OnlineTechnicalIndicators.jl tutorial notebook

```julia
begin
    import Pkg;
    #Pkg.activate("..");
    #Pkg.add("OnlineTechnicalIndicators")
    Pkg.develop("OnlineTechnicalIndicators")
end
```

```
   Resolving package versions...
     Project No packages added to or removed from `~/.julia/environments/v1.12/P
roject.toml`
    Manifest No packages added to or removed from `~/.julia/environments/v1.12/M
anifest.toml`
```

Using OnlineTechnicalIndicators indicators feeding one value at a time with `fit!`

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

You first need to import OnlineTechnicalIndicators.jl library.

and also some sample data

```julia
begin
    using OnlineTechnicalIndicators.Indicators: SMA, BB, ATR, Stoch
    using OnlineTechnicalIndicators.Indicators: fit!, value
    using OnlineTechnicalIndicators.Candlesticks: OHLCV
    using OnlineTechnicalIndicators.SampleData: CLOSE_TMPL, V_OHLCV
end
```

Import also Plots.jl for plotting

```julia
# 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, 7.53, 
```

```julia
CLOSE_TMPL
```

Calculate SMA (simple moving average)

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

```
begin
    function show_sma1()
        ind = SMA{Float64}(period = 3)  # this is a SISO indicator
        for p in CLOSE_TMPL
            fit!(ind, p)
            println(value(ind))
        end
    end
    show_sma1()
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
7.890000000000003
7.716666666666667
7.746666666666671
8.193333333333337
8.636666666666667
```

Calculate BB (Bollinger bands)

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

```julia
begin
    function show_bb1()
        ind = BB{Float64}(period = 3)  # this is a SIMO indicator
        for p in CLOSE_TMPL
            fit!(ind, p)
            println(value(ind))
        end
    end
    show_bb1()
end
```

```
missing
missing
OnlineTechnicalIndicators.Indicators.BBVal{Float64}(9.585892709687261, 10.246666
666666668, 10.907440623646075)
OnlineTechnicalIndicators.Indicators.BBVal{Float64}(9.584067070444279, 10.250000
000000002, 10.915932929555725)
OnlineTechnicalIndicators.Indicators.BBVal{Float64}(10.433030926552087, 10.50666
666666667, 10.580302406781252)
OnlineTechnicalIndicators.Indicators.BBVal{Float64}(10.411246662883366, 10.59333
3333333335, 10.775420003783305)
OnlineTechnicalIndicators.Indicators.BBVal{Float64}(10.007814640732875, 10.47666
6666666668, 10.945518692600462)
OnlineTechnicalIndicators.Indicators.BBVal{Float64}(9.885590750381258, 10.376666
666666669, 10.86774258295208)
OnlineTechnicalIndicators.Indicators.BBVal{Float64}(9.174156193150258, 9.9366666
6666667, 10.69917714018308)
OnlineTechnicalIndicators.Indicators.BBVal{Float64}(8.958012888217265, 9.7166666
66666669, 10.475320445116072)
OnlineTechnicalIndicators.Indicators.BBVal{Float64}(9.153756620035995, 9.3766666
66666669, 9.599576713297342)
OnlineTechnicalIndicators.Indicators.BBVal{Float64}(8.231865572391282, 9.0766666
66666668, 9.921467760942054)
OnlineTechnicalIndicators.Indicators.BBVal{Float64}(8.2441487021701, 8.843333333
333335, 9.44251796449657)
OnlineTechnicalIndicators.Indicators.BBVal{Float64}(8.154717556218943, 8.5433333
33333335, 8.931949110447727)
OnlineTechnicalIndicators.Indicators.BBVal{Float64}(7.171445439346782, 8.2200000
00000002, 9.268554560653222)
OnlineTechnicalIndicators.Indicators.BBVal{Float64}(7.103827329960961, 7.8233333
33333336, 8.542839336705711)
OnlineTechnicalIndicators.Indicators.BBVal{Float64}(6.558988779992052, 7.3066666
66666669, 8.054344553341286)
OnlineTechnicalIndicators.Indicators.BBVal{Float64}(6.1753966560666065, 7.663333
333333336, 9.151270010600065)
```

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, miss
```

```julia
V_OHLCV
```

Calculate ATR (Average true range)

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

```julia
begin
    function show_atr1()
        ind = ATR{OHLCV}(period = 3)  # this is a MISO indicator
        for candle in V_OHLCV
            fit!(ind, candle)
            println(value(ind))
        end
    end
    show_atr1()
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
0.7890572456232597
0.5560381637488397
0.6340254424992265
0.6826836283328174
0.5917890855552116
```

Calculate Stoch (Stochastic)

```julia
begin
    function show_stoch1()
        ind = Stoch{OHLCV{Missing,Float64,Float64}}(period = 3)  # this is a MIMO indicator
        for candle in V_OHLCV
            fit!(ind, candle)
            println(value(ind))
        end
    end
    show_stoch1()
end
```

```
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(53.57142857142858, mis
sing)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(0.0, missing)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(63.15789473684218, 38.909
77443609025)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(65.1612903225806, 42.7730
6168647426)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(67.74193548387099, 65.353
70684776458)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(58.22784810126586, 63.710
35796923915)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(3.8461538461539315, 43.27
197914376359)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(27.19999999999999, 29.758
000649139923)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(0.0, 10.348717948717969)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(22.285714285714317, 16.49
5238095238097)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(7.100591715976345, 9.795
435333896897)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(0.0, 9.795435333896897)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(26.785714285714366, 11.29
5435333896913)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(9.836065573770437, 12.207
259953161596)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(9.09090909090911, 15.2375
62983464633)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(17.19745222929939, 12.041
475631326307)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(11.965811965811977, 12.75
1391095340153)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(89.74358974358974, 39.635
61797956703)
OnlineTechnicalIndicators.Indicators.StochVal{Float64}(92.80821917808224, 64.839
```

Using OnlineTechnicalIndicators indicators with `TSFrames.TSFrame`

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

You first need to import some aditional libraries:

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

```julia
using MarketData
```

```julia
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	997.63	1001.54	996.32	998.41	63.5
2	2020-01-01T01:00:00	998.56	1001.46	995.74	999.03	10.9
3	2020-01-01T02:00:00	998.05	1001.12	996.54	998.85	28.4
4	2020-01-01T03:00:00	998.93	1002.58	998.93	1001.16	88.2
5	2020-01-01T04:00:00	1001.71	1003.75	999.19	1002.11	1.0
6	2020-01-01T05:00:00	1001.39	1004.68	997.99	999.26	74.3
7	2020-01-01T06:00:00	999.07	1003.86	996.17	1003.82	80.6
8	2020-01-01T07:00:00	1003.45	1009.07	1003.45	1004.74	26.0
9	2020-01-01T08:00:00	1003.82	1004.98	998.06	1002.87	38.5
10	2020-01-01T09:00:00	1003.27	1007.2	1003.06	1005.92	8.6
	plus					

```julia
begin
    ta = random_ohlcv()
    ta
end
```

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

	Index	Open	High	Low	Close	Volume
1	2020-01-01T00:00:00	997.63	1001.54	996.32	998.41	63.5
2	2020-01-01T01:00:00	998.56	1001.46	995.74	999.03	10.9
3	2020-01-01T02:00:00	998.05	1001.12	996.54	998.85	28.4
4	2020-01-01T03:00:00	998.93	1002.58	998.93	1001.16	88.2
5	2020-01-01T04:00:00	1001.71	1003.75	999.19	1002.11	1.0
6	2020-01-01T05:00:00	1001.39	1004.68	997.99	999.26	74.3
7	2020-01-01T06:00:00	999.07	1003.86	996.17	1003.82	80.6
8	2020-01-01T07:00:00	1003.45	1009.07	1003.45	1004.74	26.0
9	2020-01-01T08:00:00	1003.82	1004.98	998.06	1002.87	38.5
10	2020-01-01T09:00:00	1003.27	1007.2	1003.06	1005.92	8.6
	plus					
500	2020-01-21T19:00:00	840.69	848.83	839.26	847.2	35.3

```julia
ts = TSFrame(ta)
```

Calculate Simple Moving Average (SMA) of close prices

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

	Index	OnlineTechnicalIndicators.Indicators.SMA
1	2020-01-01T00:00:00	missing
2	2020-01-01T01:00:00	missing
3	2020-01-01T02:00:00	998.763
4	2020-01-01T03:00:00	999.68
5	2020-01-01T04:00:00	1000.71
6	2020-01-01T05:00:00	1000.84
7	2020-01-01T06:00:00	1001.73
8	2020-01-01T07:00:00	1002.61
9	2020-01-01T08:00:00	1003.81
10	2020-01-01T09:00:00	1004.51
	plus	
500	2020-01-21T19:00:00	840.973

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

```julia
# plot(ts)
```

Calculate Simple Moving Average (SMA) of open prices

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

	Index	OnlineTechnicalIndicators.Indicators.SMA
1	2020-01-01T00:00:00	missing
2	2020-01-01T01:00:00	missing
3	2020-01-01T02:00:00	998.08
4	2020-01-01T03:00:00	998.513
5	2020-01-01T04:00:00	999.563
6	2020-01-01T05:00:00	1000.68
7	2020-01-01T06:00:00	1000.72
8	2020-01-01T07:00:00	1001.3
9	2020-01-01T08:00:00	1002.11
10	2020-01-01T09:00:00	1003.51
plus		
500	2020-01-21T19:00:00	838.247

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

Calculate BB (Bollinger bands)

	Index	OnlineTechnicalIndicators.Indicators.BB_lower	OnlineTechnicalIndicat
1	2020-01-01T00:00:00	missing	missing
2	2020-01-01T01:00:00	missing	missing
3	2020-01-01T02:00:00	998.242	998.763
4	2020-01-01T03:00:00	997.582	999.68
5	2020-01-01T04:00:00	997.969	1000.71
6	2020-01-01T05:00:00	998.474	1000.84
7	2020-01-01T06:00:00	997.968	1001.73
8	2020-01-01T07:00:00	997.815	1002.61
9	2020-01-01T08:00:00	1002.28	1003.81
10	2020-01-01T09:00:00	1002.0	1004.51
plus			
500	2020-01-21T19:00:00	831.581	840.973

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

Calculate ATR (Average true range)

	Index	OnlineTechnicalIndicators.Indicators.ATR
1	2020-01-01T00:00:00	missing
2	2020-01-01T01:00:00	missing
3	2020-01-01T02:00:00	5.17333
4	2020-01-01T03:00:00	4.69222
5	2020-01-01T04:00:00	4.64815
6	2020-01-01T05:00:00	5.32877
7	2020-01-01T06:00:00	6.11584
8	2020-01-01T07:00:00	5.95056
9	2020-01-01T08:00:00	6.27371
10	2020-01-01T09:00:00	5.62581
plus		
500	2020-01-21T19:00:00	6.90853

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

Calculate Stoch (Stochastic)

	Index	OnlineTechnicalIndicators.Indicators.Stoch_k	OnlineTechnicalIndicato
1	2020-01-01T00:00:00	40.0383	missing
2	2020-01-01T01:00:00	56.7241	missing
3	2020-01-01T02:00:00	53.6207	50.1277
4	2020-01-01T03:00:00	79.2398	63.1949
5	2020-01-01T04:00:00	77.2538	70.0381
6	2020-01-01T05:00:00	18.9836	58.4924
7	2020-01-01T06:00:00	89.8942	62.0439
8	2020-01-01T07:00:00	66.4341	58.4373
9	2020-01-01T08:00:00	51.938	69.4221
10	2020-01-01T09:00:00	71.3896	63.2539
plus			
500	2020-01-21T19:00:00	89.5177	79.3878

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