```
using WealthLab.Backtest;
using System;
using WealthLab.Core;
using WealthLab.Data;
using WealthLab.Indicators;
using System.Collections.Generic;

namespace WealthScript4
{
	public class ScoreStrategy : UserStrategyBase
	{
		public double ScoreIt(double val)
		{
			if (val > 0.05) return 1;
			if (val < -0.05) return 0;
			return 0.5;
		}

		public ScoreStrategy()
		{
			AddParameter("Days", ParameterType.Int32, 5, 5, 30, 5);
			AddParameter("Score", ParameterType.Double, 3, 1, 6, 1);
		}

		//create indicators and other objects here, this is executed prior to the main trading
loop
		public override void Initialize(BarHistory bars)
		{
			ind4 = new PSAR(bars,0.02,0.2);
			PlotIndicator(ind4, WLColor.FromArgb(255,204,55,0), PlotStyle.Line);
			ind3 = new MACD(bars.Close,10,15);
			PlotIndicator(ind3, WLColor.FromArgb(255,0,0,0), PlotStyle.Line);
			ind2 = new MACDHist(bars.Close,10,15,5);
			PlotIndicator(ind2, WLColor.FromArgb(255,124,252,0),
PlotStyle.HistogramTwoColor);
			ind1 = new CCI(bars,10);
			PlotIndicator(ind1, WLColor.FromArgb(255,124,252,0),
PlotStyle.HistogramTwoColor);
			int days = Parameters[0].AsInt;
			double scoreThreshold = Parameters[1].AsDouble;

			const int nLines = 5;
			TimeSeries[] reglines = new TimeSeries[nLines];
			TimeSeries[] regRoc = new TimeSeries[nLines];
			score = new TimeSeries(bars.DateTimes);

			WLColor[] clr = new WLColor[5] { WLColor.Red, WLColor.Blue,
WLColor.Green, WLColor.Fuchsia, WLColor.Black };

			// Create and plot the series
			for (int n = 0; n < nLines; n++)
```

```
                {
                        reglines[n] = LR.Series(bars.Close, days);
                        regRoc[n] = ROC.Series(reglines[n], 1); // 1-day rate of change
                        days += 2;

                        PlotTimeSeries(reglines[n], reglines[n].Description, "Price", clr[n],
PlotStyle.Line);
                        PlotTimeSeries(regRoc[n], regRoc[n].Description, "rocPane", clr[n],
PlotStyle.Line);
                }

                // Now that we have the series let's create a score series
                for (int bar = 5; bar < bars.Count; bar++)
                {
                        // calculate score
                        double sum = 0;
                        for (int n = 0; n < nLines; n++)
                        {
                                sum += ScoreIt(regRoc[n][bar]);

                        }
                        score[bar] = sum;
                }

                PlotTimeSeries(score, "score", "scorePane", WLColor.Blue,
PlotStyle.Histogram);

                //calculate mean turnover (bars)

                turnover = bars.Close * bars.Volume;
                meanturnover = SMA.Series(turnover, 20);

                StartIndex = 3;
        }

        //execute the strategy rules here, this is executed once for each bar in the backtest
history
        public override void Execute(BarHistory bars, int idx)
        {
                if (!HasOpenPosition(bars, PositionType.Long))
                {
                        //code your buy conditions here
                        if (score[idx] + score[idx - 1] + score[idx - 2] >= (scoreThreshold + 6))
                                PlaceTrade(bars, TransactionType.Buy,
OrderType.MarketClose);
                }
                else
                {
                        //code your sell conditions here
                        Position p = LastPosition;
                        if (score[idx] + score[idx - 1] + score[idx - 2] < (scoreThreshold + 6))
```

```
                                        ClosePosition(p, OrderType.MarketClose);
                        }
                }

                //declare private variables below
                int days;
                double scoreThreshold;
                TimeSeries score, turnover, meanturnover;
                private IndicatorBase ind1;
                private IndicatorBase ind2;
                private IndicatorBase ind3;
                private IndicatorBase ind4;
        }
}
```