The Stata Journal (2017)
17, Number 3, pp. 736–747

Technical financial analysis tools for Stata

Mehmet F. Dicle
Loyola University New Orleans
College of Business
New Orleans, LA
mfdicle@gmail.com

John D. Levendis
Loyola University New Orleans
College of Business
New Orleans, LA

Abstract. In this article, we provide four financial technical analysis tools: moving averages, Bollinger bands, moving-average convergence divergence, and the relative strength index. The `tftools` command is used with four subcommands, each referring to a technical analysis tool: `bollingerbands`, `macd`, `movingaverage`, and `rsi`. We provide examples for each tool. `tftools` allows researchers to backtest their own investment strategies and will be of interest to investors, researchers, and students of finance.

Keywords: st0495, tftools bollingerbands, tftools macd, tftools movingaverage, tftools rsi, finance, technical analysis, moving average, Bollinger, MACD, RSI

1 Introduction

Technical financial analysis is part of the standard investment decision-making process. Many brokerage companies and public websites provide free tools to calculate various technical analysis statistics. However, investors often need to customize and backtest their analysis tools. The technical analysis tools we provide allow investors and researchers to perform some common technical analyses.

We provide one command with four different subcommands as part of this study: `tftools bollingerbands` calculates the Bollinger bands; `tftools macd` calculates the moving-average convergence and divergence (MACD); `tftools movingaverage` calculates simple and exponential moving averages (EMAs); and `tftools rsi` calculates the relative strength index (RSI).

Being able to easily calculate these statistics allows investors to better optimize their portfolios and allows researchers to test their financial hypotheses. For example, Chong and Ng (2008) find that the MACD and RSI trading rules generate better returns than a simple buy-and-hold strategy. Investigating the Singapore stock exchange, Wong, Manzur, and Chew (2003) found that RSI-based trading was quite profitable. Leung and Chong (2003) found that the Bollinger bands performed similarly to the "moving-average envelopes". For a review of these and other common technical financial indicators, see Cahen (2001), Kirkpatrick and Dahlquist (2010), and Stevens (2002), among others.

2 Bollinger bands: tftools bollingerbands

2.1 Title

`tftools bollingerbands` calculates the Bollinger bands for a single time-series variable.

2.2 Syntax

`tftools bollingerbands` [*if*] [*in*] `, symbol(`*variable*`) generate(`*newvar*`)`
[`period(`*integer*`) sdevs(`*string*`)`]

2.3 Description

Bollinger bands were developed by John Bollinger in the 1980s and consist of an upper and lower band, defining an envelope of usual trading. A stock price moving outside this envelope triggers an investment decision. The upper band is calculated as K standard deviations above the stock's moving average. The lower band is calculated as K standard deviations below the moving average. By construction, the center of this envelope is the original moving average. Bollinger bands mimic the idea that random variables tend to stay within plus or minus two standard deviations from their mean. To this insight, Bollinger bands add that the financial data might not be stationary, so the means and standard deviations are updated continually.

The usual trading system using Bollinger bands is to sell when the price reaches its upper band, because a reversion to the mean is expected. A contrarian Bollinger trading system reverses this logic: buy when the price crosses its upper band.

The `tftools bollingerbands` command calculates the Bollinger bands for a single time-series variable. The size of the window for the standard deviation and the simple moving average can be specified by the user. The command creates three new variables: *newvar*`_middle_band`, *newvar*`_upper_band`, and *newvar*`_lower_band`. The data must first be `tsset`.

2.4 Options

`symbol(`*variable*`)` specifies the variable that the Bollinger band calculation is based on (usually the stock symbol that contains the daily prices). `symbol()` is required.

`generate(`*newvar*`)` specifies the new variable prefix for calculated Bollinger band values. `generate()` is required.

`period(`*integer*`)` specifies the window for which standard deviation and simple moving average are calculated. The default is `period(20)`.

`sdevs(`*string*`)` specifies the factor that is multiplied by the standard deviation. The default is `sdevs(2)`.

2.5 Examples

Figure 1 shows the daily S&P 500 index along with the upper and lower Bollinger bands using the default two standard deviations. It is usually interpreted as support and resistance levels. When the index breaks above the upper band (resistance line), a downward move is expected. When the index breaks below the lower band (support line), an upward move is expected.

```
freduse SP500
drop if SP500==.
drop date
rename daten date
gen obs=_n
tsset obs
tftools bollingerbands if year(date)>2015, symbol(SP500) generate(SP500_BB)
twoway (line SP500 date) (line SP500_BB_upper_band date, lwidth(thick))   ///
       (line SP500_BB_lower_band date, lwidth(thick)) if year(date)>2015
```

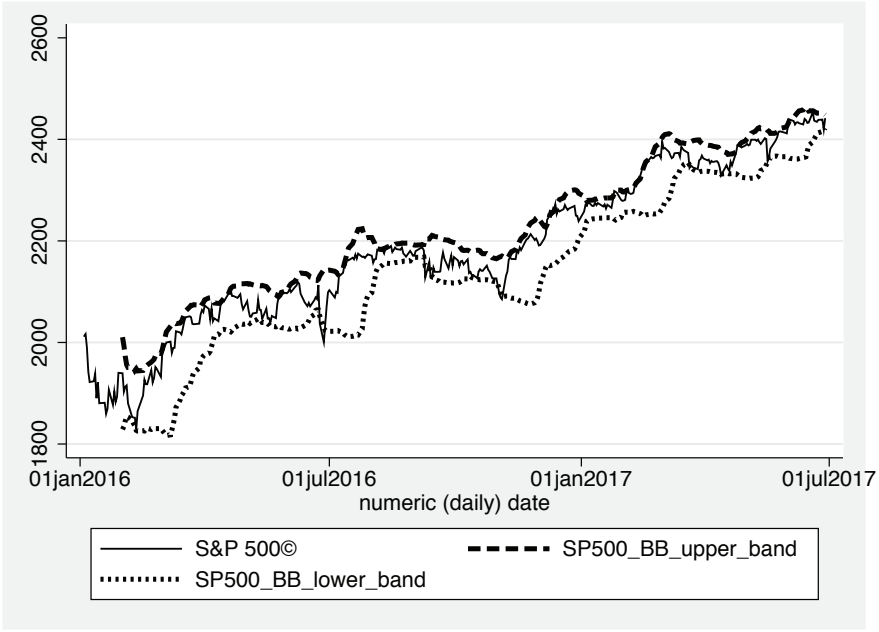

Figure 1. Daily S&P 500 index and Bollinger bands (two standard deviations)

Figure 2 shows the same Bollinger bands as figure 1; however, it has four standard deviations (instead of two). While with two standard deviations, the index line frequently breaks above and below the resistance and support lines, buy-and-sell signals are more rare—and presumably stronger—if a stock breaks through these much wider bands.

```
freduse SP500, clear
drop if SP500==.
drop date
rename daten date
gen obs=_n
tsset obs
tftools bollingerbands if year(date)>2015, symbol(SP500) generate(SP500_BB) ///
       sdevs(4)
twoway (line SP500 date) (line SP500_BB_upper_band date, lwidth(thick))     ///
       (line SP500_BB_lower_band date, lwidth(thick)) if year(date)>2015
```

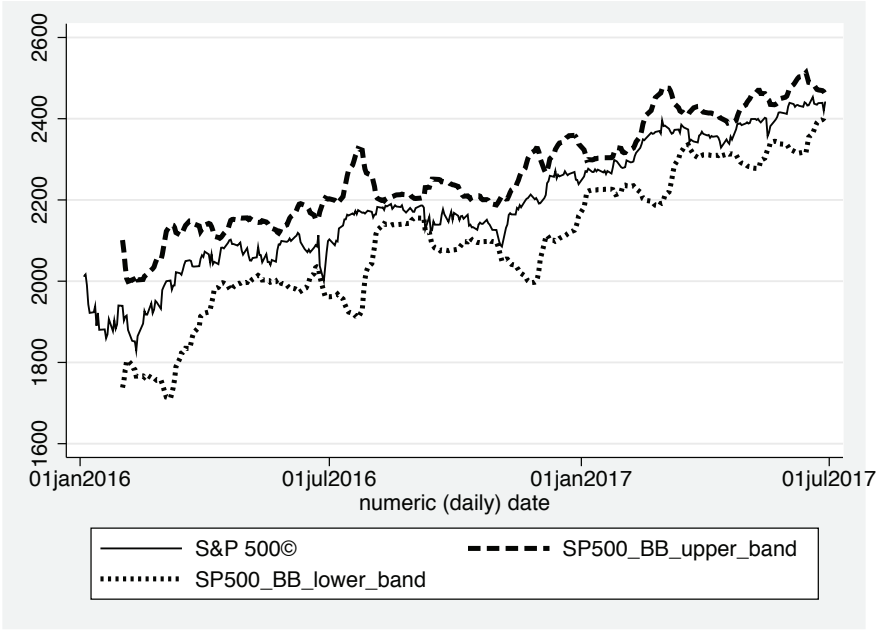

Figure 2. Daily S&P 500 index and Bollinger bands (four standard deviations)

3 MACD: tftools macd

3.1 Title

`tftools macd` calculates the MACD for a single time-series variable.

3.2 Syntax

`tftools macd` [*if*] [*in*], `symbol(`*variable*`) generate(`*newvar*`)`

3.3 Description

The MACD ("Mack-D") is a momentum indicator commonly used by technical analysts. The MACD is equal to the difference between 2 EMAs of different lengths: 1) a long-term EMA of usually 26 days and 2) a short-term EMA of usually 12 days. The MACD is defined as the short-term EMA minus the long-term EMA. A third component is the so-called signal line, which is usually a nine-day EMA of the MACD.

The convention is to plot the signal line over the MACD. A "bullish" indicator is when the signal line crosses above the MACD. A "bearish" indicator is when the signal line crosses below the MACD.

The MACD histogram is equal to the MACD less the signal line. As such, it oscillates above and below zero. A bullish signal is when the histogram becomes positive. A bearish signal is when the histogram becomes negative. The histogram is used as an indicator of momentum. When the histogram is positive and increasing, the stock price is said to have increasing positive momentum. A negative and decreasing histogram implies steeper downside momentum.

It is often viewed as signaling the end of a current trend when the actual stock price diverges significantly from the MACD. Finally, the MACD rises steeply when the short-term EMA (usually 12 days) is rising faster than the long-term EMA (usually 26 days). That is, prices have moved dramatically upward recently. Analysts sometimes infer that the current price has moved away from its equilibrium value, and they expect a downward correction.

The `tftools macd` command calculates the MACD for a single time-series variable. `tftools macd` creates three new variables: *newvar*`_MACD_line`, *newvar*`_signal_line`, and *newvar*`_MACD_histogram`. The data must first be `tsset`.

3.4 Options

`symbol(`*variable*`)` specifies the variable that the MACD calculation is based on (usually the stock symbol that contains the daily prices). `symbol()` is required.

`generate(`*newvar*`)` specifies the new variable prefix for the calculated MACD values. `generate()` is required.

3.5 Example

Figure 3 shows the daily S&P 500 index MACD. The usual interpretation for this technical analysis is based on the MACD line and the signal line. It is usually interpreted as a downward signal (bearish) if the MACD line falls below the signal line. Similarly, it is usually interpreted as an upward signal (bullish) if the MACD line goes above the signal line.

```
freduse SP500, clear
drop if SP500==.
drop date
rename daten date
gen obs=_n
tsset obs
tftools macd if year(date)>2015, symbol(SP500) generate(SP500)
twoway (line SP500_MACD_line date, lpattern(dash))            ///
       (line SP500_signal_line date) if year(date)>2015
```

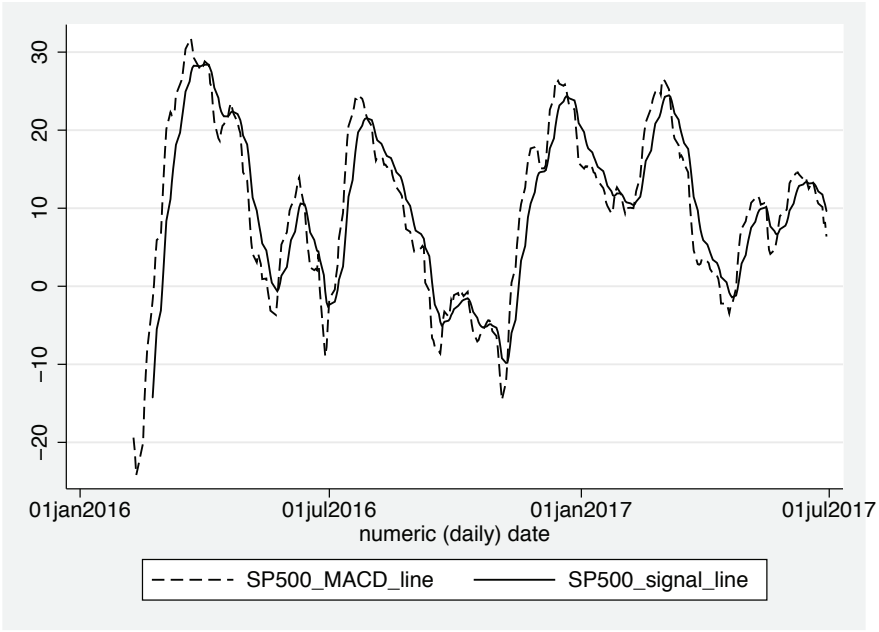

Figure 3. Daily S&P 500 index MACD

While figures 1 and 2 have no up or down signal (the most recent day's index values were in the middle of the upper and lower bands), figure 3 shows a down signal where the MACD line is below the signal line.

4 Moving average: tftools movingaverage

4.1 Title

`tftools movingaverage` calculates several different moving averages for a single time-series variable.

4.2 Syntax

`tftools movingaverage` [*if*] [*in*]`, symbol(`*variable*`) generate(`*newvar*`)`
`period(`*integer*`) ma_type(`*string*`)`

4.3 Description

`tftools movingaverage` calculates several different moving averages for a single time-series variable. The window for the moving averaging and type of averaging (that is, simple or exponential) can both be specified. You can also calculate moving standard deviation, maximum, minimum, and sum. `tftools movingaverage` creates a new variable. The data must first be `tsset`.

4.4 Options

`symbol(`*variable*`)` specifies the variable that the moving-average calculation is based on (usually the stock symbol that contains the daily prices). `symbol()` is required.

`generate(`*newvar*`)` specifies the new variable prefix for the calculated moving-average values. `generate()` is required.

`period(`*integer*`)` specifies the size of the moving-average window, expressed as an integer number of time periods. `period()` is required.

`ma_type(`*string*`)` specifies the moving-average type: `sma` (simple moving average), `ema` (EMA), `sd` (moving standard deviation), `sum` (moving sum), `min` (moving minimum), or `max` (moving maximum). `ma_type()` is required.

4.5 Examples

Figure 4 shows the simple moving average for the daily S&P 500 index. Moving averages are the building blocks of the financial technical analysis. It is common to use 20 and 50 days for short-term moving averages and 100 and 200 days for long-term moving averages.

```
freduse SP500, clear
drop if SP500==.
drop date
rename daten date
gen obs=_n
tsset obs
tftools movingaverage if year(date)>2015, symbol(SP500) generate(SP500) ///
    period(100) ma_type(sma)
twoway (line SP500 date) (line SP500_sma_100 date) if year(date)>2015
```

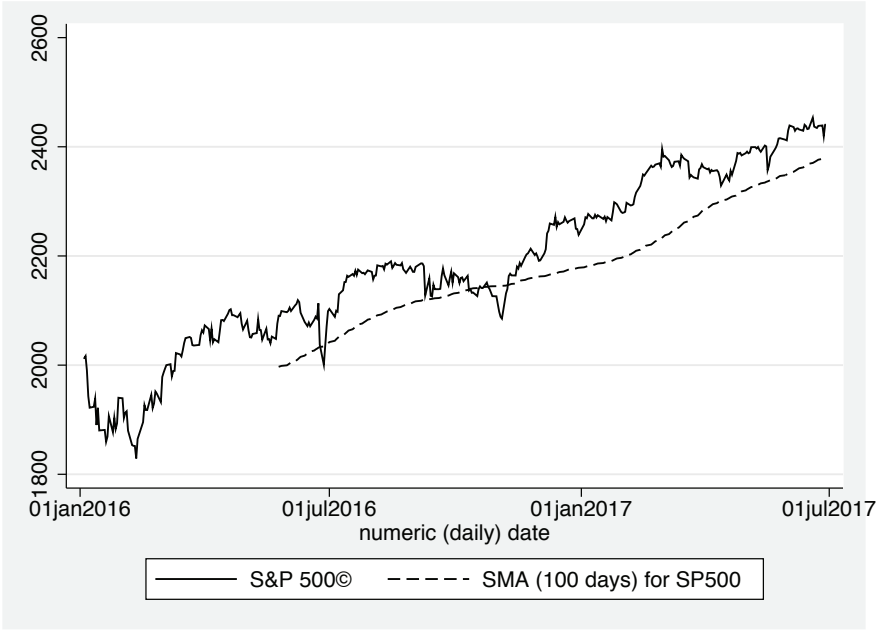

Figure 4. Daily S&P 500 index and 100 days simple moving average

Figure 5 shows short-term (50 days) and long-term (200 days) moving averages. These two, in combination, may signal two of the most common technical analysis signals, namely, the so-called golden cross (an up-trend signal) and death cross (a down-trend signal). Figure 5 shows a golden cross during 2016 for the S&P 500 index.

```
freduse SP500, clear
drop if SP500==.
drop date
rename daten date
gen obs=_n
tsset obs
tftools movingaverage if year(date)>2010, symbol(SP500) generate(SP500) ///
    period(50) ma_type(sma)
tftools movingaverage if year(date)>2010, symbol(SP500) generate(SP500) ///
    period(200) ma_type(sma)
twoway (line SP500 date) (line SP500_sma_50 date) (line SP500_sma_200 date) ///
    if year(date)>2010
```

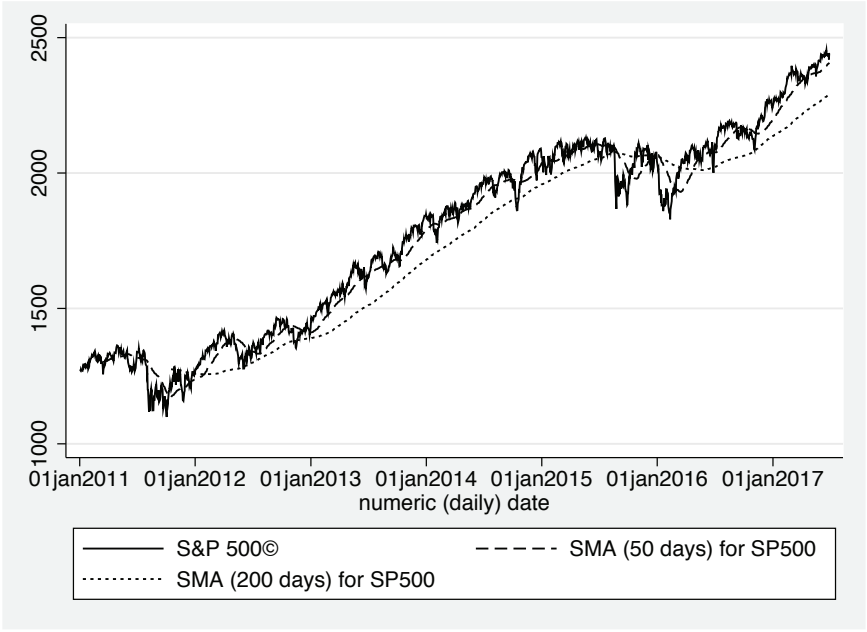

Figure 5. Daily S&P 500 index, 50 days and 200 days simple moving averages

While MACD analysis in figure 3 shows a down trend, the golden cross in figure 5 shows an up trend. These conflicting technical analysis signals are common, because these analysis tools will never provide a consistently reliable signal.

5 RSI: tftools rsi

5.1 Title

`tftools rsi` calculates the RSI for a single time-series variable.

5.2 Syntax

`tftools rsi` [*if*] [*in*], `symbol(`*variable*`) generate(`*newvar*`)`

5.3 Description

The `tftools rsi` command calculates the RSI for a single time-series variable. `tftools rsi` creates a new variable using the *newvar* variable name of the `generate()` option. The data must first be `tsset`. RSI $= 100 - \{100/(1 + \text{RS})\}$, where RS = (average of upward returns over x days)/(average of downward returns over x days). Downward

returns are expressed as positive numbers. Thus the RSI is bounded between 0 and 100. The convention is that an RSI of 70 or above indicates that a stock is overbought and that a downward correction can be expected. An RSI of 30 or below indicates that a stock is oversold and that investors can look forward to an upward correction.

5.4 Options

`symbol(`*variable*`)` specifies the variable that the RSI calculation is based on (usually the stock symbol that contains the daily prices). `symbol()` is required.

`generate(`*newvar*`)` specifies the new variable prefix for the calculated RSI values. `generate()` is required.

5.5 Example

Figure 6 shows the RSI for the daily S&P 500 index. RSI is usually interpreted to provide an up-trend signal below 30 (because it may be oversold and a reversal is in order) and a down-trend signal above 70 (because it may be overbought). Figure 6 shows a period in 2017 where the RSI levels were above 70 and quickly reversed down.

```
freduse SP500, clear
drop if SP500==.
drop date
rename daten date
gen obs=_n
tsset obs
tftools rsi if year(date)>2015, symbol(SP500) generate(SP500)
twoway (line SP500_RSI date) if year(date)>2015
```

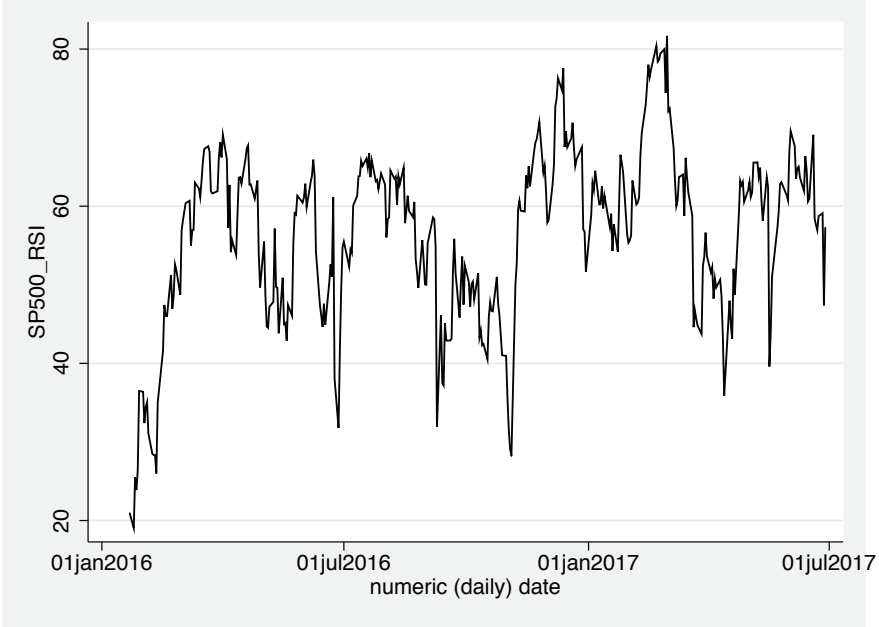

Figure 6. Daily S&P 500 index RSI

6 Conclusion

Financial technical analysis is used by academics and professionals. Our suggested command, `tftools`, provides four subcommands to calculate the most popular tools: moving averages, Bollinger bands, MACD, and the RSI. The moving average is at the core of many technical analysis tools, with analysts calculating the moving averages not only for stock prices but also for the daily minimum, maximum, standard deviation, daily high and low values, etc. Using these moving averages, analysts calculate Bollinger bands to provide support and resistance levels for security prices. While lower and more common standard deviations will provide more immediate support and resistance levels, higher standard deviations will provide more consistent ones. Based on mean reversion, the MACD provides a tool to evaluate how much the shorter-term moving average of a security's price is converging or diverging to its longer-term mean. The relative strength index provides a momentum index that complements the moving average based on

other financial technical analysis tools. While technical analysis in finance may not provide consistently reliable trading signals, it provides guidance that can complement fundamental analysis. Also, while market efficiency theory in finance argues against the long-term success of technical analysis, self-fulfilling prophecies in the markets make them important parts of the academic and professional domains.

7 References

Cahen, P. 2001. *Dynamic Technical Analysis*. Chichester, UK: Wiley.

Chong, T. T.-L., and W.-K. Ng. 2008. Technical analysis and the London stock exchange: Testing the MACD and RSI rules using the FT30. *Applied Economics Letters* 15: 1111–1114.

Kirkpatrick, C. D., II, and J. A. Dahlquist. 2010. *Technical Analysis: The Complete Resource for Financial Market Technicians*. 2nd ed. Upper Saddle River, NJ: FT Press.

Leung, J. M.-J., and T. T.-L. Chong. 2003. An empirical comparison of moving average envelopes and Bollinger Bands. *Applied Economics Letters* 10: 339–341.

Stevens, L. 2002. *Essential Technical Analysis: Tools and Techniques to Spot Market Trends*. New York: Wiley.

Wong, W.-K., M. Manzur, and B.-K. Chew. 2003. How rewarding is technical analysis? Evidence from Singapore stock market. *Applied Financial Economics* 13: 543–551.

About the authors

Mehmet F. Dicle is an associate professor of finance at Loyola University New Orleans.

John D. Levendis is an associate professor of economics at Loyola University New Orleans.