How to detect outliers in a candlestick diagram

Asked 5 years ago Modified 5 years ago Viewed 1k times

5

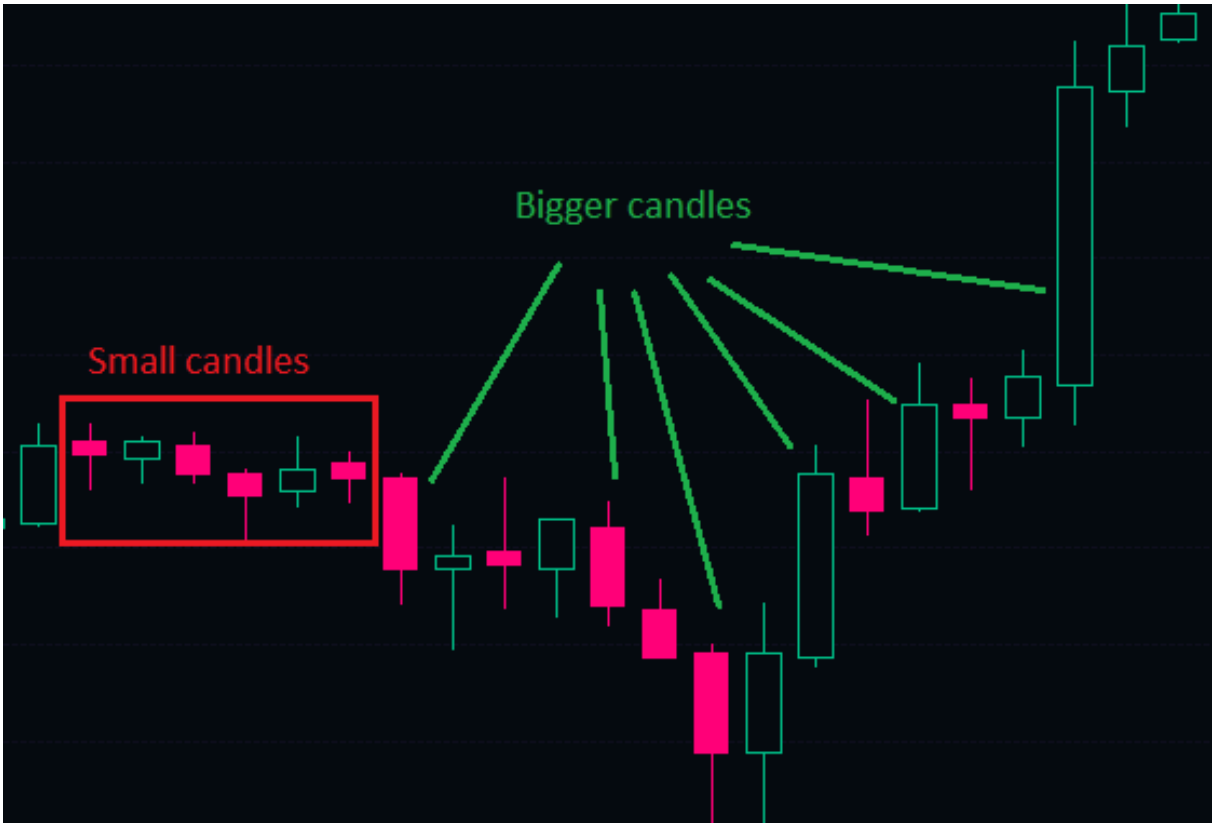
I want to implement outlier detection because I should find when the market is quiet. In other words, I have a given array of candles and for each 5 of them I should determine whether the situation is quiet or not.

When is the situation quiet?

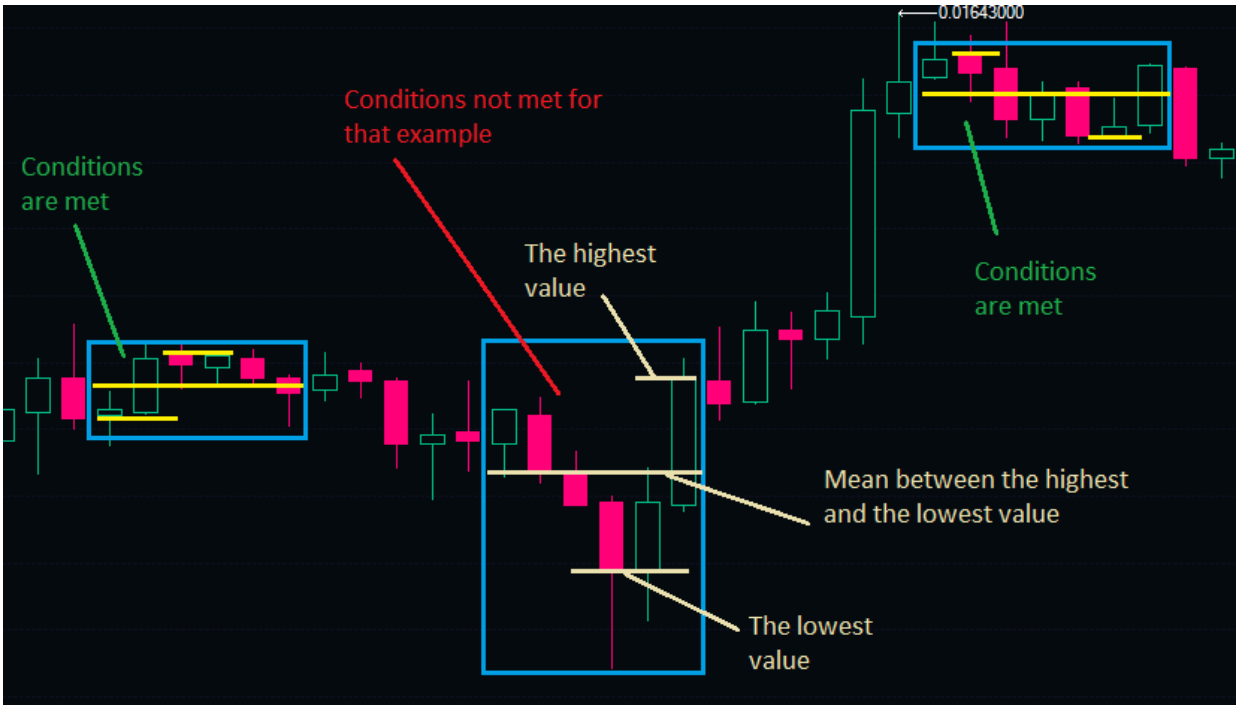
A situation is quiet when it meets the following requirements:

1. All of these 5 candles are nearly small. Candle's volume is calculated by the following formula: `Math.Abs(candle.Open - candle.Close)`.

The problem here is that I don't know how to differentiate a small from bigger candle.



2. The distance between the highest and the lowest element should not be so big.



The problem here is how to determine if the distance between the highest and the lowest value is big or small

My tries:

I tried by calculating the standard deviation but it seems like it can't help my case because the results are too random. These candles with higher volume should not meet the conditions.

- StdDev = Standard Deviation
- CV = Coefficient of Variation

```
Open time: 11/27/2019 1:00:00 AM | StdDev: 7.188532534530273E-05 | CV:
0.004710394164556892
Open time: 11/27/2019 2:00:00 AM | StdDev: 6.919176251549048E-05 | CV:
0.004528256709128957
Open time: 11/27/2019 3:00:00 AM | StdDev: 3.946517452134247E-05 | CV:
0.0025799290397687433
Open time: 11/27/2019 4:00:00 AM | StdDev: 3.6979724174201114E-05 | CV:
0.002417133418798687
Open time: 11/27/2019 5:00:00 AM | StdDev: 3.110466202999165E-05 | CV:
0.002034447120805262
Open time: 11/27/2019 6:00:00 AM | StdDev: 5.2985847166956244E-05 | CV:
0.003472887669067067
Open time: 11/27/2019 7:00:00 AM | StdDev: 8.348652585896753E-05 | CV:
0.005488201805085953 5
Open time: 11/27/2019 8:00:00 AM | StdDev: 9.300537618869152E-05 | CV:
0.006126836376066635
Open time: 11/27/2019 9:00:00 AM | StdDev: 8.136338242723194E-05 | CV:
0.005369811406232309 5
Open time: 11/27/2019 10:00:00 AM | StdDev: 3.840572873934283E-05 | CV:
0.002541742471167626
Open time: 11/27/2019 11:00:00 AM | StdDev: 7.612489737267272E-05 | CV:
```

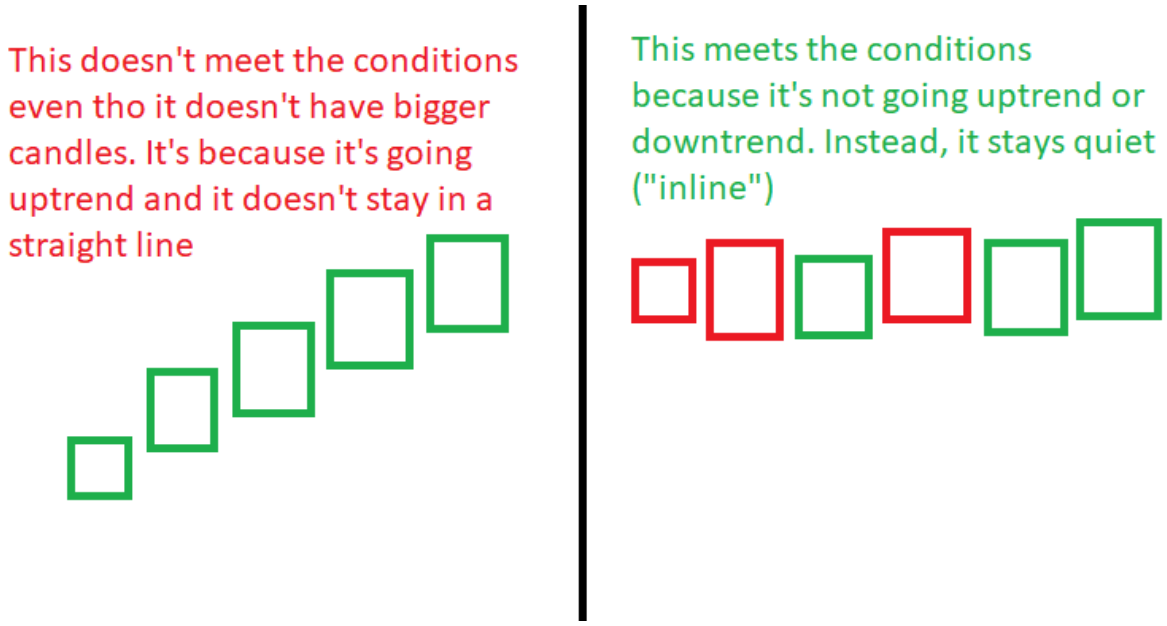
```
0.00505410286633068
Open time: 11/27/2019 12:00:00 PM | StdDev: 0.0001445423813281074 | CV:
0.009635516387447998
Open time: 11/27/2019 1:00:00 PM | StdDev: 0.0001610357103253809 2 | CV:
0.010780272481281359
Open time: 11/27/2019 2:00:00 PM | StdDev: 0.0001487615541731129 | CV:
0.009964602731134899
Open time: 11/27/2019 3:00:00 PM | StdDev: 0.0001973068675946178 5 | CV:
0.013188080181446285
Open time: 11/27/2019 4:00:00 PM | StdDev: 0.0002502149076294219 4 | CV:
0.016638842108619628
Open time: 11/27/2019 5:00:00 PM | StdDev: 0.0002476640870211103 5 | CV:
0.016331294890940345
Open time: 11/27/2019 6:00:00 PM | StdDev: 0.0001475466028073844 | CV:
0.009644829572975841
Open time: 11/27/2019 7:00:00 PM | StdDev: 0.0002146625258399797 | CV:
0.01390211293568938
Open time: 11/27/2019 8:00:00 PM | StdDev: 0.0003491847648452035 5 | CV:
0.022346394780827054
Open time: 11/27/2019 9:00:00 PM | StdDev: 0.0003890790408130464 | CV:
0.02460501111826006
Open time: 11/27/2019 10:00:00 PM | StdDev: 0.0003559950842357241 | CV:
0.022273358207828574
```

Code:

```
/// <summary>
/// Calculates the standard deviation.
/// </summary>
/// <param name="values"></param>
/// <returns></returns>
public static double StandardDeviation(this List<BinanceKline> values)
{
    if (values.Count > 0)
    {
        // Compute average
        double mean = values.Mean();

        // Perform the Sum of (value - avg)_2_2
        double sum = values.Sum(e =>
Math.Pow(Convert.ToDouble((e.GetUpperValue() + e.GetLowerValue()) / 2) - mean,
2));

        // Put it all together
        return Math.Sqrt(sum / (values.Count - 1));
    }

    return 0;
}
```

By the way, as a value, it gets the mean value of the candle `(the highest value + the lowest value) / 2`.

It might still be not clear what I want, so here is a picture:



In the left side of the picture, there are no big candles, but the graphic is going up which means it is going far from the rest candles. In the right side of the picture, the graphic stays quiet ("inline"). It doesn't go up or down but straight.

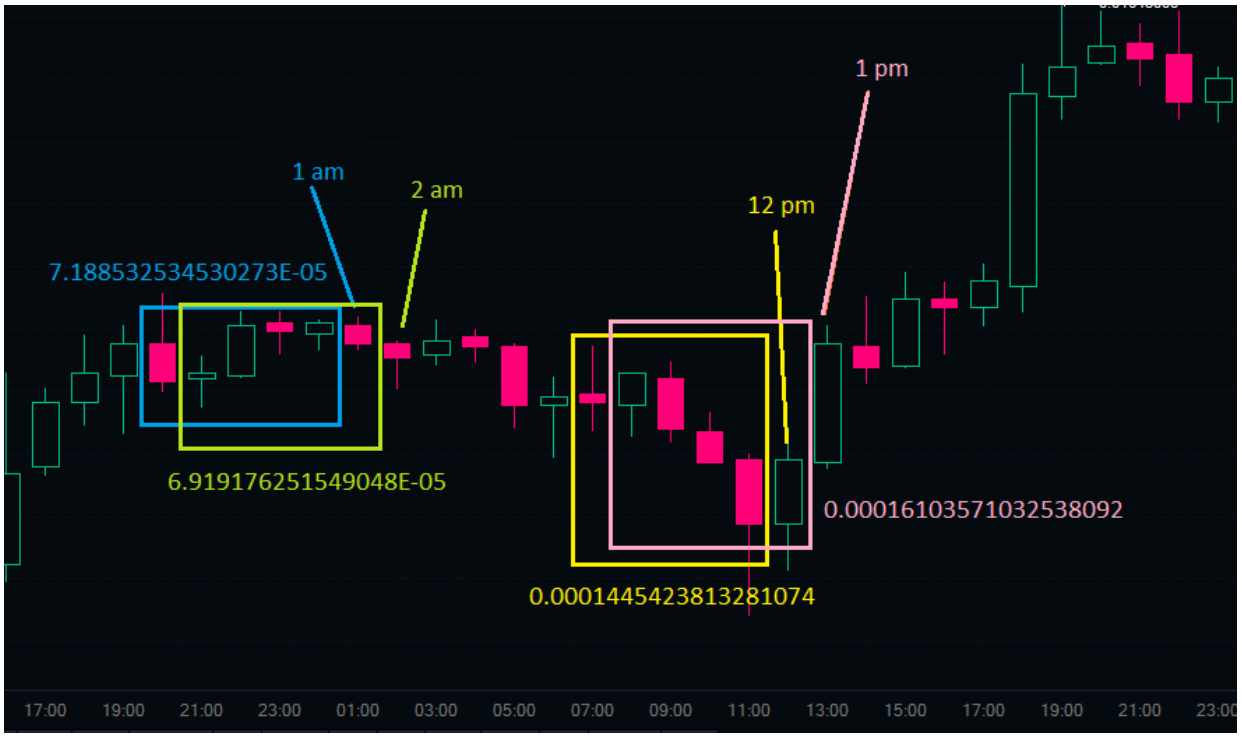
Can you suggest me how to detect those patterns? My idea is not predict for future values but by using already taken data. For example those checks will appear on each new candle and it will take the previous 5 candles excluding the new one.

Edit:

The standard deviation doesn't really work because if you look here:

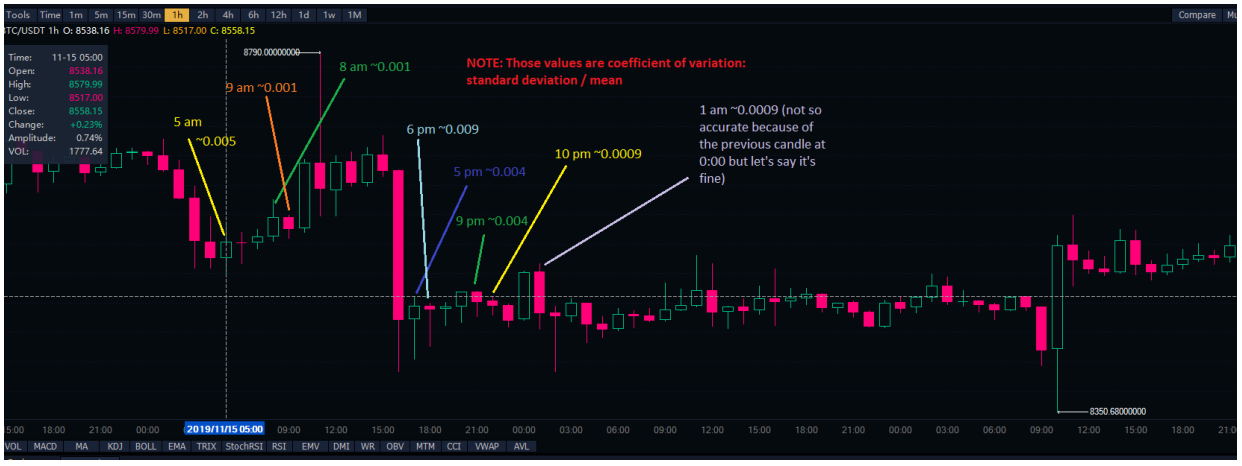
```
Open time: 11/27/2019 12:00:00 PM | StdDev: 0.0001445423813281074 | CV:
0.009635516387447998
Open time: 11/27/2019 1:00:00 PM | StdDev: 0.00016103571032538092 | CV:
0.010780272481281359
```

The standard deviation should be higher value and definitely not close to 0. The code is working fine but it's just not suitable in this case or at least not alone. In combination with something else, it might work.



Edit2:

The coefficient of variation **(CV)** seems to be more accurate. I think that might work. The optimal value for met conditions could be *0.001* as summary from both graphics. What's your opinion?



Output for that example from the picture above:

```
Open time: 11/15/2019 5:00:00 AM | StdDev: 49.24148243605151 | CV:
0.005714494887752725
Open time: 11/15/2019 6:00:00 AM | StdDev: 48.10100328995206 | CV:
0.005597444742963025
Open time: 11/15/2019 7:00:00 AM | StdDev: 34.570228051026604 | CV:
0.00403204199023016
Open time: 11/15/2019 8:00:00 AM | StdDev: 15.476614859199533 | CV:
0.0018079612622890022
Open time: 11/15/2019 9:00:00 AM | StdDev: 11.576726437123352 | CV:
0.0013526856869483194
Open time: 11/15/2019 10:00:00 AM | StdDev: 13.672853762107772 | CV:
0.0015963730504533972
Open time: 11/15/2019 11:00:00 AM | StdDev: 22.617282551182015 | CV:
```

```
0.0026365899074065743
Open time: 11/15/2019 12:00:00 PM | StdDev: 31.647747155208304 | CV:
0.0036823320018835684
Open time: 11/15/2019 1:00:00 PM | StdDev: 30.173184825271647 | CV:
0.0035044316829057126
Open time: 11/15/2019 2:00:00 PM | StdDev: 26.069585871279347 | CV:
0.0030231190376703017
Open time: 11/15/2019 3:00:00 PM | StdDev: 12.839695966026326 | CV:
0.0014866559587110115
Open time: 11/15/2019 4:00:00 PM | StdDev: 6.756121853548856 | CV:
0.0007815331369451161
Open time: 11/15/2019 5:00:00 PM | StdDev: 41.376141434406144 | CV:
0.004795673597019467
Open time: 11/15/2019 6:00:00 PM | StdDev: 79.81362075172356 | CV:
0.00928666074947359
Open time: 11/15/2019 7:00:00 PM | StdDev: 88.35104423548115 | CV:
0.010319785070149313
Open time: 11/15/2019 8:00:00 PM | StdDev: 79.06051554031244 | CV:
0.009271338691229074
Open time: 11/15/2019 9:00:00 PM | StdDev: 34.327389319609104 | CV:
0.00404133486502223
Open time: 11/15/2019 10:00:00 PM | StdDev: 8.048344239655792 | CV:
0.0009489408690167187
Open time: 11/15/2019 11:00:00 PM | StdDev: 6.030277356805678 | CV:
0.0007108047078810328
Open time: 11/16/2019 12:00:00 AM | StdDev: 7.8962145994142245 | CV:
0.0009308828374337032
Open time: 11/16/2019 1:00:00 AM | StdDev: 8.47613886153356 | CV:
0.00099890434169011
Open time: 11/16/2019 2:00:00 AM | StdDev: 9.912337262220127 | CV:
0.0011679248762150303
Open time: 11/16/2019 3:00:00 AM | StdDev: 12.102422484775284 | CV:
0.0014266919637413724
Open time: 11/16/2019 4:00:00 AM | StdDev: 12.490426133643089 | CV:
0.0014727030576166967
Open time: 11/16/2019 5:00:00 AM | StdDev: 12.640563970804447 | CV:
0.0014904318275539364
Open time: 11/16/2019 6:00:00 AM | StdDev: 15.174553947315598 | CV:
0.001790813176679162
Open time: 11/16/2019 7:00:00 AM | StdDev: 8.421461719915296 | CV:
0.0009946885198633634
Open time: 11/16/2019 8:00:00 AM | StdDev: 8.337805916426555 | CV:
0.0009848295065977384
Open time: 11/16/2019 9:00:00 AM | StdDev: 6.463572348167512 | CV:
0.0007636420370893617
Open time: 11/16/2019 10:00:00 AM | StdDev: 5.982922153931075 | CV:
0.0007068892832431122
Open time: 11/16/2019 11:00:00 AM | StdDev: 6.279278023786336 | CV:
0.0007414932821942903
Open time: 11/16/2019 12:00:00 PM | StdDev: 10.051010521335781 | CV:
0.0011860719031101202
Open time: 11/16/2019 1:00:00 PM | StdDev: 10.679646178596128 | CV:
0.0012597437184238111
Open time: 11/16/2019 2:00:00 PM | StdDev: 8.387603948685346 | CV:
0.0009891293173925218
Open time: 11/16/2019 3:00:00 PM | StdDev: 6.277716145224007 | CV:
0.0007401768729218733
Open time: 11/16/2019 4:00:00 PM | StdDev: 6.0087794101637115 | CV:
0.0007084052927520973
Open time: 11/16/2019 5:00:00 PM | StdDev: 5.152108063696313 | CV:
0.0006074484521114183
Open time: 11/16/2019 6:00:00 PM | StdDev: 5.4574552219879005 | CV:
```

```
0.0006434300416311673
Open time: 11/16/2019 7:00:00 PM | StdDev: 5.885869731823884 | CV:
0.000693675723796855
```

c# math mathematical-optimization

Share

Improve this question

Follow

edited Dec 1, 2019 at 16:10

asked Nov 30, 2019 at 17:32

nop

6,185 ● 9 ● 50 ● 155

Just to clarify, is it the case that your algorithm is basically working (it successfully differentiates between quiet and volatile periods), except you further need to exclude "quiet" climbs and falls (i.e. climbs and falls which are gradual individually, but consistent over time)? It seems to me that the answer would be to measure the trajectory over the period - the finish point relative to the start, and if the difference is beyond a threshold, then exclude it from being treated as "quiet". – Steve Nov 30, 2019 at 20:31

@Steve, not really. I explained why in my edit. From 11/27/2019 1:00:00 PM to 11/27/2019 10:00:00 PM is definitely wrong because it's close to 0 when it has to be way higher even than the previous candles. – nop Nov 30, 2019 at 22:17

Something with pattern recognition could possibly be a solution but keep in mind that this code will be executed on each candle, so it shouldn't be so slow. – nop Dec 1, 2019 at 0:19

it seems to me that the 1pm to 10pm StdDev values are indeed higher, by an order of magnitude. – Steve Dec 1, 2019 at 11:53

@Steve, yes, their bodies have higher values but keep in mind that a check at 1 pm means it compares: 8 am, 9 am, 10 am, 11 am and 12 pm (5 candles before 1 pm). If we look at the picture, 8 am and 10 am are smaller compared to the other three (9 am, 11 am and 12 pm) which are way bigger. We also see that the distance between their mean values is bigger than what we have at **1 am**. The standard deviation for 1 am should be less than what we have at 1 pm. Closer to 0 stddev (standard deviation) means less far which is wrong. – nop Dec 1, 2019 at 12:20

1 Answer

Sorted by: Highest score (default)

0

Edit:

I hope someone comes up with a better idea.

https://pastebin.com/CG3EApQJ - TRXUSDT (in the first picture in the question) - 25.11.2019 - 1 hour interval (Binance)

https://pastebin.com/mP6rFMBh - BTCUSDT (in the last picture in the question) - 15.11.2019 - 1 hour interval (Binance)

Those are the data sets.

My solution which may not work in all cases:

This solution won't work in smaller intervals when there are extreme changes in the chart.

One solution is using coefficient of variation (CV). I reduced the amount of previous candles from 5 to 3. If we look at the image below, the optimal value from CV seems to be below 0.003. Of course it has its own risks, for example I marked two candles on the chart that meet the conditions when they shouldn't. The reason is that they both have CV **~0.0024** which is below 0.003 but if I reduce the optimal value, it will make the condition get triggered less than now. That's a risk I'm willing to take.

It looks like the volumes didn't matter to my employer.

Let me know if someone has a better idea.

```
double mean = previousThree.Average(e => Convert.ToDouble((e.GetUpperValue() +
e.GetLowerValue()) / 2));
double stdDev = previousThree.StandardDeviation();
double CV = stdDev / mean;

if (CV < 0.003)
{
    ...
```

Complete output (by the way, you can see the time in the bottom part of the image):

```
Open time: 11/15/2019 6:00:00 AM | StdDev: 21.79939238449838 | CV:
0.002546419014161643
Open time: 11/15/2019 7:00:00 AM | StdDev: 5.527038839499387 | CV:
0.0006463308292687913
Open time: 11/15/2019 8:00:00 AM | StdDev: 6.721079402397433 | CV:
0.0007855694359396128
Open time: 11/15/2019 9:00:00 AM | StdDev: 9.884363071706904 | CV:
0.0011540322400728816
```

```
Open time: 11/15/2019 10:00:00 AM | StdDev: 10.56525595209722 | CV:
0.0012323968402993787
Open time: 11/15/2019 11:00:00 AM | StdDev: 20.77817023545048 | CV:
0.002418664495847048
Open time: 11/15/2019 1:00:00 PM | StdDev: 14.03211524800573 | CV:
0.0016258230041103752
Open time: 11/15/2019 2:00:00 PM | StdDev: 2.5677925020016534 | CV:
0.00029718593815783684
Open time: 11/15/2019 3:00:00 PM | StdDev: 4.399211103520008 | CV:
0.0005089810112396999
Open time: 11/15/2019 4:00:00 PM | StdDev: 5.89065856533281 | CV:
0.0006811122779212644
Open time: 11/15/2019 8:00:00 PM | StdDev: 3.4549831162148847 | CV:
0.0004076268215686829
Open time: 11/15/2019 9:00:00 PM | StdDev: 6.009339259297565 | CV:
0.0007085389024100516
Open time: 11/15/2019 10:00:00 PM | StdDev: 6.966699960048646 | CV:
0.000820984413750046
Open time: 11/15/2019 11:00:00 PM | StdDev: 3.8589765482570897 | CV:
0.0004546572963875896
Open time: 11/16/2019 12:00:00 AM | StdDev: 9.872978274057026 | CV:
0.0011639766843458792
Open time: 11/16/2019 1:00:00 AM | StdDev: 10.532862858691386 | CV:
0.0012417051013596555
Open time: 11/16/2019 2:00:00 AM | StdDev: 13.48298310216733 | CV:
0.0015886661458626638
Open time: 11/16/2019 3:00:00 AM | StdDev: 14.516428624148151 | CV:
0.0017105560467653054
Open time: 11/16/2019 4:00:00 AM | StdDev: 14.166371271429696 | CV:
0.001670413927937984
Open time: 11/16/2019 5:00:00 AM | StdDev: 3.1662714665675957 | CV:
0.0003737283544812391
Open time: 11/16/2019 6:00:00 AM | StdDev: 11.023689642461964 | CV:
0.001301947284349989
Open time: 11/16/2019 7:00:00 AM | StdDev: 8.144219320065119 | CV:
0.0009624218864802652
Open time: 11/16/2019 8:00:00 AM | StdDev: 7.101746850833978 | CV:
0.0008392963759194558
Open time: 11/16/2019 9:00:00 AM | StdDev: 3.861995382355842 | CV:
0.0004562276725712975
Open time: 11/16/2019 10:00:00 AM | StdDev: 2.094411214001339 | CV:
0.0002473434553656345
Open time: 11/16/2019 11:00:00 AM | StdDev: 6.657929482955174 | CV:
0.0007859942652924222
Open time: 11/16/2019 12:00:00 PM | StdDev: 10.629339505977429 | CV:
0.001253610389059389
Open time: 11/16/2019 1:00:00 PM | StdDev: 6.061025765770376 | CV:
0.0007143462774168898
Open time: 11/16/2019 2:00:00 PM | StdDev: 7.244390933681763 | CV:
0.0008538886060445265
Open time: 11/16/2019 3:00:00 PM | StdDev: 5.592790746428975 | CV:
0.0006595600071814469
Open time: 11/16/2019 4:00:00 PM | StdDev: 3.195912754337602 | CV:
0.0003769570371269494
Open time: 11/16/2019 5:00:00 PM | StdDev: 5.220498858666279 | CV:
0.0006154845426749733
Open time: 11/16/2019 6:00:00 PM | StdDev: 3.1226471142281893 | CV:
0.0003679999616082095
Open time: 11/16/2019 7:00:00 PM | StdDev: 2.799403210209787 | CV:
0.0003297732337714766
Open time: 11/16/2019 8:00:00 PM | StdDev: 3.1363367697571047 | CV:
0.000369478868446045
```

```
Open time: 11/16/2019 9:00:00 PM | StdDev: 5.763087280268954 | CV:
0.0006791052255747596
Open time: 11/16/2019 10:00:00 PM | StdDev: 4.148072845712606 | CV:
0.0004890656230962655
Open time: 11/16/2019 11:00:00 PM | StdDev: 9.092711825046113 | CV:
0.0010729950221168257
Open time: 11/17/2019 12:00:00 AM | StdDev: 7.247889922821099 | CV:
0.000855676558811373
Open time: 11/17/2019 1:00:00 AM | StdDev: 8.573442035340188 | CV:
0.0010120719571685153
Open time: 11/17/2019 2:00:00 AM | StdDev: 7.988032194060166 | CV:
0.0009421901067363134
```

Share

Improve this answer

Follow

edited Dec 1, 2019 at 17:06

answered Dec 1, 2019 at 15:46

nop

6,185 ● 9 ● 50 ● 155

Like I said, it's not even a good solution because it might fail in some cases, I hope someone can come up with a better solution. – nop Dec 1, 2019 at 16:57