CHAPTER 11

A New Breed of Technical Indicators

This chapter presents the best indicators from my personal library of indicators—*K's collection*—that I have developed over the years with the aim of diversifying and improving the market predictive capabilities. K's collection is different from the rainbow indicators presented in Chapter 3 in the sense that the indicators of the former are more sophisticated, while the latter can mostly be considered as techniques applied to indicators. Classic technical indicators like moving averages remain, by default, the most monitored by traders, and thus, potentially have a stronger market impact in case they show a clear configuration. Nevertheless, modern indicators, especially the ones from K's collection, seek to improve classic technical analysis by combining indicators or by using new techniques to forecast market reactions and deliver a useful piece of information about the market's state. Each section of this chapter will present an indicator from K's collection.

K's Reversal Indicator I

This indicator will give us the chance to discuss a popular classic technical indicator before we present it. K's reversal indicator I is composed of Bollinger bands and the moving average convergence divergence (MACD) oscillator. The MACD is a classic technical indicator used to identify changes in the strength and direction of a trend. It consists of three components:

- MACD line: This is the difference between a 12-period EMA and a 26-period EMA.
- Signal line: This is a 9-period EMA applied to the MACD line.
- Histogram: This is the difference between the MACD line and the signal line.

Crossovers between the MACD line and signal line are the most common signals watched. Use the following code to create the MACD given an OHLC data frame.

```
def macd(my_time_series, source='close', short_window=12, long_window=26, signal_window=9):
  # calculate the short-term EMA
  my_time_series['EMA_short'] = my_time_series[source].ewm(span=short_window,
                                                           adjust=False).mean()
  # calculate the long-term EMA
  my_time_series['EMA_long'] = my_time_series[source].ewm(span=long_window, adjust=False).mean()
  # calculate the MACD line
  my_time_series['MACD_line'] = my_time_series['EMA_short'] - my_time_series['EMA_long']
  # calculate the Signal line
  my_time_series['MACD_signal'] = my_time_series['MACD_line'].ewm(span=signal_window,
                                                                  adjust=False).mean()
  # calculate the MACD Histogram
  my_time_series['MACD_histogram'] = my_time_series['MACD_line'] - my_time_series['MACD_signal']
  # drop the EMA columns as they are not needed anymore
  my_time_series.drop(['EMA_short', 'EMA_long'], axis=1, inplace=True)
  return my_time_series.dropna()
```

Figure 11.1 shows how the MACD looks with respect to the time series it represents.

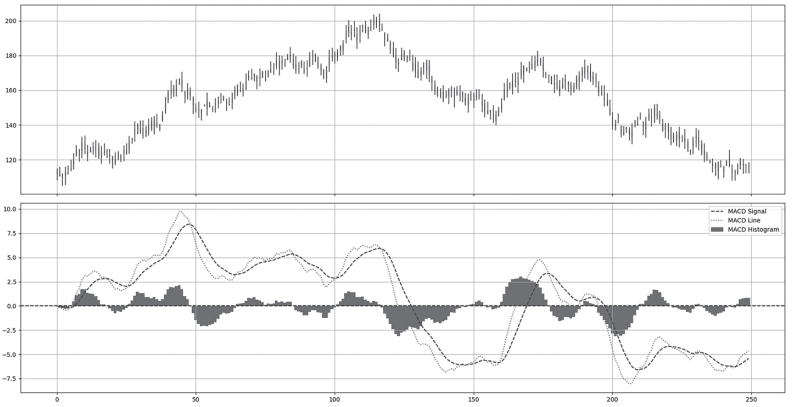

Figure 11.1 MACD

The MACD can be used as a trend-following indicator. When the MACD line crosses above the signal line, it generates a bullish signal, suggesting upward momentum. When the MACD line crosses below the signal line, it generates a bearish signal, suggesting downward momentum. Similarly, when the MACD line crosses above the zero line, it suggests upward momentum, and when it crosses below the zero line, it signals a

shift toward downward momentum. Let's go back to the discussion on K's reversal indicator I. The signals are found through a combination of the MACD and Bollinger bands. Here's how:

- A bullish signal is generated whenever the smaller value between the open and close price is below the 100-period lower Bollinger band. Simultaneously, the MACD line must cross over the MACD signal.
- A bearish signal is generated whenever the larger value between the open and close price is above the 100-period upper Bollinger band. Simultaneously, the MACD line must cross under the MACD signal.

Therefore, in simple terms, the signal is given by combining a reversal indicator (Bollinger bands) with a trend-following technique (the MACD signal lines cross). This is a hybrid technique that has the potential to yield very interesting directional signals. The bullish signal is found by an extremely oversold condition with the market price below the lower Bollinger band, combined with a bullish continuation confirmation stemming from the MACD line. In parallel, the bearish signal is found by an extremely overbought condition with the market price above the upper Bollinger band, combined with a bearish continuation confirmation stemming from the MACD line. The following code block shows how to define K's reversal indicator I.

```
def k_reversal_indicator_I(my_time_series):
  my_time_series = macd(my_time_series, source='close', short_window=12, long_window=26, signal_window=9)
  my_time_series = bollinger_bands(my_time_series, source='close', bb_lookback=100, num_std_dev=2)
  my_time_series['bullish_signal'] = 0
  my_time_series['bearish_signal'] = 0
  for i in range(0, len(my_time_series)):
    # bullish signal
    if my_time_series['low'].iloc[i] < my_time_series['lower_band'].iloc[i] and \
      my_time_series['high'].iloc[i] < my_time_series['middle_band'].iloc[i] and \
      my_time_series['MACD_line'].iloc[i] > my_time_series['MACD_signal'].iloc[i] and \
      my_time_series['MACD_line'].iloc[i-1] < my_time_series['MACD_signal'].iloc[i-1]:
       my_time_series.at[my_time_series.index[i+1], 'bullish_signal'] = 1
    # bearish signal
    elif my_time_series['high'].iloc[i] > my_time_series['upper_band'].iloc[i] and \
      my_time_series['low'].iloc[i] > my_time_series['middle_band'].iloc[i] and \
      my_time_series['MACD_line'].iloc[i] < my_time_series['MACD_signal'].iloc[i] and \
      my_time_series['MACD_line'].iloc[i-1] > my_time_series['MACD_signal'].iloc[i-1]:
       my_time_series.at[my_time_series.index[i+1], 'bearish_signal'] = 1
  return my_time_series
```

Figure 11.2 shows a few signals generated by the indicator.

Figure 11.2 A signal chart showing K's reversal indicator I

The best practices to follow while using this indicator are as follows:

- Ideally, as with any reversal indicator (classic or modern), it is best used during a sideways market. This ensures that the balance between supply and demand favors more precision.
- The indicator has an inherent confirmation factor in it, which is the MACD crossing over or under the signal line; therefore, there is no need for an extra filter to validate the signals.
- The lookback periods of the indicator are the result of a large number of experiments and are theoretically suitable for all types of time series. Therefore, even though you are encouraged to tweak the parameters (e.g., change the 100-period lookback), it is important to consider that it is the result of a generalized optimization effort.

In conclusion, K's reversal indicator I is a powerful reversal tool in modern technical analysis. It combines the strengths of Bollinger bands and the MACD to predict market inflections.

K's Reversal Indicator II

If I have to select a favorite among all the indicators of K's collection, then my choice goes to K's reversal indicator II. This does not in any way compromise the quality of the other indicators, but simply points to how well

this indicator does a good job of determining local tops and bottoms. K's reversal indicator II has no relation whatsoever to K's reversal indicator I, as it is based on a combination of price, time, and mean reversion. Let's see how to develop this indicator:

1. Calculate a moving average (by default, a 13-period moving average).
2. Calculate the number of times the market is above its moving average. Whenever that number hits 21, a bearish signal is generated, and whenever that number hits zero, a bullish signal is generated.

The following code block shows how to define K's reversal indicator II.

```
def k_reversal_indicator_I(my_time_series):
  my_time_series = macd(my_time_series, source='close', short_window=12, long_window=26, signal_window=9)
  my_time_series = bollinger_bands(my_time_series, source='close', bb_lookback=100, num_std_dev=2)
  my_time_series['bullish_signal'] = 0
  my_time_series['bearish_signal'] = 0
  for i in range(0, len(my_time_series)):
    # bullish signal
    if my_time_series['low'].iloc[i] < my_time_series['lower_band'].iloc[i] and \
      my_time_series['high'].iloc[i] < my_time_series['middle_band'].iloc[i] and \
      my_time_series['MACD_line'].iloc[i] > my_time_series['MACD_signal'].iloc[i] and \
      my_time_series['MACD_line'].iloc[i-1] < my_time_series['MACD_signal'].iloc[i-1]:
       my_time_series.at[my_time_series.index[i+1], 'bullish_signal'] = 1
    # bearish signal
    elif my_time_series['high'].iloc[i] > my_time_series['upper_band'].iloc[i] and \
      my_time_series['low'].iloc[i] > my_time_series['middle_band'].iloc[i] and \
      my_time_series['MACD_line'].iloc[i] < my_time_series['MACD_signal'].iloc[i] and \
      my_time_series['MACD_line'].iloc[i-1] > my_time_series['MACD_signal'].iloc[i-1]:
       my_time_series.at[my_time_series.index[i+1], 'bearish_signal'] = 1
  return my_time_series
```

Figure 11.3 shows a few signals generated by the indicator.

Figure 11.3 A signal chart showing K's reversal indicator II

You can see how this indicator has more frequent signals than its predecessor. This can be an advantage to opportunistic traders. The best practices to follow using this indicator are as follows:

- It is best used during a ranging (sideways) market for the same reasons mentioned previously in the section on K's reversal indicator I.
- The indicator takes as variables: time, price, and a price-derived calculation (SMA). It is therefore a three-dimensional measure that aims to detect hidden market maxima and minima. This means that it is quite possible to use it as a stand-alone indicator, something that is discouraged with other indicators.
- The lookback periods of the indicator are the result of a large number of experiments and are theoretically suitable for all types of time series. As with K's reversal indicator I, you can tweak the parameters, but keep in mind the default lookback periods.

In conclusion, K's reversal indicator II can be categorized as a market directional system as opposed to just an indicator. It is a prime example of modern technical analysis, which fuses three elements (time, price, and a price-derived indicator) in order to detect market reactions.

K's ATR-Adjusted RSI

It is worth asking what happens when we try to fuse volatility and momentum. For example, if we take the ATR and try to fuse it with the RSI, what would be the result? K's ATR-adjusted RSI answers this question by properly combining both indicators into one that detects reversals while taking into account volatility. Follow these steps to create the indicator:

1. Calculate a 13-period RSI on the close price.
2. Calculate a 5-period ATR on the OHLC data.
3. Multiply the RSI calculation from step 1 by the ATR calculation from step 2.
4. Calculate a 13-period RSI of the result from step 3.

Use the following code snippet to create the indicator's function.

```
def atr_adjusted_rsi(my_time_series, source='close', rsi_lookback=13, vol_lookback=5, rsi_atr_lookback=13):
  my_time_series = rsi(my_time_series, source='close', output_name='RSI', rsi_lookback=rsi_lookback)
  my_time_series = atr(my_time_series, vol_lookback=vol_lookback)
  my_time_series['RSI_times_atr'] = my_time_series['RSI'] * my_time_series['volatility']
  my_time_series = rsi(my_time_series, source='RSI_times_atr', output_name='atr_adjusted_rsi',
rsi_lookback=rsi_atr_lookback)
  return my_time_series.dropna()
```

Figure 11.4 shows the ATR-adjusted RSI in action.

Figure 11.4 ATR-adjusted RSI in action

It's worth noting that you should tweak the lookback period of both components (RSI and ATR) to find the optimal one for you. I have chosen 13 as the default after a few optimizations, but markets are dynamic and change properties constantly; therefore, you are encouraged to play around with the parameters.

K's RSI²

So far, you have been applying the RSI's formula to the close price. This means that the source value of the RSI is the time series' close price. But what if the source price of the RSI was the RSI itself? The RSI² (pronounced RSI square) is a simple indicator I've developed to predict the

RSI's future direction by using the divergence technique. The steps required to calculate the RSI² are as follows:

- Calculate a 14-period RSI with the source value set to the close price.
- Calculate a 5-period RSI with the source value set to the RSI from step 1.

By now, you should be familiar with how we code indicators. Therefore, the following code snippet should not be complicated to understand.

```
def rsi_square(my_time_series, source='close', rsi_prime_lookback=14, rsi_square_lookback=5):
  my_time_series = rsi(my_time_series, source=source, output_name='RSI',
                       rsi_lookback=rsi_prime_lookback)
  my_time_series = rsi(my_time_series, source='RSI', output_name='RSI²',
                       rsi_lookback=rsi_prime_lookback)
  return my_time_series.dropna()
```

Previously, with the Yellow indicator, you have seen the slope divergence technique. K's RSI² will give you a chance to fully understand the real divergence technique. Divergence is different from slope divergence, and it is a means to spot early signs of a possible change in the ongoing trend. Imagine you're walking uphill (price is going up), but you're getting more tired with each step (momentum is going down). At some point, you'll probably stop or turn around. That's divergence: the price is doing one thing (e.g., going higher), but the strength behind it (measured by indicators like the RSI or MACD) is doing the opposite. It often signals that the trend may soon reverse or slow down. There are two main types:

- Bullish divergence: Price makes lower lows, but the indicator makes higher lows. This indicates that the market might go up soon.
- Bearish divergence: Price makes higher highs, but the indicator makes lower highs. This indicates that the market might go down soon.

When RSI² is showing higher lows while the RSI is showing lower lows, a bullish divergence is confirmed. In contrast, when RSI² is showing

lower highs while the RSI is showing higher highs, a bearish divergence is confirmed. The divergence technique plays on the strong correlation between the RSI and the underlying security price. The hypothesis is that if it's possible to forecast reversals in the RSI, then it may be possible to relate these reversals to the security's price. Figure 11.5 shows the RSI².

Figure 11.5 RSI² (dashed) versus regular RSI

Figure 11.6 shows divergence signals on RSI² versus RSI.

Figure 11.6 Divergence signals using RSI²

Using the divergence technique on the RSI² adds a fresh new way to detect market exhaustion. Note that originally, the divergence is to be