Open in app Get started

Published in DataDrivenInvestor

You have **2** free member-only stories left this month. Sign up for Medium and get an extra one

Sofien Kaabar, CFA Follow

Apr 16, 2021 · 12 min read · ✨ · Listen

Save

Combining the SuperTrend Indicator With Moving Averages in a Trading Strategy.

Can the SuperTrend Indicator and Moving Averages Provide Good Signals Together?

www.pxfuel.com

The Countdown Indicator is a contrarian indicator that I have developed for mostly

Open in app Get started

we will use signals from the Countdown Indicator and the Fibonacci moving average. The first two parts will introduce the indicators and the third part will present the strategy. The research is as always done in Python.

I have just published a new book after the success of New Technical Indicators in Python. It features a more complete description and addition of complex trading strategies with a Github page dedicated to the continuously updated code. If you feel that this interests you, feel free to visit the below link, or if you prefer to buy the PDF version, you could contact me on Linkedin.

The Book of Trading Strategies

Amazon.com: The Book of Trading Strategies: 9798532885707: Kaabar, Sofien: Books

www.amazon.com

The Concept of Moving Averages

Moving averages help us confirm and ride the trend. They are the most known technical indicator and this is because of their simplicity and their proven track record of adding value to the analyses. We can use them to find support and resistance levels, stops and targets, and to understand the underlying trend. This versatility makes them an indispensable tool in our trading arsenal.

Open in app Get started

EURUSD hourly values with the 200-period simple moving average.

As the name suggests, this is your plain simple mean that is used everywhere in statistics and basically any other part in our lives. It is simply the total values of the observations divided by the number of observations. Mathematically speaking, it can be written down as:

$$SMA = \frac{A_1 + A_2 + \cdots + A_n}{n}$$

Open in app Get started

```
def ma(Data, lookback, what, where):

  for i in range(len(Data)):
      try:
        Data[i, where] = (Data[i - lookback + 1:i + 1, what].mean())

            except IndexError:
                pass
    return Data
```

To use it, we need to have an OHLC data array with an extra empty column. This can be done by using the following code:

```
# Defining the function that adds a certain number of columns
def adder(Data, times):

    for i in range(1, times + 1):

        z = np.zeros((len(Data), 1), dtype = float)
        Data = np.append(Data, z, axis = 1)

    return Data

# Adding 1 extra column
my_data = adder(my_data, 1)

# Calling the moving average function
my_data = ma(my_data, 200, 3, 4)
```

The above states that the moving average function will be called on the array named *my_data* for a lookback period of 200, on the column indexed at 3 (closing prices in an OHLC array). The moving average values will then be put in the column indexed at 4 which is the one we have added using the adder function.

Open in app Get started

USDCHF hourly values with the 200-period simple moving average.

Another even more dynamic moving average is the exponential one. Its idea is to give more weight to the more recent values so that it reduces the lag between the price and the average.

```
def ema(Data, alpha, lookback, what, where):

    # alpha is the smoothing factor
    # window is the lookback period
    # what is the column that needs to have its average calculated
    # where is where to put the exponential moving average

    alpha = alpha / (lookback + 1.0)
    beta  = 1 - alpha
```

Open in app Get started

```
    Data[lookback + 1, where] = (Data[lookback + 1, what] * alpha) +
(Data[lookback, where] * beta)

    # Calculating the rest of EMA
    for i in range(lookback + 2, len(Data)):
      try:
        Data[i, where] = (Data[i, what] * alpha) + (Data[i - 1, where]
* beta)

       except IndexError:
          pass
    return Data
```

The SuperTrend Indicator

The first concept we should understand before creating the SuperTrend indicator is volatility. We sometimes measure volatility using the Average True Range. Although the ATR is considered a lagging indicator, it gives some insights as to where volatility is right now and where has it been last period (day, week, month, etc.). But before that, we should understand how the **True Range** is calculated (the ATR is just the average of that calculation).

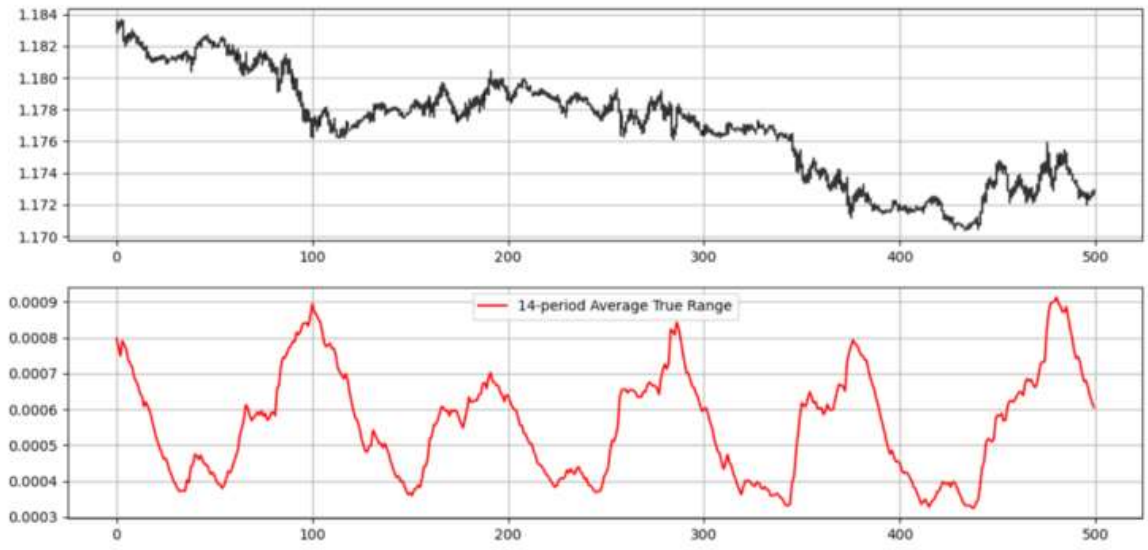
The true range is simply the greatest of the three price differences:

- **High — Low**
- **| High — Previous close |**
- **| Previous close — Low |**

Once we have got the maximum out of the above three, we simply take an average of n periods of the true ranges to get the Average True Range. Generally, since in periods of panic and price depreciation we see volatility go up, the ATR will most likely trend higher during these periods, similarly in times of steady uptrends or downtrends, the ATR will

Open in app Get started

```
def atr(Data, lookback, high, low, close, where):

    # TR
    for i in range(len(Data)):
      try:

        Data[i, where] = max(Data[i, high] - Data[i, low],
                             abs(Data[i, high] - Data[i - 1, close]),
                             abs(Data[i, low] - Data[i - 1, close]))

        except ValueError:
            pass

    Data[0, where] = 0
    Data = ema(Data, 2, lookback, where, where + 1)
    Data = deleter(Data, where, 1)
    Data = jump(Data, lookback)

    return Data
```



EURUSD M30 values in the first panel with the 14-period Average True Range in the second panel.

Now that we have understood what the ATR is and how to calculate it, we can proceed further with the SuperTrend indicator.

Open in app Get started

1. **Calculate the ATR using the function provided above.**
2. **Use the below formulas to calculate the SuperTrend.**
3. **Plot the indicator, create trading rules and analyze the results.**

The indicator seeks to provide entry and exit levels for trend followers. You can think of it as a moving average or an MACD. Its uniqueness is its main advantage and although its calculation method is much more complicated than the other two indicators, it is intuitive in nature and not that hard to understand. Basically, we have two variables to choose from. The ATR lookback and the multiplier's value. The former is just the period used to calculated the ATR while the latter is generally an integer (usually 2 or 3).

$$Basic\ Upperband = \left(\frac{High + Low}{2}\right) + (multiplier\ .\ eATR)$$

$$Basic\ Lowerband = \left(\frac{High + Low}{2}\right) - (multiplier\ .\ eATR)$$

The eATR above in the function is simply a normal ATR with a lookback multiplied by 2, then we subtract 1. But this is for illustrative purposes only and the only difference is the lookback period. You can use 5 or 500 if you wish.

The first thing to do is to average the high and low of the price bar, then we will either add or subtract the average with the selected multiplier multiplied by the ATR as shown in the above formulas. This will give us two arrays, the basic upper band and the basic lower band which form the first building blocks in the SuperTrend indicator. The next step is to calculate the final upper band and the final lower band using the below formulas.

$$Final\ Upperband = If\ Current\ Basic\ Upperband\ <\ Previous\ Final\ Upperband\ OR$$
$$Previous\ Close > Previous\ Final\ Upperband\ Then$$

Open in app Get started

$$Final\ Lowerband = If\ Current\ Basic\ Lowerband > Previous\ Final\ Lowerband\ OR\ Previous\ Close < Previous\ Final\ Lowerband\ Then\ Current\ Basic\ Lowerband\ Else\ Previous\ Final\ Lowerband$$

It may seem complicated but most of these conditions are repetitive and in any case, I will provide the Python code below so that you can play with the function and optimize it to your trading preferences. Finally, using the previous two calculations, we can find the SuperTrend.

$$SuperTrend = If\big((Previous\ SuperTrend = Previous\ Final\ Upperband)and\ (Current\ Close <= Current\ Final\ Upperband)\big)Then\ Current\ Final\ Upperband$$

$$If\big((Previous\ SuperTrend = Previous\ Final\ Upperband)and\ (Current\ Close > Current\ Final\ Upperband)\big)\ Then\ Current\ Final\ Lowerband$$

$$Else$$

$$If\big((Previous\ SuperTrend = Previous\ Final\ Lowerband)and\ (Current\ Close >= Current\ Final\ Lowerband)\big)Then\ Current\ Final\ Lowerband$$

$$Else$$

$$If\big((Previous\ SuperTrend = Previous\ Final\ Lowerband)and\ (Current\ Close < Current\ Final\ Upperband)\big)Then\ Current\ Final\ Upperband$$

```
def deleter(Data, index, times):
    for i in range(1, times + 1):
        Data = np.delete(Data, index, axis = 1)
    return Data
def supertrend(Data, multiplier, lookback):
    for i in range(len(Data)):
            # Average Price
```

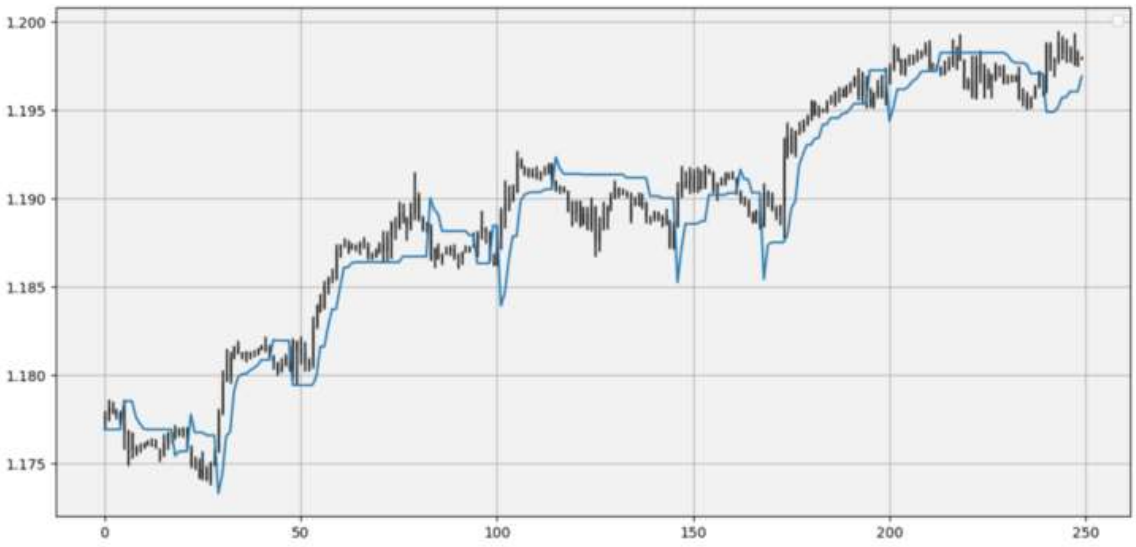
Open in app Get started

```
    # Final Upper Band
    for i in range(len(Data)):

        if i == 0:
            Data[i, 8] = 0

        else:
            if (Data[i, 6] < Data[i - 1, 8]) or (Data[i - 1, 3] >
Data[i - 1, 8]):
                Data[i, 8] = Data[i, 6]

            else:
                Data[i, 8] = Data[i - 1, 8]

    # Final Lower Band
    for i in range(len(Data)):

        if i == 0:
            Data[i, 9] = 0

        else:
            if (Data[i, 7] > Data[i - 1, 9]) or (Data[i - 1, 3] <
Data[i - 1, 9]):
                Data[i, 9] = Data[i, 7]

            else:
                Data[i, 9] = Data[i - 1, 9]

    # SuperTrend
    for i in range(len(Data)):

        if i == 0:
            Data[i, 10] = 0

        elif (Data[i - 1, 10] == Data[i - 1, 8]) and (Data[i, 3] <=
Data[i, 8]):
            Data[i, 10] = Data[i, 8]

        elif (Data[i - 1, 10] == Data[i - 1, 8]) and (Data[i, 3] >
Data[i, 8]):
            Data[i, 10] = Data[i, 9]

        elif (Data[i - 1, 10] == Data[i - 1, 9]) and (Data[i, 3] >=
Data[i, 9]):
            Data[i, 10] = Data[i, 9]
```

Open in app Get started

```
    Data = deleter(Data, 5, 5)

    return Data
```

Applying the above function on an OHLC array will give us something like the below when we plot it. The way we should understand the indicator is that when it goes above the market price, we should be looking to short and when it goes below the market price, we should be looking to go long as we anticipate a bullish trend. Remember that the SuperTrend is a trend-following indicator. The aim here is to capture trends at the beginning and to close out when they are over.



Example of the SuperTrend indicator on EURUSD pair

If you are also interested by more technical indicators and using Python to create strategies, then my best-selling book on Technical Indicators may interest you:

New Technical Indicators in Python

Amazon.com: New Technical Indicators in Python: 9798711128861: Kaabar, Mr Sofien: Books

Open in app Get started

Creating the Combined Strategy

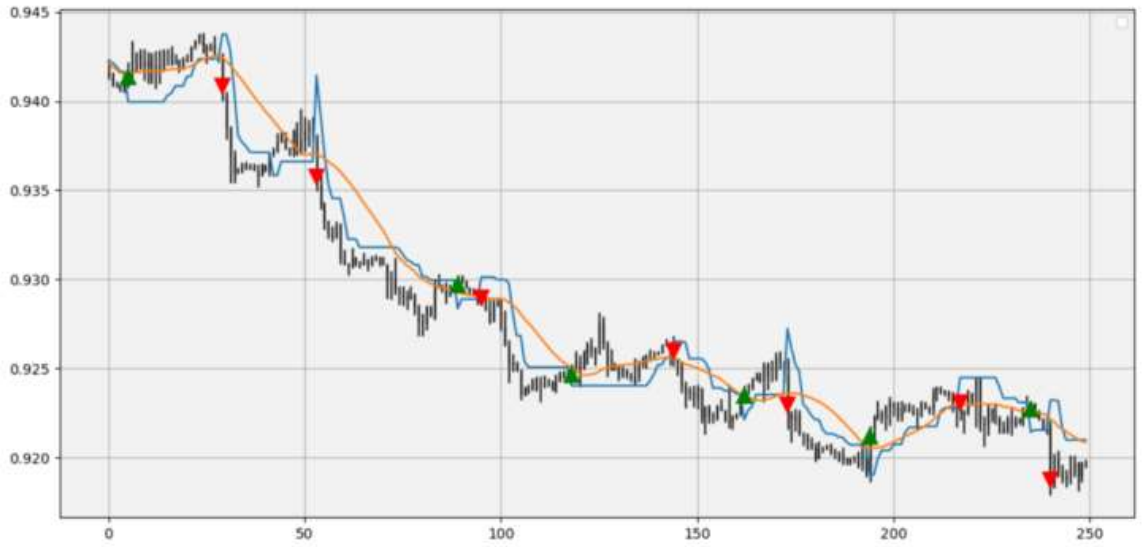
Combining indicators and techniques is the first step towards a robust trading system as getting a confluence of signals around a time period reinforces the conviction. There are many ways to create such structured strategies namely, **scorecards, indices, and simultaneous conditions**. The strategy we will be discussing will revolve around the simultaneous conditions and it is when certain conditions are met at the same time thus giving us a trading trigger.

The conditions for the combined strategy are as follows:

- **Go long (Buy) whenever the market price turns above the SuperTrend Indicator while being above the moving average. Hold this position until getting a new signal.**
- **Go short (Sell) whenever the market price turns below the SuperTrend Indicator while being below the moving average. Hold this position until getting a new signal.**

Open in app Get started

```
def signal(Data, close, super_trend_col, ma_col, buy, sell):

    for i in range(len(Data)):

        if Data[i, close] > Data[i, super_trend_col] and Data[i - 1,
close] < Data[i - 1, super_trend_col] and Data[i, close] > Data[i,
ma_col]:
            Data[i, buy] = 1

        if Data[i, close] < Data[i, super_trend_col] and Data[i - 1,
close] > Data[i - 1, super_trend_col] and Data[i, close] < Data[i,
ma_col]:
            Data[i, sell] = -1
```



Signal chart on the USDCHF following the strategy.

Even though the signals above show superior quality signals, the back-testing results were disappointing (lookback of 10 on the SuperTrend with a multiplier of 2 and a lookback of 21 on the moving average), however, much optimization can be done to improve it. Until then, the strategy works well in discretionary trading:

Open in app Get started

Equity curves following the strategy.

If you are interested by technical indicators, feel free to have a look at the below article:

Combining the Countdown Indicator With the Fibonacci Moving Averages in a Trading Strategy.

Can the Countdown Indicator and Fibonacci Moving Averages Provide Good Signals Together?

medium.com

Conclusion

If you regularly follow my articles, you will find that many of the indicators I develop or optimize have a high hit ratio and on average are profitable. This is mostly due to the **risk management method** I use. But what about market randomness and the fact that many underperformers blaming Technical Analysis for their failure?

Open in app Get started

trading. Hence, I have no motive to publish biased research. My goal is to share back what I have learnt from the online community.

Remember to always do your back-tests. Even though I supply the indicator's function (as opposed to just brag about it and say it is the holy grail and its function is a secret), you should always believe that other people are wrong. My indicators and style of trading works for me but maybe not for everybody. I rely on this rule:

> *The market price cannot be predicted or is very hard to be predicted more than 50% of the time. But market reactions can be predicted.*

What the above quote means is that we can form a small zone around an area and say with some degree of confidence that the market price will show a reaction around that area. **But we cannot really say that it will go down 4% from there, then test it again, and breakout on the third attempt to go to $103.85. The error term becomes exponentially higher because we are predicting over predictions.**

While we are discussing this topic, I should point out a few things about my back-tests and articles:

- **The spread I use is based on institutional quotes of a small pip fraction. Generally, retail traders are given a whopping spread of 1–2 pips per trade. This is huge and unfair to them. I use 0.2–0.5 spread. However, most of the strategies that use the hourly time frame still work with 1 pip spread. For the ones that use M15 or M5 time frames, they cannot be profitable with a spread of 1 pip.**
- **The holding period calculation I use is close-to-close in case there is no risk management process.**
- **Although I discourage trading based on just one indicator, the numbers do not lie. What I am presenting is what could have happened when taking into account a low spread.**
- **Some of the back-tests I provide are losers and they are published either to**

Open in app Get started

- **Finally, I am a firm believer of not spoon-feeding the learners. I have learnt by doing and not by copying. You should get the idea, the function, the intuition, the conditions of the strategy, and then elaborate (an even better) one yourself so that you back-test it and improve it before deciding to take it live or to eliminate it.**

To sum up, are the strategies I provide realistic? ***Yes***, but only by optimizing the environment (robust algorithm, low costs, honest broker, proper risk management, and order management). Are the strategies provided only for the sole use of trading? ***No,*** *it is to stimulate brainstorming and getting more trading ideas as we are* ***all sick of hearing about an oversold RSI as a reason to go short or a resistance being surpassed as a reason to go long. I am trying to introduce a new field called Objective Technical Analysis where we use hard data to judge our techniques rather than rely on outdated classical methods.***

Get an Email Whenever I Publish a New Story.

By signing up, you will create a Medium account if you don't already have one. Review our Privacy Policy for more information about our privacy practices.

Subscribe

Open in app Get started

Download on the App Store GET IT ON Google Play