*close[ix];
                end;
          value = 0;
          for ix = period to middle - 1 begin
                value = value + 1;
                sumofweights = sumofweights + value;
                sumofprices = sumofprices + value*close[ix];
                end;
          trend = sumofprices/sumofweights;
          end;

// 5. Pivot-Point moving average
      if strategy = 5 then begin
          start = period - period/3;
          sumofprices = 0;
          sumofweights = 0;
          value = start + 1;
          for ix = 0 to period - 1 begin
                value = value - 1;
                sumofweights = sumofweights + value;
                sumofprices = sumofprices + value*close[ix];
                end;
          trend = sumofprices/sumofweights;
          end;

// 6. Standard deviation moving average
      if strategy = 6 then begin
          SD = stddev(close,period);
          SDpercent = (SD - SD[1])/SD;
          trend = average(close,period) +
SDvalue*SDpercent;
          end;

// 7. Moving median
      if strategy = 7 then begin
// move prices to array
          for ix = 0 to period - 1 begin
                pricearray[ix + 1] = close[ix];
                end;
          value1 = sortarray(pricearray,period,1);
          trend = pricearray[period/2];
          end;
```

```
// 8. Geometric moving average
      if strategy = 8 then begin
            prod = 1;
            for ix = 0 to period - 1 begin
                  prod = prod*close[ix];
                  end;
            root = 1/period;
            trend = power(prod,root);
            end;

// 9. Exponential smoothing
      if strategy = 9 then begin
            smooth = 2/(period + 1);
            trend = trend[1] + smooth*(close - trend[1]);
            end;

// 10. Linear regression, slope increasing or decreasing
      if strategy = 10 then begin
            trend = linearregslope(close,period);
            if trend > 0 then begin
                        buy to cover all shares next bar on open;
                        buy size share next bar on open;
                        signal = 1;
                        end
                  else if trend < 0 then begin
                        sell all shares next bar on open;
                        if longonly = false then begin
                              sell short size share next bar on open;
                              end;
                        signal = -1;
                  end;
            end;

// common entry code
      if strategy <> 10 then begin
            if trend > trend[1] then signal = 1
                        else if trend < trend[1] then signal = -1;
// new long position
            if psignal <= 0 and signal > 0 then begin
                        buy to cover all shares next bar on open;
                        buy size shares next bar on open;
                        end
                  else if psignal > 0 and signal < 0 then begin
                        sell all shares next bar on open;
                        if longonly = false then begin
                              sell short size shares next bar on
open;
                        end;
                  end;
            end;

      psignal = signal;

      equity = netprofit + openpositionprofit;
      todayPL = equity - equity[1];
      if psignal > 0 then
                  totalLong = totalLong + todayPL
            else if psignal < 0 then
                  totalShort = totalShort + todayPL;
      totalPL = totalPL + equity - equity[1];
      returns = (equity - equity[1])/stockinvestment;
      cumreturn = cumreturn + returns;
      if usefutures = false then
                  NAV = 100
            else
                  NAV = NAV[1]*(1 + returns);

      adate = ELdatetostring(date);
      If Currentbar = 1 then print(file("c:\TradeStation\
Smoothing.csv"),
                  "Date,Close,Signal,Size,TotalShort,TotalLong,To
talPL,Returns,CumReturn,NAV");
      print(file("c:\TradeStation\Smoothing.csv"),adate, ",",
close:8:6, ",",
                  signal:5:0, ",", size:8:3, ",", totalShort:8:0, ",", total-
Long:8:0, ",",
                  totalPL:8:0, ",", returns:6:4, ",", cumreturn:8:4, ",",
NAV:9:4);
```

	SPY Using 5 Days (Long Only)	Net Profit	Profit Factor	Number of Trades	% Profitable Trades	Maximum Drawdown	Ratio Profit to Drawdown
4	Triangular	$5,843	1.10	829	50.3%	($5,803)	1.01
5	Pivot-Point	($853)	0.99	1,056	46.1%	($11,042)	-0.08

FIGURE 3: RESULTS OF TESTING SPY WITH A 5-DAY CALCULATION PERIOD. Here, the test from Figure 2 is repeated for the triangular and pivot point smoothing methods, this time using only a 5-day calculation period. Results were worse.

The upshot is that smoothing works best over a longer period. However, the triangular method was my choice, with the median second. So a shorter period helped the triangular and to some degree the pivot point. Based on that, I reduced the calculation period to 5 days and retested both the triangular and pivot point. Results are in Figure 3.

It seemed good in theory, but the results were poor. I believe these techniques are best incorporated into another strategy. Some techniques are not meant to be standalone systems.

For the Euro-Bund, we'll use a 60-day trend, which was best for the simple moving average. Results are shown in Figure 4.

I've marked in red that the median is the best choice, with the simple moving average and linear regression as the next best. There is quite a variation in the percentage of profitable trades.

COMPARING THE BEST RESULTS

While we can choose our selection, I've charted the results of the best 110-day SPY smoothing methods in Figure 5. While there is some variation, results are remarkably the same.

Using the best results from the 20-day SPY tests, we get Figure 6. The shorter calculation period causes the results

to be different. In this case, the median technique looks like the better choice.

For the results of the Euro-Bund (Figure 7), we see that the median is slightly better than the next two choices. Just as with the 110-day SPY tests, these strategies track closely.

	Bund Futures Using 60 Days (Long Only)	Net Profit	Profit Factor	Number of Trades	% Profitable Trades	Maximum Drawdown	Ratio Profit to Drawdown
1	Moving Average	$3,576,810	2.43	159	40.9%	($847,440)	4.22
2	Average-Off	$3,593,450	2.73	156	23.1%	($851,830)	4.22
3	Weighted	$3,255,740	1.83	218	53.7%	($887,200)	3.67
4	Triangular	$2,933,220	1.62	281	53.0%	($644,840)	4.55
5	Pivot-Point	$1,578,630	1.33	304	35.2%	($1,403,260)	1.12
6	Standard Deviation	$3,604,520	2.39	165	40.0%	($876,880)	4.11
7	Median	$3,795,150	4.06	56	55.4%	($812,670)	4.67
8	Geometric	$3,576,810	2.43	159	40.9%	($847,440)	4.22
9	Exponential	$3,554,540	2.28	225	28.9%	($836,080)	4.25
10	Linear Regression	$3,478,220	3.57	47	55.3%	($765,680)	4.54

FIGURE 4: RESULTS OF TESTING EURO-BUND USING A 60-DAY CALCULATION PERIOD. The Euro-Bund futures market (10-year note) is tested using a 60-day trend, which was found to be best calculation period for the simple moving average. The median method gave the best result here. The moving average and linear regression methods gave the next-best results, albeit there is a large variation in the percentage of profitable trades.

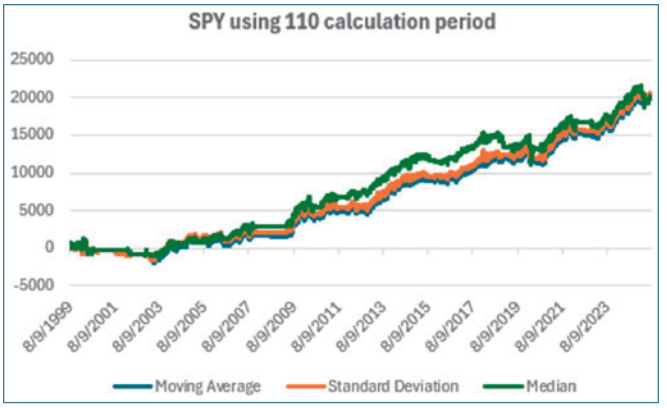


FIGURE 5: RESULTS OF TESTING SPY USING A 110-DAY CALCULATION PERIOD—BEST 3 RESULTS. The three best results from testing the 10 smoothing methods on SPY using a 110-day trend period are shown here. While there is some variation, results are very similar among them.

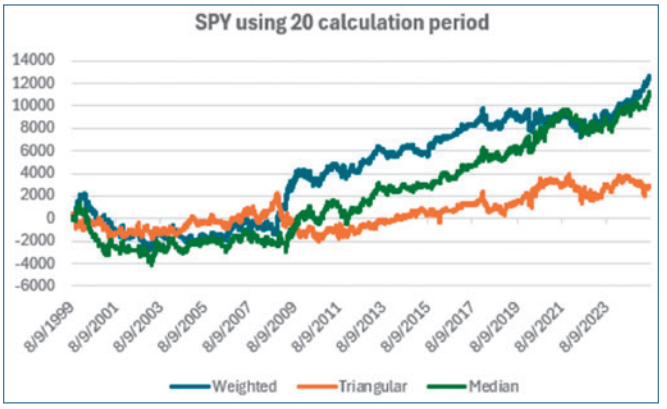


FIGURE 6: RESULTS OF TESTING SPY USING A 20-DAY CALCULATION PERIOD—BEST 3 RESULTS. The three best results from testing the 10 smoothing methods on SPY using a 20-day trend are shown. The median method had slightly better results than the next two choices, but they track closely.

CONCLUSION

I admit this is a small test. Only two markets and three calculation periods. But the long-term trend surfaces using the 110-day period no matter which strategy you use. But perhaps one method will get you in ahead of the moving average, which would be a trading advantage.

You can use the program in the sidebar to test other markets, optimize, allow short sales, and change the data interval. If you can program at all, you can make small changes in the coding to alter the strategies.

Perry J. Kaufman is a trader and financial engineer. He is the author of many books on trading and market analysis, including the sixth edition (2020) of Trading Systems and Methods *(with the first edition published in 1978 as a seminal book in the field of technical analysis), as well as* Kaufman Constructs Trading Systems *(2020), and* Learn To Trade *(2022). For questions or comments, please go to www.kaufmansignals.com.*

FURTHER READING

Kaufman, Perry J. [2022]. *Learn To Trade*, Amazon.

_____ [2020]. *Trading Systems and Methods*, 6th Edition, Wiley.

_____ [2020]. *Kaufman Constructs Trading Systems* (print and ebook editions), Amazon.

†TradeStation

†See Editorial Index

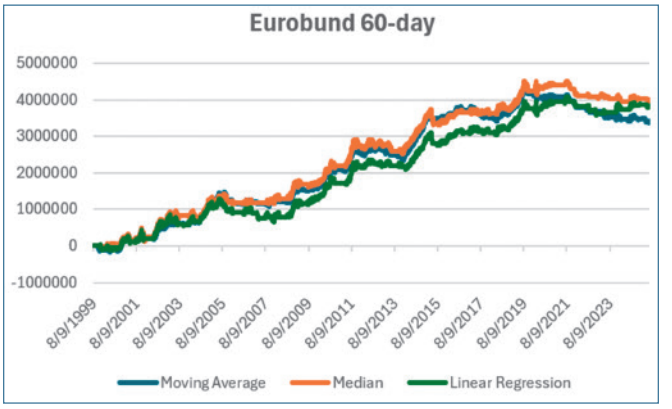


FIGURE 7: RESULTS OF TESTING EURO-BUND USING A 60-DAY CALCULATION PERIOD—BEST 3 RESULTS. The median method had slightly better results than the next two choices.