# jhTAlib

Joost Hoeks

2019-03-08

## 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
```

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| **behavioral_techniques** | | |
| ATH | All Time High | DONE |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| dict of lists | `jhta.ATH(df, price='High')` | |
| LMC | Last Major Correction | DONE |
| dict of lists | `jhta.LMC(df, price='Low')` | |
| PP | Pivot Point | DONE |
| dict of lists | `jhta.PP(df)` | |
| FIBOPR | Fibonacci Price Retracements | DONE |
| dict of lists | `jhta.FIBOPR(df, price='Close')` | |
| FIBOTR | Fibonacci Time Retracements | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| GANNPR | W.D. Gann Price Retracements | DONE |
| dict of lists | `jhta.GANNPR(df, price='Close')` | |
| GANNTR | W.D. Gann Time Retracements | |
| JDN | Julian Day Number | DONE |
| jdn | `jhta.JDN(utc_year, utc_month, utc_day)` | |
| JD | Julian Date | DONE |
| jd | `jhta.JD(utc_year, utc_month, utc_day, utc_hour, utc_minute, utc_second)` | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| SUNC | Sun Cycle | |
| MERCYC | 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 | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| 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 | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| HT_SINE | Hilbert Transform - SineWave | |
| HT_TRENDLINE | Hilbert Transform - Instantaneous Trendline | |
| HT_TRENDMODE | Hilbert Transform - Trend vs Cycle Mode | |
| TS | Trend Score | DONE |
| list | `jhta.TS(df, n, price='Close')` | |

**data**

| | | |
|---|---|---|
| CSV2DF | CSV file 2 DataFeed | DONE |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| dict of tuples | `jhta.CSV2DF(csv_file_path)` | |
| CSVURL2DF | CSV file url 2 DataFeed | DONE |
| dict of tuples | `jhta.CSVURL2DF(csv_file_url)` | |
| DF2CSV | DataFeed 2 CSV file | DONE |
| csv file | `jhta.DF2CSV(df, csv_file_path)` | |
| DF2DFREV | DataFeed 2 DataFeed Reversed | DONE |
| dict of tuples | `jhta.DF2DFREV(df)` | |
| DF2DFWIN | DataFeed 2 DataFeed Window | DONE |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| dict of tuples | `jhta.DF2DFWIN(df, start=0, end=10)` | |
| DF_HEAD DataFeed | DONE HEAD | |
| dict of tuples | `jhta.DF_HEAD(df, n=5)` | |
| DF_TAIL DataFeed | DONE TAIL | |
| dict of tuples | `jhta.DF_TAIL(df, n=5)` | |
| DF2HEIKIN_ASHI DataFeed | DONE 2 Heikin-Ashi DataFeed | |
| dict of tuples | `jhta.DF2HEIKIN_ASHI(df)` | |

**event_driven**

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| ASI | Accumulation Swing Index (J. Welles Wilder) | DONE |
| list | `jhta.ASI(df, L)` | |
| SI | Swing Index (J. Welles Wilder) | DONE |
| list | `jhta.SI(df, L)` | |
| **experimental** | | |
| JH_SAVGP | Swing Average Price - previous Average Price | DONE |
| list | `jhta.JH_SAVGP(df)` | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| JH_SAVGPS | Saving Average Price - previous Average Price Summation | DONE |
| list | `jhta.JH_SAVGPS(df)` | |
| JH_SCO | Saving Close - Open | DONE |
| list | `jhta.JH_SCO(df)` | |
| JH_SCOS | Saving Close - Open Summation | DONE |
| list | `jhta.JH_SCOS(df)` | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| JH_SMEDP | Swing Median Price - previous Median Price | DONE |
| list | `jhta.JH_SMEDP(df)` | |
| JH_SMEDPS | Swings Median Price - previous Median Price Summation | DONE |
| list | `jhta.JH_SMEDPS(df)` | |
| JH_SPP | Swing Price - previous Price | DONE |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| list | `jhta.JH_SPP(df, price='Close')` | |
| JH_SPPS | Swing Price - previous Price Summation | DONE |
| list | `jhta.JH_SPPS(df, price='Close')` | |
| JH_STYPP | Swing Typical Price - previous Typical Price | DONE |
| list | `jhta.JH_STYPP(df)` | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| JH_STYPPS | Swing Typical Price - previous Typical Price Summation | DONE |
| list | `jhta.JH_STYPPS(df)` | |
| JH_SWCLP | Swing Weighted Close Price - previous Weighted Close Price | DONE |
| list | `jhta.JH_SWCLP(df)` | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| JH_SWCLPS | Weighted Close Price - previous Weighted Close Price Summation | DONE |
| list | `jhta.JH_SWCLPS(df)` | |

**general**

| | | |
|---|---|---|
| NORMALIZE | Normalize | DONE |
| list | `jhta.NORMALIZE(df, price_max='High', price_min='Low', price='Close')` | |
| STANDARDIZE | Standardize | DONE |
| list | `jhta.STANDARDIZE(df, price='Close')` | |
| SPREAD | Spread | DONE |
| list | `jhta.SPREAD(df1, df2, price1='Close', price2='Close')` | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| CP | Comparative Performance | DONE |
| list | `jhta.CP(df1, df2, price1='Close', price2='Close')` | |
| CRSI | Comparative Relative Strength Index | DONE |
| list | `jhta.CRSI(df1, df2, n, price1='Close', price2='Close')` | |
| CS | Comparative Strength | DONE |
| list | `jhta.CS(df1, df2, price1='Close', price2='Close')` | |
| HR | Hit Rate / Win Rate | DONE |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| float | `jhta.HR(hit_trades_int, total_trades_int)` | |
| PLR | Profit/Loss Ratio | DONE |
| float | `jhta.PLR(mean_trade_profit_float, mean_trade_loss_float)` | |
| EV | Expected Value | DONE |
| float | `jhta.EV(hitrade_float, mean_trade_profit_float, mean_trade_loss_float)` | |
| POR | Probability of Ruin (Table of Lucas and LeBeau) | DONE |
| int | `jhta.POR(hitrade_float, profit_loss_ratio_float)` | |
| **information** | | |
| INFO | Print df Information | DONE |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| print | `jhta.INFO(df, price='Close')` | |
| INFO_TRADES | Print Trades Information | DONE |
| print | `jhta.INFO_TRADES(profit_trades_list, loss_trades_list)` | |

**math_functions**

| | | |
|---|---|---|
| EXP | Exponential | DONE |
| list | `jhta.EXP(df, price='Close')` | |
| LOG | Logarithm | DONE |
| list | `jhta.LOG(df, price='Close')` | |
| LOG10 | Base-10 Logarithm | DONE |
| list | `jhta.LOG10(df, price='Close')` | |
| SQRT | Square Root | DONE |
| list | `jhta.SQRT(df, price='Close')` | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| ACOS | Arc Cosine | DONE |
| list | `jhta.ACOS(df, price='Close')` | |
| ASIN | Arc Sine | DONE |
| list | `jhta.ASIN(df, price='Close')` | |
| ATAN | Arc Tangent | DONE |
| list | `jhta.ATAN(df, price='Close')` | |
| COS | Cosine | DONE |
| list | `jhta.COS(df, price='Close')` | |
| SIN | Sine | DONE |
| list | `jhta.SIN(df, price='Close')` | |
| TAN | Tangent | DONE |
| list | `jhta.TAN(df, price='Close')` | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| ACOSH | Inverse Hyperbolic Cosine | DONE |
| list | `jhta.ACOSH(df, price='Close')` | |
| ASINH | Inverse Hyperbolic Sine | DONE |
| list | `jhta.ASINH(df, price='Close')` | |
| ATANH | Inverse Hyperbolic Tangent | DONE |
| list | `jhta.ATANH(df, price='Close')` | |
| COSH | Hyperbolic Cosine | DONE |
| list | `jhta.COSH(df, price='Close')` | |
| SINH | Hyperbolic Sine | DONE |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| list | `jhta.SINH(df, price='Close')` | |
| TANH | Hyperbolic Tangent | DONE |
| list | `jhta.TANH(df, price='Close')` | |
| PI | Mathematical constant PI | DONE |
| float | `jhta.PI()` | |
| E | Mathematical constant E | DONE |
| float | `jhta.E()` | |
| TAU | Mathematical constant TAU | DONE |
| float | `jhta.TAU()` | |
| PHI | Mathematical constant PHI | DONE |
| float | `jhta.PHI()` | |
| CEIL | Ceiling | DONE |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| list | `jhta.CEIL(df, price='Close')` | |
| FLOOR | Floor | DONE |
| list | `jhta.FLOOR(df, price='Close')` | |
| DEGREES | Radians to Degrees | DONE |
| list | `jhta.DEGREES(df, price='Close')` | |
| RADIANS | Degrees to Radians | DONE |
| list | `jhta.RADIANS(df, price='Close')` | |
| ADD | Addition High + Low | DONE |
| list | `jhta.ADD(df)` | |
| DIV | Division High / Low | DONE |
| list | `jhta.DIV(df)` | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| MAX | Highest value over a specified period | DONE |
| list | `jhta.MAX(df, n, price='Close')` | |
| MAXINDEX | Index of highest value over a specified period | DONE |
| MIN | Lowest value over a specified period | DONE |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| 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 | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| MINMAXINDEX | Indexes of lowest and highest values over a specified period | |
| MULT | Multiply High * Low | DONE |
| list | `jhta.MULT(df)` | |
| SUB | Subtraction High - Low | DONE |
| list | `jhta.SUB(df)` | |
| SUM | Summation | DONE |
| list | `jhta.SUM(df, n, price='Close')` | |

**momentum_indicators**

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| ADX | Average Directional Movement Index | |
| ADXR | Average Directional Movement Index Rating | |
| APO | Absolute Price Oscillator | DONE |
| list | `jhta.APO(df, n_fast, n_slow, price='Close')` | |
| AROON | Aroon | |
| AROONOSC | Aroon Oscillator | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| BOP | Balance Of Power | |
| CCI | Commodity Channel Index | |
| CMO | Chande Momentum Oscillator | |
| DX | Directional Movement Index | |
| IMI | Intraday Momentum Index | DONE |
| list | `jhta.IMI(df)` | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| 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 | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| MINUS_DM | Minus Directional Movement | |
| MOM | Momentum | DONE |
| list | `jhta.MOM(df, n, price='Close')` | |
| PLUS_DI | Plus Directional Indicator | |
| PLUS_DM | Plus Directional Movement | |
| PPO | Percentage Price Oscillator | |
| ROC | Rate of Change | DONE |

| Indicator / Name / Returns / Params | TODO |
|---|---|
| list `jhta.ROC(df, n, price='Close')` | |
| ROCP Rate of Change Percentage<br>list `jhta.ROCP(df, n, price='Close')` | DONE |
| ROCR Rate of Change Ratio<br>list `jhta.ROCR(df, n, price='Close')` | DONE |
| ROCR100 Rate of Change Ratio 100 scale<br>list `jhta.ROCR100(df, n, price='Close')` | DONE |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| RSI | Relative Strength Index | DONE |
| 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 | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| WILLR | Williams %R | DONE |
| list | `jhta.WILLR(df, n)` | |

**overlap_studies**

| | | |
|---|---|---|
| BBANDS | Bollinger Bands | DONE |
| dict of lists | `jhta.BBANDS(df, n, f=2)` | |
| BBANDW | Bollinger Band Width | DONE |
| list | `jhta.BBANDW(df, n, f=2)` | |
| DEMA | Double Exponential Moving Average | |
| EMA | Exponential Moving Average | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| ENVP | Envelope Percent | DONE |
| dict of lists | `jhta.ENVP(df, pct=.01, price='Close')` | |
| KAMA | Kaufman Adaptive Moving Average | |
| MA | Moving Average | |
| MAMA | MESA Adaptive Moving Average | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| MAVP | Moving Average with Variable Period | |
| MIDPOINT | MidPoint over period | DONE |
| list | `jhta.MIDPOINT(df, n, price='Close')` | |
| MIDPRICE | Midpoint Price over period | DONE |
| list | `jhta.MIDPRICE(df, n)` | |
| MMR | Mayer Multiple Ratio | DONE |
| list | `jhta.MMR(df, n=200, price='Close')` | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| SAR | Parabolic SAR | DONE |
| list | `jhta.SAR(df, af_step=.02, af_max=.2)` | |
| SAREXT | Parabolic SAR - Extended | TODO |
| SMA | Simple Moving Average | DONE |
| list | `jhta.SMA(df, n, price='Close')` | |
| T3 | Triple Exponential Moving Average (T3) | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| TEMA | Triple Exponential Moving Average | |
| TRIMA | Triangular Moving Average | DONE |
| list | `jhta.TRIMA(df, n, price='Close')` | |
| WMA | Weighted Moving Average | |

**pattern_recognition**

| | | |
|---|---|---|
| CDL2CROWS | Two Crows | |
| CDL3BLACKCROWS | Three Black Crows | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| 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 | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| CDLCLOSINGMARUBOZU | Closing Marubozu | |
| CDLCONCEALBABYSWALL | 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 | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| 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 | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| 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 | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| CDLLONGLINE | Long Line Candle | |
| CDLMARUBOZU | Marubozu | |
| CDLMATCHINGLOW | Matching Low | |
| CDLMATHOLD | Mat Hold | |
| CDLMORNINGDOJISTAR | Morning Doji Star | |
| CDLMORNINGSTAR | Morning Star | |
| CDLONECK | On-Neck Pattern | |
| CDLPIERCING | Piercing Pattern | |
| CDLRICKSHAWMAN | Rickshaw Man | |
| CDLRISEFALL3METHODS | Rising/Falling Three Methods | |
| CDLSEPARATINGLINES | Separating Lines | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| 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 | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| CDLTHRUSTAR | Thrusting Star 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 |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| 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')` | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| MEDIAN_LOW | Low median of data<br><br>list | DONE<br><br>`jhta.MEDIAN_LOW(df, n, price='Close')` |
| MEDIAN_HIGH | High median of data<br><br>list | DONE<br><br>`jhta.MEDIAN_HIGH(df, n, price='Close')` |
| MEDIAN_GROUPED | Median, or 50th percentile, of grouped data<br><br>list | DONE<br><br>`jhta.MEDIAN_GROUPED(df, n, price='Close', interval=1)` |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| 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)` | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| 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 |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| 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)` | |

| Indicator / Returns | Name / Params | TODO |
|---|---|---|
| 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)` | |