

---

# **technical\_indicators Documentation**

***Release 0.0.14***

**Joao Matos**

June 02, 2014



## CONTENTS

<b>1</b>	<b>Reference</b>	<b>3</b>
1.1	technical_indicators . . . . .	3
<b>2</b>	<b>License</b>	<b>9</b>
<b>3</b>	<b>ChangeLog</b>	<b>15</b>
<b>4</b>	<b>Code authors</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>



This module provides some technical indicators for analysing stocks.

When I can I will add more.

If anyone wishes to contribute with new code or corrections/suggestions, feel free.

**Features:**

Relative Strength Index (RSI), ROC, MA envelopes Simple Moving Average (SMA), Weighted Moving Average (WMA), Exponential Moving Average (EMA) Bollinger Bands (BB), Bollinger Bandwidth, %B

**Dependencies:**

It requires numpy.

This module was done and tested under Windows with Python 2.7.3 and numpy 1.6.1.



## REFERENCE

## 1.1 technical\_indicators

This module provides some technical indicators for analysing stocks.

When I can I will add more. If anyone wishes to contribute with new code or corrections/suggestions, feel free.

Features:

Relative Strength Index (RSI), ROC, MA envelopes Simple Moving Average (SMA), Weighted Moving Average (WMA), Exponential Moving Average (EMA) Bollinger Bands (BB), Bollinger Bandwidth, %B

Dependencies:

It requires numpy. This module was tested under Windows with Python 2.7.3 and numpy 1.6.1.

`technical_indicators.technical_indicators.bb(prices, period, num_std_dev=2.0)`

Bollinger bands (BB) are volatility bands placed above and below a moving average. Volatility is based on the standard deviation, which changes as volatility increases and decreases. The bands automatically widen when volatility increases and narrow when volatility decreases. This dynamic nature of Bollinger Bands also means they can be used on different securities with the standard settings. For signals, Bollinger Bands can be used to identify M-Tops and W-Bottoms or to determine the strength of the trend. Signals derived from narrowing BandWidth are also important.

Bollinger BandWidth is an indicator that derives from Bollinger Bands, and measures the percentage difference between the upper band and the lower band. BandWidth decreases as Bollinger Bands narrow and increases as Bollinger Bands widen. Because Bollinger Bands are based on the standard deviation, falling BandWidth reflects decreasing volatility and rising BandWidth reflects increasing volatility.

%B quantifies a security's price relative to the upper and lower Bollinger Band. There are six basic relationship levels: %B equals 1 when price is at the upper band %B equals 0 when price is at the lower band %B is above 1 when price is above the upper band %B is below 0 when price is below the lower band %B is above .50 when price is above the middle band (20-day SMA) %B is below .50 when price is below the middle band (20-day SMA)

They were developed by John Bollinger. Bollinger suggests increasing the standard deviation multiplier to 2.1 for a 50-period SMA and decreasing the standard deviation multiplier to 1.9 for a 10-period SMA.

[http://www.csidata.com/?page\\_id=797](http://www.csidata.com/?page_id=797) [http://stockcharts.com/school/doku.php?id=chart\\_school:technical\\_indicators:bollinger\\_bands](http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:bollinger_bands)  
[http://stockcharts.com/school/doku.php?id=chart\\_school:technical\\_indicators:bollinger\\_band\\_width](http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:bollinger_band_width)  
[http://stockcharts.com/school/doku.php?id=chart\\_school:technical\\_indicators:bollinger\\_band\\_perce](http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:bollinger_band_perce)

**Input:** prices ndarray period int > 1 and < len(prices) num\_std\_dev float > 0.0 (optional and defaults to 2.0)

**Output:** bbs ndarray with upper, middle, lower bands, bandwidth, range and %B

Test:

```
>>> import numpy as np
>>> import technical_indicators as tai
>>> prices = np.array([86.16, 89.09, 88.78, 90.32, 89.07, 91.15, 89.44,
... 89.18, 86.93, 87.68, 86.96, 89.43, 89.32, 88.72, 87.45, 87.26, 89.50,
... 87.90, 89.13, 90.70, 92.90, 92.98, 91.80, 92.66, 92.68, 92.30, 92.77,
... 92.54, 92.95, 93.20, 91.07, 89.83, 89.74, 90.40, 90.74, 88.02, 88.09,
... 88.84, 90.78, 90.54, 91.39, 90.65])
>>> period = 20
>>> print(tai.bb(prices, period))
[[ 9.12919107e+01  8.87085000e+01  8.61250893e+01  5.82449423e-02
  5.16682146e+00  6.75671306e-03]
 [ 9.19497209e+01  8.90455000e+01  8.61412791e+01  6.52300429e-02
  5.80844179e+00  5.07661263e-01]
 [ 9.26132536e+01  8.92400000e+01  8.58667464e+01  7.55995881e-02
  6.74650724e+00  4.31816571e-01]
 [ 9.29344497e+01  8.93910000e+01  8.58475503e+01  7.92797873e-02
  7.08689946e+00  6.31086945e-01]
 [ 9.33114122e+01  8.95080000e+01  8.57045878e+01  8.49848539e-02
  7.60682430e+00  4.42420124e-01]
 [ 9.37270110e+01  8.96885000e+01  8.56499890e+01  9.00563838e-02
  8.07702198e+00  6.80945403e-01]
 [ 9.38972812e+01  8.97460000e+01  8.55947188e+01  9.25117832e-02
  8.30256250e+00  4.63143909e-01]
 [ 9.42636418e+01  8.99125000e+01  8.55613582e+01  9.67861377e-02
  8.70228361e+00  4.15826692e-01]
 [ 9.45630193e+01  9.00805000e+01  8.55979807e+01  9.95225220e-02
  8.96503854e+00  1.48579313e-01]
 [ 9.47851634e+01  9.03815000e+01  8.59778366e+01  9.74461225e-02
  8.80732672e+00  1.93266744e-01]
 [ 9.50411874e+01  9.06575000e+01  8.62738126e+01  9.67087637e-02
  8.76737475e+00  7.82660026e-02]
 [ 9.49062071e+01  9.08630000e+01  8.68197929e+01  8.89956780e-02
  8.08641429e+00  3.22789193e-01]
 [ 9.49015375e+01  9.08830000e+01  8.68644625e+01  8.84332063e-02
  8.03707509e+00  3.05526266e-01]
 [ 9.48939343e+01  9.09040000e+01  8.69140657e+01  8.77834713e-02
  7.97986867e+00  2.26311285e-01]
 [ 9.48594576e+01  9.09880000e+01  8.71165424e+01  8.50982021e-02
  7.74291521e+00  4.30661576e-02]
 [ 9.46722663e+01  9.11525000e+01  8.76327337e+01  7.72280810e-02
  7.03953265e+00 -5.29486389e-02]
 [ 9.45543042e+01  9.11905000e+01  8.78266958e+01  7.37753219e-02
  6.72760849e+00  2.48722001e-01]
 [ 9.46761721e+01  9.11200000e+01  8.75638279e+01  7.80546993e-02
  7.11234420e+00  4.72660054e-02]
 [ 9.45733946e+01  9.11670000e+01  8.77606054e+01  7.47286754e-02
  6.81278915e+00  2.01003516e-01]
 [ 9.45322396e+01  9.12495000e+01  8.79667604e+01  7.19508503e-02
  6.56547911e+00  4.16304661e-01]
 [ 9.45303313e+01  9.12415000e+01  8.79526687e+01  7.20906879e-02
  6.57766250e+00  7.52141243e-01]
 [ 9.43672335e+01  9.11660000e+01  8.79647665e+01  7.02286710e-02
  6.40246702e+00  7.83328285e-01]
 [ 9.41460689e+01  9.10495000e+01  8.79529311e+01  6.80194599e-02
  6.19313782e+00  6.21182512e-01]]
```

`technical_indicators.technical_indicators.ema (prices, period, ema_type=0)`

Exponential Moving Average (EMA) are used to smooth the data in an array to help eliminate noise and identify



trends. Exponential moving averages reduce the lag by applying more weight to recent prices. The weighting applied to the most recent price depends on the number of periods in the moving average.

They do not predict price direction, but can be used to identify the direction of the trend or define potential support and resistance levels.

EMA type 0  $EMA_n = w.P_n + (1 - w).EMA_{n-1}$   $EMA_n = EMA_{n-1} + w.(P_n - EMA_{n-1})$   $EMA_n = w.P_n + w.(1 - w).P_{n-1} + w.(1 - w)^2.P_{n-2} + \dots + w.(1 - w)^{(n-1)}.P_1 + w.(1 - w)^n.EMA_0$  where  $w = 2 / (n + 1)$  and  $EMA_0 = \text{mean}(\text{oldest period})$  or  $EMA_n = w.EMA_{n-1} + (1 - w).P_n$  where  $w = 1 - 2 / (n + 1)$  and  $P_n$  is the most recent price and  $EMA_0 = \text{mean}(\text{oldest period})$

EMA type 1 The above formulas with  $EMA_0 = P_1$  (oldest price)

EMA type 2  $EMA = (P_n + w.P_{n-1} + w^2.P_{n-2} + w^3.P_{n-3} + \dots) / K$  where  $K = 1 + w + w^2 + \dots = 1 / (1 - w)$  and  $P_n$  is the most recent price and  $w = 2 / (N + 1)$

<http://www.financialwebring.org/gummy-stuff/MA-stuff.htm>

[http://www.csidata.com/?page\\_id=797](http://www.csidata.com/?page_id=797) [http://stockcharts.com/school/doku.php?st=moving+average&id=chart\\_school:technical\\_i](http://stockcharts.com/school/doku.php?st=moving+average&id=chart_school:technical_i)

**Input:** prices ndarray period int > 1 and < len(prices) ema\_type can be 0, 1 or 2

**Output:** emas ndarray

Tests:

```
>>> import numpy as np
>>> import technical_indicators as tai
>>> prices = np.array([22.27, 22.19, 22.08, 22.17, 22.18, 22.13, 22.23,
... 22.43, 22.24, 22.29, 22.15, 22.39, 22.38, 22.61, 23.36, 24.05, 23.75,
... 23.83, 23.95, 23.63, 23.82, 23.87, 23.65, 23.19, 23.10, 23.33, 22.68,
... 23.10, 22.40, 22.17])
>>> period = 10
>>> print(tai.ema(prices, period))
[ 22.221      22.20809091  22.24116529  22.26640796  22.32887924
  22.51635574  22.79520015  22.96880013  23.12538192  23.27531248
  23.33980112  23.42711001  23.50763546  23.53351992  23.47106176
  23.40359598  23.39021489  23.26108491  23.23179675  23.08056097
  22.91500443]
>>> print(tai.ema(prices, period, 1))
[ 22.27      22.25545455  22.22355372  22.21381668  22.20766819
  22.1935467  22.20017457  22.24196102  22.24160447  22.25040366
  22.23214845  22.26084873  22.2825126  22.34205576  22.52713653
  22.8040208  22.97601702  23.13128665  23.28014362  23.34375387
  23.43034408  23.51028152  23.53568488  23.47283308  23.40504525
  23.39140066  23.26205508  23.23259052  23.08121043  22.9155358 ]
>>> print(tai.ema(prices, period, 2))
[ 22.28588695  22.174706  22.35085492  22.37470018  22.5672175
  23.21585701  23.89833692  23.77696963  23.82035739  23.9264279
  23.68389526  23.79525297  23.85640891  23.68752817  23.28045894
  23.13280996  23.29414649  22.79166223  23.04393782  22.51707883
  22.23310448]
```

`technical_indicators.technical_indicators.ma_env(prices, period, percent, ma_type=0)`

Moving Average Envelopes are percentage-based envelopes set above and below a moving average. They can be used as a trend following indicator. The envelopes can also be used to identify overbought and oversold levels when the trend is relatively flat.

Upper Envelope:  $MA + (MA \times \text{percent})$  Lower Envelope:  $MA - (MA \times \text{percent})$

[http://www.csidata.com/?page\\_id=797](http://www.csidata.com/?page_id=797)

[http://stockcharts.com/school/doku.php?id=chart\\_school:technical\\_indicators:moving\\_average\\_envel](http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:moving_average_envel)

[http://stockcharts.com/school/doku.php?id=chart\\_school:technical\\_indicators:bollinger\\_band\\_perce](http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:bollinger_band_perce)

**Input:** prices ndarray period int > 1 and < len(prices) percent float > 0.00 and < 1.00 ma\_type 0=EMA type 0, 1=EMA type 1, 2=EMA type 2, 3=WMA, 4=SMA

**Output:** ma\_envs ndarray with upper, middle, lower bands, range and %B

Test:

```
>>> import numpy as np
>>> import technical_indicators as tai
>>> prices = np.array([86.16, 89.09, 88.78, 90.32, 89.07, 91.15, 89.44,
... 89.18, 86.93, 87.68, 86.96, 89.43, 89.32, 88.72, 87.45, 87.26, 89.50,
... 87.90, 89.13, 90.70, 92.90, 92.98, 91.80, 92.66, 92.68, 92.30, 92.77,
... 92.54, 92.95, 93.20, 91.07, 89.83, 89.74, 90.40, 90.74, 88.02, 88.09,
... 88.84, 90.78, 90.54, 91.39, 90.65])
>>> period = 20
>>> print(tai.ma_env(prices, period, 0.1, 4))
[[ 97.57935  88.7085  79.83765  17.7417  0.35635537]
 [ 97.95005  89.0455  80.14095  17.8091  0.50249872]
 [ 98.164    89.24    80.316    17.848   0.4742268 ]
 [ 98.3301   89.391   80.4519   17.8782  0.55196273]
 [ 98.4588   89.508   80.5572   17.9016  0.47553291]
 [ 98.65735  89.6885  80.71965  17.9377  0.58147644]
 [ 98.7206   89.746   80.7714   17.9492  0.48295189]
 [ 98.90375  89.9125  80.92125  17.9825  0.45926595]
 [ 99.08855  90.0805  81.07245  18.0161  0.32512863]
 [ 99.41965  90.3815  81.34335  18.0763  0.35055017]
 [ 99.72325  90.6575  81.59175  18.1315  0.29607313]
 [ 99.9493   90.863   81.7767   18.1726  0.42114502]
 [ 99.9713   90.883   81.7947   18.1766  0.41401032]
 [ 99.9944   90.904   81.8136   18.1808  0.37987327]
 [100.0868   90.988   81.8892   18.1976  0.30557876]
 [100.26775  91.1525  82.03725  18.2305  0.28648419]
 [100.30955  91.1905  82.07145  18.2381  0.40730942]
 [100.232    91.12    82.008    18.224   0.32330992]
 [100.2837   91.167   82.0503   18.2334  0.38828194]
 [100.37445  91.2495  82.12455  18.2499  0.46989025]
 [100.36565  91.2415  82.11735  18.2483  0.59088518]
 [100.2826   91.166   82.0494   18.2332  0.59948884]
 [100.15445  91.0495  81.94455  18.2099  0.54121385]]
```

`technical_indicators.technical_indicators.roc(prices, period=21)`

The Rate-of-Change (ROC) indicator, a.k.a. Momentum, is a pure momentum oscillator that measures the percent change in price from one period to the next. The plot forms an oscillator that fluctuates above and below the zero line as the Rate-of-Change moves from positive to negative. ROC signals include centerline crossovers, divergences and overbought-oversold readings. Identifying overbought or oversold extremes comes natural to the Rate-of-Change oscillator. It can be used to measure the ROC of any data series, such as price or another indicator. Also known as PROC when used with price.

$$ROC = [(Close - Close\ n\ periods\ ago) / (Close\ n\ periods\ ago)] * 100$$

[http://www.csidata.com/?page\\_id=797](http://www.csidata.com/?page_id=797) [http://stockcharts.com/school/doku.php?id=chart\\_school:technical\\_indicators:rate\\_of\\_change](http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:rate_of_change)

**Input:** prices ndarray period int > 1 and < len(prices) (optional and defaults to 21)

**Output:** rocs ndarray

Test:

```
>>> import numpy as np
>>> import technical_indicators as tai
>>> prices = np.array([11045.27, 11167.32, 11008.61, 11151.83, 10926.77,
... 10868.12, 10520.32, 10380.43, 10785.14, 10748.26, 10896.91, 10782.95,
... 10620.16, 10625.83, 10510.95, 10444.37, 10068.01, 10193.39, 10066.57,
... 10043.75])
>>> print(tai.roc(prices, period=12))
[-3.84879682 -4.84888048 -4.52064339 -6.34389154 -7.85923013 -6.20834146
-4.31308173 -3.24341092]
```

technical\_indicators.technical\_indicators.**rsi**(prices, period=14)

The Relative Strength Index (RSI) is a momentum oscillator. It oscillates between 0 and 100. It is considered overbought/oversold when it's over 70/below 30. Some traders use 80/20 to be on the safe side. RSI becomes more accurate as the calculation period (min\_periods) increases. This can be lowered to increase sensitivity or raised to decrease sensitivity. 10-day RSI is more likely to reach overbought or oversold levels than 20-day RSI. The look-back parameters also depend on a security's volatility.

Like many momentum oscillators, overbought and oversold readings for RSI work best when prices move sideways within a range.

You can also look for divergence with price. If the price has new highs/lows, and the RSI hasn't, expect a reversal. Signals can also be generated by looking for failure swings and centerline crossovers.

RSI can also be used to identify the general trend.

The RSI was developed by J. Welles Wilder and was first introduced in his article in the June, 1978 issue of Commodities magazine, now known as Futures magazine. It is detailed in his book New Concepts In Technical Trading Systems.

[http://www.csidata.com/?page\\_id=797](http://www.csidata.com/?page_id=797) [http://stockcharts.com/help/doku.php?id=chart\\_school:technical\\_indicators:relative\\_strength\\_index](http://stockcharts.com/help/doku.php?id=chart_school:technical_indicators:relative_strength_index)

**Input:** prices ndarray period int > 1 and < len(prices) (optional and defaults to 14)

**Output:** rsi ndarray

Test:

```
>>> import numpy as np
>>> import technical_indicators as tai
>>> prices = np.array([44.55, 44.3, 44.36, 43.82, 44.46, 44.96, 45.23,
... 45.56, 45.98, 46.22, 46.03, 46.17, 45.75, 46.42, 46.42, 46.14, 46.17,
... 46.55, 46.36, 45.78, 46.35, 46.39, 45.85, 46.59, 45.92, 45.49, 44.16,
... 44.31, 44.35, 44.7, 43.55, 42.79, 43.26])
>>> print(tai.rsi(prices))
[ 70.02141328  65.77440817  66.01226849  68.95536568  65.88342192
  57.46707948  62.532685    62.86690858  55.64975092  62.07502976
  54.39159393  50.10513101  39.68712141  41.17273382  41.5859395
  45.21224077  37.06939108  32.85768734  37.58081218]
```

technical\_indicators.technical\_indicators.**sma**(prices, period)

Simple Moving Average (SMA) are used to smooth the data in an array to help eliminate noise and identify trends. In SMA, each value in the time period carries equal weight.

They do not predict price direction, but can be used to identify the direction of the trend or define potential support and resistance levels.

$SMA = (P1 + P2 + \dots + Pn) / K$  where  $K = n$  and  $Pn$  is the most recent price

<http://www.financialwebring.org/gummy-stuff/MA-stuff.htm>

[http://www.csidata.com/?page\\_id=797](http://www.csidata.com/?page_id=797) [http://stockcharts.com/school/doku.php?id=chart\\_school:technical\\_indicators:simple\\_moving\\_average](http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:simple_moving_average)

**Input:** prices ndarray period int > 1 and < len(prices)

**Output:** smas ndarray

Test:

```
>>> import numpy as np
>>> import technical_indicators as tai
>>> prices = np.array([22.27, 22.19, 22.08, 22.17, 22.18, 22.13, 22.23,
... 22.43, 22.24, 22.29, 22.15, 22.39, 22.38, 22.61, 23.36, 24.05, 23.75,
... 23.83, 23.95, 23.63, 23.82, 23.87, 23.65, 23.19, 23.10, 23.33, 22.68,
... 23.10, 22.40, 22.17])
>>> period = 10
>>> print(tai.sma(prices, period))
[ 22.221  22.209  22.229  22.259  22.303  22.421  22.613  22.765  22.905
  23.076  23.21   23.377  23.525  23.652  23.71   23.684  23.612  23.505
  23.432  23.277  23.131]
```

technical\_indicators.technical\_indicators.**wma**(prices, period)

Weighted Moving Average (WMA) is a type of moving average that assigns a higher weighting to recent price data.

$WMA = (P_1 + 2 P_2 + 3 P_3 + \dots + n P_n) / K$  where  $K = (1+2+\dots+n) = n(n+1)/2$  and  $P_n$  is the most recent price after the 1st WMA we can use another formula  $WMA_n = WMA_{n-1} + w.(P_n - SMA(prices, n-1))$  where  $w = 2 / (n + 1)$

[http://www.csidata.com/?page\\_id=797](http://www.csidata.com/?page_id=797)

<http://www.financialwebring.org/gummy-stuff/MA-stuff.htm>

<http://www.investopedia.com/terms/l/linearlyweightedmovingaverage.asp>

<http://fxtrade.oanda.com/learn/forex-indicators/weighted-moving-average>

**Input:** prices ndarray period int > 1 and < len(prices)

**Output:** wmas ndarray

Test:

```
>>> import numpy as np
>>> import technical_indicators as tai
>>> prices = np.array([77, 79, 79, 81, 83, 49, 55])
>>> period = 5
>>> print(tai.wma(prices, period))
[ 80.73333333  70.46666667  64.06666667]
```

## LICENSE

technical\_indicators - blablabla  
Copyright (C) 2014 Joao Carlos Roseta Matos

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

### GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc., <<http://fsf.org/>>  
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it

in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty;

and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or



otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

## CHANGELOG

### 0.0.14 2014-06-02

Added end user documentation to .gitignore.  
Added option PROJ\_TYPE to build.bat to distinguish between modules and applications.  
Added pythonhosted.org files to MANIFEST.in.  
Changed \_\_init\_\_.py to use glob to select py2exe and cxf data files.  
Added options to py2exe config in setup.py.  
Fill some Docstrings.

### 0.0.13 2014-05-31

Remarked bdist\_egg, bdist\_wininst, cxf and py2exe builds from build.bat.

### 0.0.12 2014-05-31

Added zip\_safe to setup.py.

### 0.0.11 2014-05-31

Added PyPI documentation in dir pythonhosted.org (redirects to ReadTheDocs).  
Changed doc\index.rst to include README.rst.  
Updated build.bat.

### 0.0.10 2014-05-31

Corrected classifiers in \_\_init\_\_.py. Added ReadTheDocs doc.  
Added prep\_rst2pdf.py and prep\_rst2pdf.py to help build.bat.  
Changed build.bat.

### 0.0.9 2014-05-30

Added py\_app\_ver.py and changed build.bat.

### 0.0.8 2014-05-30

Corrected yml and \_\_init\_\_.py because numpy is not installing in Py3

### 0.0.7 2014-05-30

Corrected test and yml files

### 0.0.6 2014-05-29

Added Shippable CI

**0.0.5 2014-05-29**

Added doctests, packaging, build automation, sphinx doc, travis.  
Changed license and versioning.

**0.0.4 2013-07-03**

Added ROC and MA envelopes

**0.0.3 2013-06-30**

Added WMA and more EMA types.

**0.0.2 2013-06-18**

Added Bollinger bandwidth and %B  
Created a GitHub repository

**0.0.1 2013-06-05**

Includes RSI, SMA, EMA and BB

## CODE AUTHORS

Joao Matos <jcrmatos@gmail.com>



**t**

`technical_indicators.technical_indicators,`  
3