

IncTA.jl tutorial notebook

```
1 import Pkg; Pkg.activate("../")
```



```
Activating project at 'C:\Users\scell\.julia\dev\IncTA'
```



Using IncTA indicators feeding one value at a time with fit!

.....

The following examples demonstrate how to use an IncTA technical analysis indicator in an incremental approach feeding new data one observation at a time.

You first need to import `IncTA.jl` library.

```
1 using IncTA
```

and also some sample data

```
1 using IncTA.SampleData: CLOSE_TMPL, V_OHLCV
```

Import also `Plots.jl` for plotting

```
1 # using Plots
```

Show close prices

```
▶ [10.5, 9.78, 10.46, 10.51, 10.55, 10.72, 10.16, 10.25, 9.4, 9.5, 9.23, 8.5, 8.8, 8.33,
```

```
1 CLOSE_TMPL
```

Calculate SMA (simple moving average)

```
1 md"""### Calculate SMA (simple moving average)"""
```

```

1 begin
2   function show_sma1()
3     ind = SMA{Float64}(period = 3) # this is a SISO indicator
4     for p in CLOSE_TMPL
5       fit!(ind, p)
6       println(value(ind))
7     end
8   end
9   show_sma1()
10 end

```

```

missing
missing
10.246666666666668
10.250000000000002
10.506666666666667
10.593333333333335
10.476666666666668
10.376666666666669
9.936666666666667
9.716666666666669
9.376666666666669
9.076666666666668
8.843333333333335
8.543333333333335
8.220000000000002
7.823333333333336
7.306666666666669
7.663333333333336
8.196666666666669
8.920000000000002
9.126666666666669
9.090000000000002
8.876666666666669
8.593333333333335
8.326666666666667
8.150000000000002
8.043333333333337
7.976666666666669
8.076666666666667

```

Calculate BB (Bollinger bands)

```

1 md""""### Calculate BB (Bollinger bands)""""

```

```

1 begin
2   function show_bb1()
3     ind = BB{Float64}(period = 3) # this is a SIMO indicator
4     for p in CLOSE_TMPL
5       fit!(ind, p)
6       println(value(ind))
7     end
8   end
9   show_bb1()
10 end

```

```

> missing
missing
IncTA.BBVal{Float64}(9.585892709687261, 10.246666666666668, 10.9074406236460
75)
IncTA.BBVal{Float64}(9.584067070444279, 10.250000000000002, 10.9159329295557
25)
IncTA.BBVal{Float64}(10.433030926552087, 10.506666666666667, 10.5803024067812
52)
IncTA.BBVal{Float64}(10.411246662883366, 10.593333333333335, 10.775420003783
305)
IncTA.BBVal{Float64}(10.007814640732875, 10.476666666666668, 10.945518692600
462)
IncTA.BBVal{Float64}(9.885590750381258, 10.376666666666669, 10.8677425829520
8)
IncTA.BBVal{Float64}(9.174156193150258, 9.936666666666667, 10.69917714018308)
IncTA.BBVal{Float64}(8.958012888217265, 9.716666666666669, 10.47532044511607
2)
IncTA.BBVal{Float64}(9.153756620035995, 9.376666666666669, 9.59957671329734
2)
IncTA.BBVal{Float64}(8.231865572391282, 9.076666666666668, 9.92146776094205
4)
IncTA.BBVal{Float64}(8.2441487021701, 8.843333333333335, 9.44251796449657)
IncTA.BBVal{Float64}(8.154717556218943, 8.543333333333335, 8.93194911044772
7)
IncTA.BBVal{Float64}(7.171445439346782, 8.220000000000002, 9.26855456065322
2)
IncTA.BBVal{Float64}(7.103827329960961, 7.823333333333336, 8.54283933670571
1)
IncTA.BBVal{Float64}(6.558988770992052, 7.306666666666669, 8.05434455334128

```

Show candlestick data

► [OHLCV(10.81, 11.02, 9.9, 10.5, 55.03, missing), OHLCV(10.58, 10.74, 9.78, 9.78, 117.86]

```
1 V_OHLCV
```

Calculate ATR (Average true range)

```
1 md"""### Calculate ATR (Average true range)"""
```

```

1 begin
2   function show_atr1()
3     ind = ATR{OHLCV}(period = 3) # this is a MISO indicator
4     for candle in V_OHLCV
5       fit!(ind, candle)
6       println(value(ind))
7     end
8   end
9   show_atr1()
10 end

```

```

missing
missing
1.0766666666666669
0.9144444444444445
0.7562962962962961
0.6141975308641975
0.7561316872427986
0.8207544581618654
0.8438363054412431
1.1258908702941623
0.9172605801961082
0.8948403867974054
0.9065602578649369
0.8377068385766243
1.0584712257177495
0.8023141504784997
0.904876100319
1.2899174002126665
1.2832782668084441
1.1155188445389626
0.9736792296926415
0.8191194864617609
0.8660796576411736
0.6673864384274489
0.7415909589516323
0.8277273059677546
0.9418182039785027
0.8978788026523349
0.72858586843489

```

Calculate Stoch (Stochastic)

```
1 begin
2   function show_stoch1()
3     ind = Stoch{OHLCV{Missing,Float64,Float64}}(period = 3) # this is a MIMO
4       indicator
5     for candle in V_OHLCV
6       fit!(ind, candle)
7       println(value(ind))
8     end
9   show_stoch1()
10 end
```

```
IncTA.StochVal{Float64}(53.57142857142858, missing)
IncTA.StochVal{Float64}(0.0, missing)
IncTA.StochVal{Float64}(63.15789473684218, 38.90977443609025)
IncTA.StochVal{Float64}(65.1612903225806, 42.77306168647426)
IncTA.StochVal{Float64}(67.74193548387099, 65.35370684776458)
IncTA.StochVal{Float64}(58.22784810126586, 63.71035796923915)
IncTA.StochVal{Float64}(3.8461538461539315, 43.27197914376359)
IncTA.StochVal{Float64}(27.199999999999999, 29.758000649139923)
IncTA.StochVal{Float64}(0.0, 10.348717948717969)
IncTA.StochVal{Float64}(22.285714285714317, 16.495238095238097)
IncTA.StochVal{Float64}(7.1005917159763845, 9.795435333896897)
IncTA.StochVal{Float64}(0.0, 9.795435333896897)
IncTA.StochVal{Float64}(26.785714285714366, 11.295435333896913)
IncTA.StochVal{Float64}(9.836065573770437, 12.207259953161596)
IncTA.StochVal{Float64}(9.09090909090911, 15.237562983464633)
IncTA.StochVal{Float64}(17.19745222929939, 12.041475631326307)
IncTA.StochVal{Float64}(11.965811965811977, 12.751391095340153)
IncTA.StochVal{Float64}(89.74358974358974, 39.63561797956703)
IncTA.StochVal{Float64}(92.80821917808224, 64.83920696249464)
IncTA.StochVal{Float64}(77.6422764227642, 86.73136178147871)
IncTA.StochVal{Float64}(79.2592592592593, 83.23658495336856)
IncTA.StochVal{Float64}(55.55555555555536, 70.81903041252633)
IncTA.StochVal{Float64}(13.492063492063489, 49.43562610229276)
IncTA.StochVal{Float64}(18.25396825396815, 29.100529100529045)
IncTA.StochVal{Float64}(23.076923076923116, 18.27431827431824)
IncTA.StochVal{Float64}(6.400000000000001, 15.91029711029708)
IncTA.StochVal{Float64}(28.000000000000064, 19.158974358974383)
IncTA.StochVal{Float64}(31.999999999999998, 22.133333333333336)
IncTA.StochVal{Float64}(19.444444444444457, 26.481481481481485)
```

Using IncTA indicators with TSFrames.TSFrame

The following examples demonstrate how to use an IncTA technical analysis indicator by feeding a compatible Tables.jl table such as TSFrame.

You first need to import some additional libraries:

- `MarketData.jl` : to get some random data
- `TSFrames.jl` : to get a kind of DataFrame structure which is specialized for timeseries

```
1 using MarketData
```

```
1 using TSFrames
```

Get input data

Get a `TimeSeries.TimeArray` with random prices and volume

	timestamp	Open	High	Low	Close	Volume
1	2020-01-01T00:00:00	60.63	63.31	59.15	62.47	96.3
2	2020-01-01T01:00:00	63.29	64.02	55.38	58.34	87.7
3	2020-01-01T02:00:00	58.46	58.46	51.4	52.76	71.9
4	2020-01-01T03:00:00	52.0	54.48	50.13	54.11	46.9
5	2020-01-01T04:00:00	54.42	58.4	52.57	53.51	2.4
6	2020-01-01T05:00:00	53.73	54.2	47.54	47.95	38.2
7	2020-01-01T06:00:00	48.29	52.64	46.22	51.21	73.0
8	2020-01-01T07:00:00	51.81	52.66	47.99	52.04	42.4
9	2020-01-01T08:00:00	51.66	61.38	51.66	60.69	53.0
10	2020-01-01T09:00:00	59.75	61.38	54.59	58.27	3.9
	⋮ more					

```
1 begin
2     ta = random_ohlc()
3     ta
4 end
```

Converts a `TimeSeries.TimeArray` to `TSFrames.TSFrame`

ts =

	Index	Open	High	Low	Close	Volume
1	2020-01-01T00:00:00	60.63	63.31	59.15	62.47	96.3
2	2020-01-01T01:00:00	63.29	64.02	55.38	58.34	87.7
3	2020-01-01T02:00:00	58.46	58.46	51.4	52.76	71.9
4	2020-01-01T03:00:00	52.0	54.48	50.13	54.11	46.9
5	2020-01-01T04:00:00	54.42	58.4	52.57	53.51	2.4
6	2020-01-01T05:00:00	53.73	54.2	47.54	47.95	38.2
7	2020-01-01T06:00:00	48.29	52.64	46.22	51.21	73.0
8	2020-01-01T07:00:00	51.81	52.66	47.99	52.04	42.4
9	2020-01-01T08:00:00	51.66	61.38	51.66	60.69	53.0
10	2020-01-01T09:00:00	59.75	61.38	54.59	58.27	3.9
: more						
500	2020-01-21T19:00:00	193.73	197.54	191.94	193.31	80.6

```
1 ts = TSFrame(ta)
```

Calculate Simple Moving Average (SMA) of close prices

```
1 md"""### Calculate Simple Moving Average (SMA) of close prices"""
```


	Index	IncTA.SMA
1	2020-01-01T00:00:00	missing
2	2020-01-01T01:00:00	missing
3	2020-01-01T02:00:00	57.8567
4	2020-01-01T03:00:00	55.07
5	2020-01-01T04:00:00	53.46
6	2020-01-01T05:00:00	51.8567
7	2020-01-01T06:00:00	50.89
8	2020-01-01T07:00:00	50.4
9	2020-01-01T08:00:00	54.6467
10	2020-01-01T09:00:00	57.0
	: more	
500	2020-01-21T19:00:00	196.12

```
1 SMA(ts; period = 3)
```

```
1 # plot(ts)
```

Calculate Simple Moving Average (SMA) of open prices

```
1 md"""### Calculate Simple Moving Average (SMA) of open prices"""
```


	Index	IncTA.SMA
1	2020-01-01T00:00:00	missing
2	2020-01-01T01:00:00	missing
3	2020-01-01T02:00:00	60.7933
4	2020-01-01T03:00:00	57.9167
5	2020-01-01T04:00:00	54.96
6	2020-01-01T05:00:00	53.3833
7	2020-01-01T06:00:00	52.1467
8	2020-01-01T07:00:00	51.2767
9	2020-01-01T08:00:00	50.5867
10	2020-01-01T09:00:00	54.4067
	: more	
500	2020-01-21T19:00:00	199.07

1 SMA(ts; period = 3, default = :Open)

Calculate BB (Bollinger bands)

	Index	IncTA.BB_lower	IncTA.BB_central	IncTA.BB_upper
1	2020-01-01T00:00:00	missing	missing	missing
2	2020-01-01T01:00:00	missing	missing	missing
3	2020-01-01T02:00:00	49.8991	57.8567	65.8143
4	2020-01-01T03:00:00	50.316	55.07	59.824
5	2020-01-01T04:00:00	52.3555	53.46	54.5645
6	2020-01-01T05:00:00	46.3101	51.8567	57.4032
7	2020-01-01T06:00:00	46.3278	50.89	55.4522
8	2020-01-01T07:00:00	46.8695	50.4	53.9305
9	2020-01-01T08:00:00	46.0733	54.6467	63.2201
10	2020-01-01T09:00:00	49.7125	57.0	64.2875
: more				
500	2020-01-21T19:00:00	188.299	196.12	203.941

```
1 BB(ts; period = 3)
```

Calculate ATR (Average true range)

	Index	IncTA.ATR
1	2020-01-01T00:00:00	missing
2	2020-01-01T01:00:00	missing
3	2020-01-01T02:00:00	6.62
4	2020-01-01T03:00:00	5.86333
5	2020-01-01T04:00:00	5.85222
6	2020-01-01T05:00:00	6.12148
7	2020-01-01T06:00:00	6.22099
8	2020-01-01T07:00:00	5.70399
9	2020-01-01T08:00:00	7.04266
10	2020-01-01T09:00:00	6.95844
⋮ more		
500	2020-01-21T19:00:00	6.28099

```
1 ATR(ts; period = 3)
```

Calculate Stoch (Stochastic)

	Index	IncTA.Stoch_k	IncTA.Stoch_d
1	2020-01-01T00:00:00	79.8077	missing
2	2020-01-01T01:00:00	34.2593	missing
3	2020-01-01T02:00:00	10.7765	41.6145
4	2020-01-01T03:00:00	28.6537	24.5632
5	2020-01-01T04:00:00	40.5762	26.6688
6	2020-01-01T05:00:00	3.77532	24.3351
7	2020-01-01T06:00:00	40.9688	28.4401
8	2020-01-01T07:00:00	72.9323	39.2255
9	2020-01-01T08:00:00	95.4485	69.7832
10	2020-01-01T09:00:00	76.7737	81.7182
⋮ more			
500	2020-01-21T19:00:00	13.6157	22.2265

```
1 Stoch(ts; period = 3)
```