

```

sma[crypto] = prices[crypto].rolling(window=30).mean()
sd[crypto] = prices[crypto].rolling(window=30).std()

POS = [False, False, False, False]
for i in range(len(prices[crypto])):
    if prices[crypto][i] > sma[crypto][i] + 2 * sd[crypto][i] and not POS[0]:
        positions[crypto].append(-0.5)
        POS[0] = True
    elif prices[crypto][i] > sma[crypto][i] + 4 * sd[crypto][i] and not POS[1]:
        positions[crypto].append(-0.5)
        POS[1] = True
    elif prices[crypto][i] < sma[crypto][i] - 2 * sd[crypto][i] and not POS[2]:
        positions[crypto].append(0.5)
        POS[2] = True
    elif prices[crypto][i] < sma[crypto][i] - 4 * sd[crypto][i] and not POS[3]:
        positions[crypto].append(0.5)
        POS[3] = True

    elif (sma[crypto][i] - sd[crypto][i] < prices[crypto][i]) and (POS[0]):
        positions[crypto].append(-2)
        POS = [False, False, False, False]

    elif (sma[crypto][i] - sd[crypto][i] < prices[crypto][i]) and (POS[1]):
        positions[crypto].append(-2)
        POS = [False, False, False, False]

    elif (prices[crypto][i] < sma[crypto][i] + sd[crypto][i]) and (POS[2]):
        positions[crypto].append(2)
        POS = [False, False, False, False]

    elif (prices[crypto][i] < sma[crypto][i] + sd[crypto][i]) and (POS[3]):
        positions[crypto].append(2)
        POS = [False, False, False, False]

    else:
        positions[crypto].append(0)

```

روی تمام کریپتوهایی که از تمرین ۵ گرفتیم پیمایش انجام می‌دهیم. به این صورت که هر روز را به Bollinger band می‌دهیم. دو مورد اول شرایط فروش، دو مورد دوم، شرایط خرید و ۴ مورد بعدی برای close است که تفاوت آنها در این است که از کدام ناحیه را پیمایش کرده است.

دلیل تعریف POS برای مواقعی است که صرفاً یکبار وارد پوزیشن شود و مکرراً به اشتباه وارد POS نشود.

```

total[crypto] = 0
for j in range(len(prices[crypto])):
    if positions[crypto][j] == 0.5:
        number_of_shares_consider_money = initial_cap / df[crypto][j]
        number_of_buy_share = number_of_shares_consider_money / 2
        total[crypto] += number_of_buy_share
        initial_cap -= number_of_buy_share * df[crypto][j]

        history[crypto]['Position'].append('BUY')
        history[crypto]['Total'].append(total[crypto])
        history[crypto]['Balance'].append(initial_cap)

    elif positions[crypto][j] == -0.5:
        number_of_shares_consider_money_s = initial_cap / df[crypto][j]
        initial_cap += number_of_shares_consider_money_s/2 * df[crypto][j]
        number_of_shares_consider_money_s /= 2
        history[crypto]['Position'].append('SELL')
        history[crypto]['Total'].append(total[crypto])
        history[crypto]['Balance'].append(initial_cap)

    elif positions[crypto][j] == -2:
        initial_cap -= (number_of_shares_consider_money_s*df[crypto][j])
        total[crypto] = 0
        history[crypto]['Position'].append('CLOSE')
        history[crypto]['Total'].append(total[crypto])
        history[crypto]['Balance'].append(initial_cap)

    elif positions[crypto][j] == 2:
        initial_cap += total[crypto] * df[crypto][j]
        total[crypto] = 0
        history[crypto]['Position'].append('CLOSE')
        history[crypto]['Total'].append(total[crypto])
        history[crypto]['Balance'].append(initial_cap)

    else:
        history[crypto]['Position'].append('EASY')
        history[crypto]['Total'].append(total[crypto])
        history[crypto]['Balance'].append(initial_cap)

```

در اینجا **backtest** را انجام دادم. به این صورت که اگر **position** بای داشته باشیم، با نصف توان خریدمان وارد سهام می‌شویم و تعداد سهم‌ها را زیاد و مقدار بودجه‌مان را کاهش می‌دهیم.

جدای از این مورد، برای فروش به این صورت فرض کرده‌ام که سیاست **borrow** را پیش گرفته‌ام. وقتی به **2SD** رسیدم فرض می‌کنم چه تعداد سهم می‌توانم بخرم با توجه به **capital** و به همان اندازه سهم غرض می‌گیرم. وقتی به **2SD** رسیدم نصف آن تعداد سهم را می‌خرم و وقتی به **4SD** رسیدم نصف دیگر آنها را. در ادامه وقتی به **SD** رسیدم آن سهم‌ها را می‌فروشم و از این طریق نیز کسب سود می‌کنم.

چون تابع sharpe ratio و equity را چون در تمرین ۴ پیاده کردم و جزو هدف این تمرین نبوده، آنها را شرح نمیدهم.

```
['USDC-USD', 'DAI-USD', 'LTC-USD', 'TUSD-USD'] Sharpe Ratio: 0.0035996257683347144
['USDC-USD', 'DAI-USD', 'LEO-USD', 'TUSD-USD'] Sharpe Ratio: 0.003441328592474485
['USDC-USD', 'DAI-USD', 'XMR-USD', 'TUSD-USD'] Sharpe Ratio: 0.0036508645752641538
['SOL-USD', 'USDC-USD', 'DAI-USD', 'XMR-USD'] Sharpe Ratio: 0.0036337236335360606
['USDC-USD', 'SHIB-USD', 'DAI-USD', 'CRO-USD'] Sharpe Ratio: 0.0030291117254165616
['USDC-USD', 'DOGE-USD', 'DAI-USD', 'XMR-USD'] Sharpe Ratio: 0.0035561358409922465
['USDC-USD', 'DAI-USD', 'ATOM-USD', 'XMR-USD'] Sharpe Ratio: 0.0035551077098601317
['USDC-USD', 'DAI-USD', 'XMR-USD'] Sharpe Ratio: 0.003554370834378109
['ETH-USD', 'USDC-USD', 'DAI-USD', 'CRO-USD'] Sharpe Ratio: 0.0037341999293072884
['USDC-USD', 'DAI-USD', 'XMR-USD', 'HBAR-USD'] Sharpe Ratio: 0.0035531421940316467
```

مقادیر sharpe ratio.

نمودارهای equity نیز در فایل کد موجود می باشد.