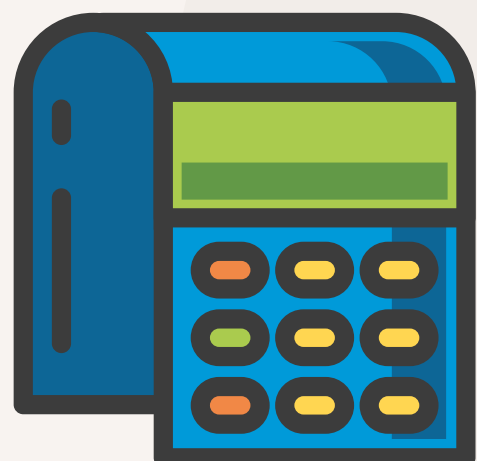


TRADING ALGORITMICO



FACULTAD DE
CIENCIAS EXACTAS
UNICEN

Ardito Lautaro
Mazzoni Lucas

Indicadores utilizados

MACD

- Señal de entrada: MACD cruza de abajo hacia arriba a la señal
- Señal de salida: MACD cruza de arriba hacia abajo a la señal

Bandas de Bollinger

- Señal de entrada: el precio toca repetidamente la banda inferior.
- Señal de salida: el precio toca repetidamente la banda superior

RSI

- Señal de entrada: cuando el RSI cruza de arriba hacia abajo el 30% .
- Señal de salida: cuando el RSI cruza de abajo hacia arriba el 70%





Parametros




```
class estrategias(bt.Strategy):  
    params = (  
        ("period", 20), # Bollinger, periodo de la media movil simple (SMA)  
        ("devfactor", 2), # Bollinger, factor de desviacion estandar  
        ('macd1', 12), # MACD, periodo de la primera media movil exponencial (EMA)  
        ('macd2', 26), # MACD, periodo de la segunda media movil exponencial (EMA)  
        ('signal', 9), # MACD, periodo de la media movil exponencial (EMA) que se utiliza como linea de senial  
        ("rsi_period", 14), # RSI, periodo utilizado para calcularlo.  
        ("rsi_overbought", 70), # RSI, nivel que se considera sobrecompra.  
        ("rsi_oversold", 30), # RSI, nivel que se considera sobreventa.  
    )
```



Inicializacion de indicadores

```
... # Crear el indicador de Bandas de Bollinger
... self.bollinger = bt.indicators.BollingerBands(self.data.close, period=self.params.period, devfactor=self.params.devfactor)
... # Creo indicador MACD
... bt.indicators.ExponentialMovingAverage(self.datas[0], period=12, plotname='Segunda Media Movil 1(Exponencial)')
... bt.indicators.ExponentialMovingAverage(self.datas[0], period=26, plotname='Segunda Media Movil 2(Exponencial)')
... self.macd = bt.indicators.MACD(self.data, period_me1=self.params.macd1, period_me2=self.params.macd2, period_signal=self.params.signal)
... self.buy_signal = bt.indicators.CrossUp(self.macd.macd, self.macd.signal)
... self.sell_signal = bt.indicators.CrossDown(self.macd.macd, self.macd.signal)
... # Creo indicador RSI
... self.rsi = bt.indicators.RelativeStrengthIndex(period=self.params.rsi_period)
```





¿Cuándo compramos?

Decidimos que para comprar acciones, no debemos poseer ninguna

```
... if self.position.size == 0 :  
...     if (self.data.close < self.bollinger.lines.bot) or (self.buy_signal) or (not self.position and self.rsi < self.params.rsi_oversold):  
...         self.log('CREAR COMPRA, %.2f' % self.data.close[0])  
...         self.order = self.buy()
```



¿Cuándo vendemos?

Nos fijamos que tengamos acciones para poder vender

```
elif self.position.size > 0 :  
    if (self.data.close > self.bollinger.lines.top and self.position.size > 0) or (self.sell_signal and self.position.size > 0) or (not self.pos  
        self.log('CREAR VENTA, %.2f' % self.data.close[0])  
        self.order = self.sell()
```



Contador de trades

Nos ayuda a saber la cantidad total de trades positivos y trades negativos



```
def notify_trade(self, trade):  
    if trade.isclosed:  
        if trade.pnl > 0:  
            self.net_profit_positive_count += 1  
        elif trade.pnl < 0:  
            self.net_profit_negative_count += 1
```

```
print("POSITIVOS: " + str(self.net_profit_positive_count))  
print("NEGATIVOS: " + str(self.net_profit_negative_count))
```

Ejemplo con acciones de Coca - Cola (KO)

desde 1/1/14 hasta 31/12/15

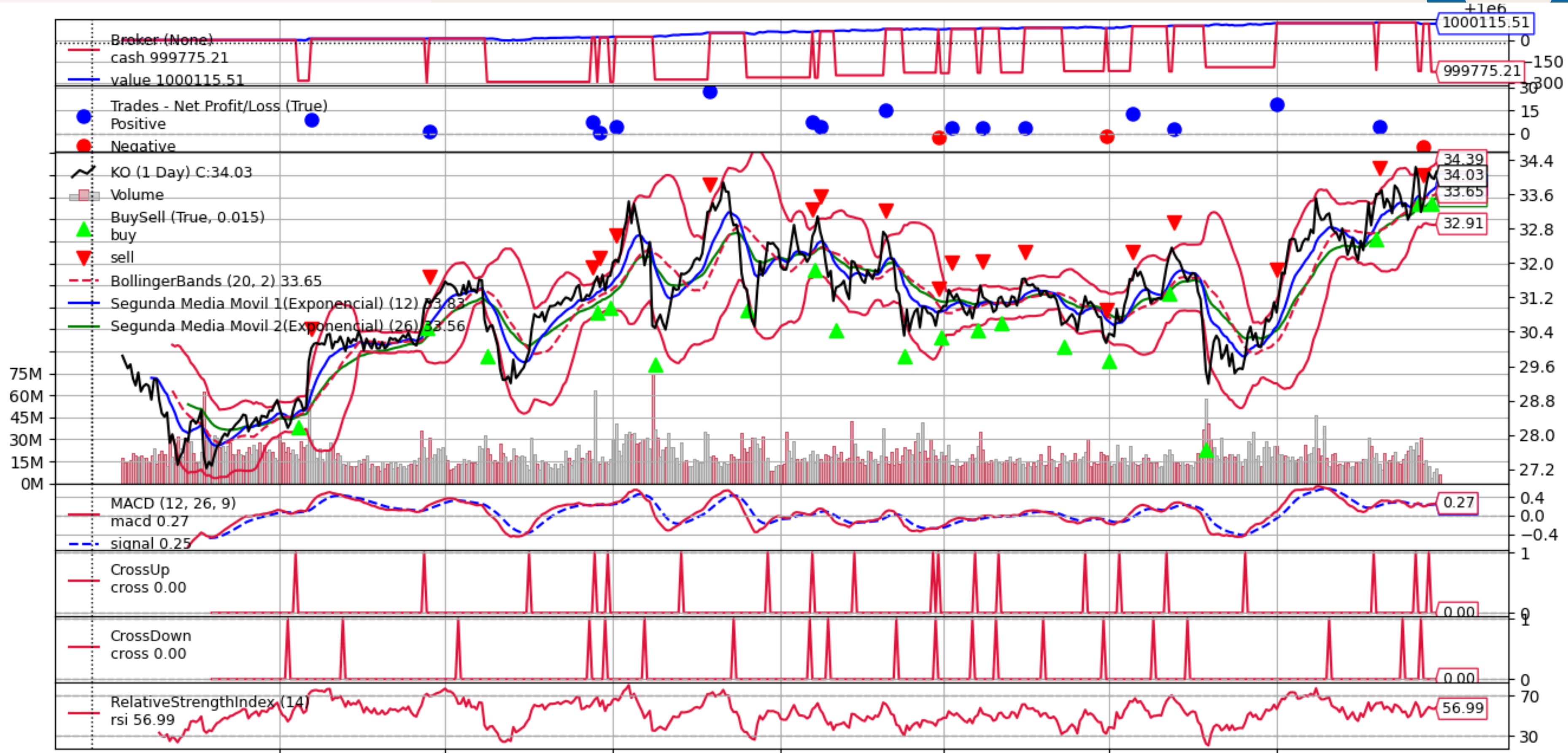
Con un portfolio inicial de 1.000.000

```
2015-06-30, CREAM COMPRA, 30.14  
2015-07-01, Compra con exito, Precio: 30.26, Costo: 302.60, Comision 0.30  
2015-07-14, CREAM VENTA, 31.63  
2015-07-15, Venta con exito, Precio: 31.62, Costo: 302.60, Comision 0.32
```

POSITIVOS: 16

NEGATIVOS: 3

Valor del portfolio final: 1000115.51





**Gracias por
su atencion!**

