jhTAlib

Joost Hoeks

2019-03-11

Contents

jhTAlib

Technical Analysis Library Time-Series

You can use and import it for your:

- Technical Analysis Software
- Charting Software
- Backtest Software
- Trading Robot Software
- Trading Software in general

Work in progress…

Depends only on

- The Python Standard Library

Install

From PyPI:

```
$ [sudo] pip3 install jhtalib
```

From source:

```
$ git clone https://github.com/joosthoeks/jhTAlib.git
$ cd jhTAlib
$ [sudo] pip3 install -e .
```

Update

From PyPI:

```
$ [sudo] pip3 install --upgrade jhtalib
```

From source:

```
$ cd jhTAlib
$ git pull [upstream master]
```

Examples

```
$ cd example/
```

Example 1

```
$ python3 example-1-plot.py
```

or

Open In Colab

Example 2

```
$ python3 example-2-plot.py
```

or

Open In Colab

Example 3

```
$ python3 example-3-plot.py
```

or

Open In Colab

Example 4

```
$ python3 example-4-plot-quandl.py
```

or

Open In Colab

Example 5

```
$ python3 example-5-plot-quandl.py
```

or

Open In Colab

Example 6

```
$ python3 example-6-plot-quandl.py
```

or

Open In Colab

Example 7

```
$ python3 example-7-quandl-2-df.py
```

or

Open In Colab

Example 8

```
$ python3 example-8-alphavantage-2-df.py
```

or

Open In Colab

Example 9

```
$ python3 example-9-cryptocompare-2-df.py
```

or

Open In Colab

Example 10

DF NumPy Pandas

Open In Colab

Test

```
$ cd test/
$ python3 test.py
```

Reference

```
import jhtalib as jhta
```

Behavioral Techniques

All Time High

- `dict of lists = jhta.ATH(df, price='High')`

Last Major Correction

- `dict of lists = jhta.LMC(df, price='Low')`

Pivot Point

- `dict of lists = jhta.PP(df)`

Fibonacci Price Retracements

- `dict of lists = jhta.FIBOPR(df, price='Close')`

Fibonacci Time Retracements

W. D. Gann Price Retracements

- `dict of lists = jhta.GANNPR(df, price='Close')`

W. D. Gann Time Retracements

Julian Day Number

- `jdn = jhta.JDN(utc_year, utc_month, utc_day)`

Julian Date

- `jd = jhta.JD(utc_year, utc_month, utc_day, utc_hour, utc_minute, utc_second)`

SUNC | Sun Cycle

MERCURYC | Mercury Cycle

VENUSC | Venus Cycle

EARTHC | Earth Cycle

MARSC | Mars Cycle

JUPITERC | Jupiter Cycle

SATURNC | Saturn Cycle

URANUSC | Uranus Cycle

NEPTUNEC | Neptune Cycle

PLUTOC | Pluto Cycle

MOONC | Moon Cycle

Cycle Indicators

HT_DCPERIOD | Hilbert Transform - Dominant Cycle Period

HT_DCPHASE | Hilbert Transform - Dominant Cycle Phase

HT_PHASOR | Hilbert Transform - Phasor Components

HT_SINE | Hilbert Transform - SineWave

HT_TRENDLINE | Hilbert Transform - Instantaneous Trendline

HT_TRENDMODE | Hilbert Transform - Trend vs Cycle Mode

Trend Score

- `list = jhta.TS(df, n, price='Close')`

Data

CSV file 2 DataFeed

- `dict of tuples = jhta.CSV2DF(csv_file_path)`

CSV file url 2 DataFeed

- `dict of tuples = jhta.CSVURL2DF(csv_file_url)`

DataFeed 2 CSV file

- `csv file = jhta.DF2CSV(df, csv_file_path)`

DataFeed 2 DataFeed Reversed

- `dict of tuples = jhta.DF2DFREV(df)`

DataFeed 2 DataFeed Window

- `dict of tuples = jhta.DF2DFWIN(df, start=0, end=10)`

DataFeed HEAD

- `dict of tuples = jhta.DF_HEAD(df, n=5)`

DataFeed TAIL

- `dict of tuples = jhta.DF_TAIL(df, n=5)`

DataFeed 2 Heikin-Ashi DataFeed

- `dict of tuples = jhta.DF2HEIKIN_ASHI(df)`

Event Driven

Accumulation Swing Index (J. Welles Wilder)

- `list = jhta.ASI(df, L)`

Swing Index (J. Welles Wilder)

- `list = jhta.SI(df, L)`

Experimental

Swing Average Price - previous Average Price

- `list = jhta.JH_SAVGP(df)`

Swing Average Price - previous Average Price Summation

- `list = jhta.JH_SAVGPS(df)`

Swing Close - Open

- `list = jhta.JH_SCO(df)`

Swing Close - Open Summation

- `list = jhta.JH_SCOS(df)`

Swing Median Price - previous Median Price

- `list = jhta.JH_SMEDP(df)`

Swing Median Price - previous Median Price Summation

- `list = jhta.JH_SMEDPS(df)`

Swing Price - previous Price

- `list = jhta.JH_SPP(df, price='Close')`

Swing Price - previous Price Summation

- `list = jhta.JH_SPPS(df, price='Close')`

Swing Typical Price - previous Typical Price

- `list = jhta.JH_STYPP(df)`

Swing Typical Price - previous Typical Price Summation

- `list = jhta.JH_STYPPS(df)`

Swing Weighted Close Price - previous Weighted Close Price

- `list = jhta.JH_SWCLP(df)`

Swing Weighted Close Price - previous Weighted Close Price Summation

- `list = jhta.JH_SWCLPS(df)`

General

Normalize

- `list = jhta.NORMALIZE(df, price_max='High', price_min='Low', price='Close')`

Standardize

- `list = jhta.STANDARDIZE(df, price='Close')`

Spread

- `list = jhta.SPREAD(df1, df2, price1='Close', price2='Close')`

Comparative Performance

- `list = jhta.CP(df1, df2, price1='Close', price2='Close')`

Comparative Relative Strength Index

- `list = jhta.CRSI(df1, df2, n, price1='Close', price2='Close')`

Comparative Strength

- `list = jhta.CS(df1, df2, price1='Close', price2='Close')`

Hit Rate / Win Rate

- `float = jhta.HR(hit_trades_int, total_trades_int)`

Profit/Loss Ratio

- `float = jhta.PLR(mean_trade_profit_float, mean_trade_loss_float)`

Expected Value

- `float = jhta.EV(hitrade_float, mean_trade_profit_float, mean_trade_loss_float)`

Probability of Ruin (Table of Lucas and LeBeau)

- `int = jhta.POR(hitrade_float, profit_loss_ratio_float)`

Information

Print df Information

- `print = jhta.INFO(df, price='Close')`

Print Trades Information

- `print = jhta.INFO_TRADES(profit_trades_list, loss_trades_list)`

Math Functions

Exponential

- `list = jhta.EXP(df, price='Close')`

Logarithm

- `list = jhta.LOG(df, price='Close')`

Base-10 Logarithm

- `list = jhta.LOG10(df, price='Close')`

Square Root

- `list = jhta.SQRT(df, price='Close')`

Arc Cosine

- `list = jhta.ACOS(df, price='Close')`

Arc Sine

- `list = jhta.ASIN(df, price='Close')`

Arc Tangent

- `list = jhta.ATAN(df, price='Close')`

Cosine

- `list = jhta.COS(df, price='Close')`

Sine

- `list = jhta.SIN(df, price='Close')`

Tangent

- `list = jhta.TAN(df, price='Close')`

Inverse Hyperbolic Cosine

- `list = jhta.ACOSH(df, price='Close')`

Inverse Hyperbolic Sine

- `list = jhta.ASINH(df, price='Close')`

Inverse Hyperbolic Tangent

- `list = jhta.ATANH(df, price='Close')`

Hyperbolic Cosine

- `list = jhta.COSH(df, price='Close')`

Hyperbolic Sine

- `list = jhta.SINH(df, price='Close')`

Hyperbolic Tangent

- `list = jhta.TANH(df, price='Close')`

Mathematical constant PI

- `float = jhta.PI()`

Mathematical constant E

- `float = jhta.E()`

Mathematical constant TAU

- `float = jhta.TAU()`

Mathematical constant PHI

- `float = jhta.PHI()`

Ceiling

- `list = jhta.CEIL(df, price='Close')`

Floor

- `list = jhta.FLOOR(df, price='Close')`

Radians to Degrees

- `list = jhta.DEGREES(df, price='Close')`

Degrees to Radians

- `list = jhta.RADIANS(df, price='Close')`

Addition High + Low

- `list = jhta.ADD(df)`

Division High / Low

- `list = jhta.DIV(df)`

Highest value over a specified period

- `list = jhta.MAX(df, n, price='Close')`

MAXINDEX | Index of highest value over a specified period

Lowest value over a specified period

- `list = jhta.MIN(df, n, price='Close')`

MININDEX | Index of lowest value over a specified period

MINMAX | Lowest and Highest values over a specified period

MINMAXINDEX | Indexes of lowest and highest values over a specified period

Multiply High * Low

- `list = jhta.MULT(df)`

Subtraction High - Low

- `list = jhta.SUB(df)`

Summation

- `list = jhta.SUM(df, n, price='Close')`

Momentum Indicators

ADX | Average Directional Movement Index

ADXR | Average Directional Movement Index Rating

Absolute Price Oscillator

- `list = jhta.APO(df, n_fast, n_slow, price='Close')`

AROON | Aroon

AROONOSC | Aroon Oscillator

BOP | Balance Of Power

CCI | Commodity Channel Index

CMO | Chande Momentum Oscillator

DX | Directional Movement Index

Intraday Momentum Index

- `list = jhta.IMI(df)`

MACD | Moving Average Convergence/Divergence

MACDEXT | MACD with controllable MA type

MACDFIX | Moving Average Convergence/Divergence Fix 12/26

MFI | Money Flow Index

MINUS_DI | Minus Directional Indicator

MINUS_DM | Minus Directional Movement

Momentum

- `list = jhta.MOM(df, n, price='Close')`

PLUS_DI | Plus Directional Indicator

PLUS_DM | Plus Directional Movement

PPO | Percentage Price Oscillator

Rate of Change

- `list = jhta.ROC(df, n, price='Close')`

Rate of Change Percentage

- `list = jhta.ROCP(df, n, price='Close')`

Rate of Change Ratio

- `list = jhta.ROCR(df, n, price='Close')`

Rate of Change Ratio 100 scale

- `list = jhta.ROCR100(df, n, price='Close')`

Relative Strength Index

- `list = jhta.RSI(df, n, price='Close')`

STOCH | Stochastic

STOCHF | Stochastic Fast

STOCHRSI | Stochastic Relative Strength Index

TRIX | 1-day Rate-Of-Change (ROC) of a Triple Smooth EMA

ULTOSC | Ultimate Oscillator

Williams' %R

- `list = jhta.WILLR(df, n)`

Overlap Studies

Bollinger Bands

- `dict of lists = jhta.BBANDS(df, n, f=2)`

Bollinger Band Width

- `list = jhta.BBANDW(df, n, f=2)`

DEMA | Double Exponential Moving Average

EMA | Exponential Moving Average

Envelope Percent

- `dict of lists = jhta.ENVP(df, pct=.01, price='Close')`

KAMA | Kaufman Adaptive Moving Average

MA | Moving Average

MAMA | MESA Adaptive Moving Average

MAVP | Moving Average with Variable Period

MidPoint over period

- `list = jhta.MIDPOINT(df, n, price='Close')`

MidPoint Price over period

- `list = jhta.MIDPRICE(df, n)`

Mayer Multiple Ratio

- `list = jhta.MMR(df, n=200, price='Close')`

Parabolic SAR

- `list = jhta.SAR(df, af_step=.02, af_max=.2)`

SAREXT | Parabolic SAR - Extended

Simple Moving Average

- `list = jhta.SMA(df, n, price='Close')`

T3 | Triple Exponential Moving Average (T3)

TEMA | Triple Exponential Moving Average

Triangular Moving Average

- `list = jhta.TRIMA(df, n, price='Close')`

WMA | Weighted Moving Average

Pattern Recognition

CDL2CROWS | Two Crows |

CDL3BLACKCROWS | Three Black Crows |

CDL3INSIDE | Three Inside Up/Down |

CDL3LINESTRIKE | Three-Line Strike |

CDL3OUTSIDE | Three Outside Up/Down |

CDL3STARSINSOUTH | Three Stars In The South |

CDL3WHITESOLDIERS | Three Advancing White Soldiers |

CDLABANDONEDBABY | Abandoned Baby |

CDLADVANCEBLOCK | Advance Block |

CDLBELTHOLD | Belt-hold |

CDLBREAKAWAY | Breakaway |

CDLCLOSINGMARUBOZU | Closing Marubozu |

CDLCONSEALBABYSWALL | Concealing Baby Swallow |

CDLCOUNTERATTACK | Counterattack |

CDLDARKCLOUDCOVER | Dark Cloud Cover |

CDLDOJI | Doji |

CDLDOJISTAR | Doji Star |

CDLDRAGONFLYDOJI | Dragonfly Doji |

CDLENGULFING | Engulfing Pattern |

CDLEVENINGDOJISTAR | Evening Doji Star |

CDLEVENINGSTAR | Evening Star |

CDLGAPSIDESIDEWHITE | Up/Down-gap side-by-side white lines |

CDLGRAVESTONEDOJI | Gravestone Doji |

CDLHAMMER | Hammer |

CDLHANGINGMAN | Hanging Man |

CDLHARAMI | Harami Pattern |

CDLHARAMICROSS | Harami Cross Pattern |

CDLHIGHWAVE | High-Wave Candle |

CDLHIKKAKE | Hikkake Pattern |

CDLHIKKAKEMOD | Modified Hikkake Pattern |

CDLHOMINGPIGEON | Homing Pigeon |

CDLIDENTICAL3CROWS | Identical Three Crows |

CDLINNECK | In-Neck Pattern |

CDLINVERTEDHAMMER | Inverted Hammer |

CDLKICKING | Kicking |

CDLKICKINGBYLENGTH | Kicking - bull/bear determined by the longer marubozu |

CDLLADDERBOTTOM | Ladder Bottom |

CDLLONGLEGGEDDOJI | Long Legged Doji |

CDLLONGLINE | Long Line Candle |

CDLMARUBOZU | Marubozu |

CDLMATCHINGLOW | Matching Low |

CDLMATHOLD | Mat Hold |

CDLMORNINGDOJISTAR | Morning Doji Star |

CDLMORNINGSTAR | Morning Star |

CDLONNECK | On-Neck Pattern |

CDLPIERCING | Piercing Pattern |

CDLRICKSHAWMAN | Rickshaw Man |

CDLRISEFALL3METHODS | Rising/Falling Three Methods |

CDLSEPARATINGLINES | Separating Lines |

CDLSHOOTINGSTAR | Shooting Star |

CDLSHORTLINE | Short Line Candle |

CDLSPINNINGTOP | Spinning Top |

CDLSTALLEDPATTERN | Stalled Pattern |

CDLSTICKSANDWICH | Stick Sandwich |

CDLTAKURI | Takuri (Dragonfly Doji with very long lower shadow) |

CDLTASUKIGAP | Tasuki Gap |

CDLTHRUSTING | Thrusting Pattern |

CDLTRISTAR | Tristar Pattern |

CDLUNIQUE3RIVER | Unique 3 River |

CDLUPSIDEGAP2CROWS | Upside Gap Two Crows |

CDLXSIDEGAP3METHODS | Upside/Downside Gap Three Methods |

Price Transform

AVGPRICE | Average Price | DONE

- `list = jhta.AVGPRICE(df)`

MEDPRICE | Median Price | DONE

- `list = jhta.MEDPRICE(df)`

TYPPRICE | Typical Price | DONE

- `list = jhta.TYPPRICE(df)`

WCLPRICE | Weighted Close Price | DONE

- `list = jhta.WCLPRICE(df)`

Statistic Functions

MEAN | Arithmetic mean (average) of data | DONE

- `list = jhta.MEAN(df, n, price='Close')`

HARMONIC_MEAN | Harmonic mean of data | DONE

- `list = jhta.HARMONIC_MEAN(df, n, price='Close')`

MEDIAN | Median (middle value) of data | DONE

- `list = jhta.MEDIAN(df, n, price='Close')`

MEDIAN_LOW | Low median of data | DONE

- `list = jhta.MEDIAN_LOW(df, n, price='Close')`

MEDIAN_HIGH | High median of data | DONE

- `list = jhta.MEDIAN_HIGH(df, n, price='Close')`

MEDIAN_GROUPED | Median, or 50th percentile, of grouped data | DONE

- `list = jhta.MEDIAN_GROUPED(df, n, price='Close', interval=1)`

MODE | Mode (most common value) of discrete data | DONE

- `list = jhta.MODE(df, n, price='Close')`

PSTDEV | Population standard deviation of data | DONE

- `list = jhta.PSTDEV(df, n, price='Close', mu=None)`

PVARIANCE | Population variance of data | DONE

- `list = jhta.PVARIANCE(df, n, price='Close', mu=None)`

STDEV | Sample standard deviation of data | DONE

- `list = jhta.STDEV(df, n, price='Close', xbar=None)`

VARIANCE | Sample variance of data | DONE

- `list = jhta.VARIANCE(df, n, price='Close', xbar=None)`

COV | Covariance | DONE

- `float = jhta.COV(list1, list2)`

COVARIANCE | Covariance | DONE

- `list = jhta.COVARIANCE(df1, df2, n, price1='Close', price2='Close')`

BETA | Beta | DONE

- `list = jhta.BETA(df1, df2, n, price1='Close', price2='Close')`

LSR | Least Squares Regression | DONE

- `list = jhta.LSR(df, price='Close', predictions_int=0)`

SLR | Simple Linear Regression | DONE

- `list = jhta.SLR(df, price='Close', predictions_int=0)`

Volatility Indicators

ATR | Average True Range | DONE

- `list = jhta.ATR(df, n)`

NATR | Normalized Average True Range |

TRANGE | True Range | DONE

- `list = jhta.TRANGE(df)`

Volume Indicators

AD | Chaikin A/D Line | DONE

- `list = jhta.AD(df)`

ADOSC | Chaikin A/D Oscillator |

OBV | On Balance Volume | DONE

- `list = jhta.OBV(df)`