Analyse

July 4, 2023

[159]:
```
import backtesting
from backtesting import Backtest
backtesting.set_bokeh_output(notebook=False)

import Strategy
import talib
import yfinance as yf
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import plotly.express as px
```

[160]:
```
tickers = 'BTC-USD'
chiffre = 9
strategy= Strategy.TurtleTrading
```

1 Comparaison Stratégies

[169]:
```
data = yf.download(tickers=tickers)
bt = Backtest(data, strategy, commission=.00_06, cash=data.Close.
 ↪max()*(10**chiffre))
stats = bt.run()
print(stats)
```

```
[**********************100%**********************]  1 of 1 completed
Start                     2014-09-17 00:00:00
End                       2023-07-04 00:00:00
Duration                   3212 days 00:00:00
Exposure Time [%]                   39.962652
Equity Final [$]          151873643647974…
Equity Peak [$]           169305027371580…
Return [%]                         124.775452
Buy & Hold Return [%]             6626.378232
Return (Ann.) [%]                   9.637475
Volatility (Ann.) [%]              14.559119
Sharpe Ratio                        0.661955
Sortino Ratio                       1.117273
```

```
Calmar Ratio                         0.546537
Max. Drawdown [%]                  -17.633706
Avg. Drawdown [%]                   -4.136526
Max. Drawdown Duration      812 days 00:00:00
Avg. Drawdown Duration      108 days 00:00:00
# Trades                                 316
Win Rate [%]                        62.025316
Best Trade [%]                     330.462772
Worst Trade [%]                    -36.625487
Avg. Trade [%]                      21.900218
Max. Trade Duration         178 days 00:00:00
Avg. Trade Duration          54 days 00:00:00
Profit Factor                        9.764593
Expectancy [%]                      32.96256
SQN                                  7.852262
_strategy                      TurtleTrading
_equity_curve                              …
_trades                             Size …
dtype: object
```

[171]:
```
# Création outils d'analyse
results = bt._results.to_dict()
equity_curve = results['_equity_curve']
equity_curve = pd.merge(equity_curve, data, left_index=True, right_index=True)
trades = results['_trades']
trades.Duration = [day.days for day in trades.Duration]
```

[172]:
```
# Return over MaxDrawdown
-results["Return [%]"] / results["Max. Drawdown [%]"]
```

[172]: 7.075963261095622

2 Amélioration stratégie

[173]:
```
trades["Type"] = "Sell"
trades.loc[trades.Size > 0, "Type"] = "Buy"
trades.groupby("Type")["PnL", "ReturnPct"].agg(["mean", "count", "sum"])
```

```
/tmp/ipykernel_8169/126086596.py:3: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be
deprecated, use a list instead.
```

[173]:

	PnL			ReturnPct		
	mean	count	sum	mean	count	sum

```
Type
Buy   2.667937e+11   316  8.430682e+13  0.329626   316  104.161691
```

[174]:
```
equity_curve
```

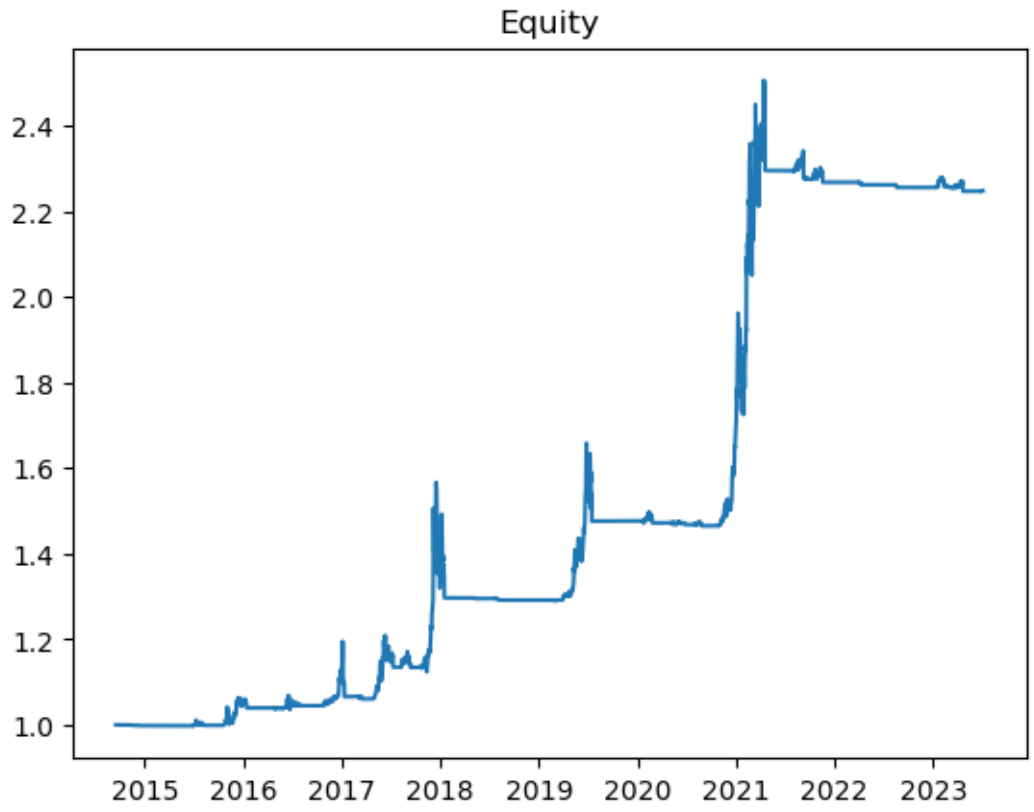
[174]:
```
                  Equity  DrawdownPct DrawdownDuration          Open  \
Date
2014-09-17  6.756683e+13     0.000000              NaT    465.864014
2014-09-18  6.756683e+13     0.000000              NaT    456.859985
2014-09-19  6.756683e+13     0.000000              NaT    424.102997
2014-09-20  6.756683e+13     0.000000              NaT    394.673004
2014-09-21  6.756683e+13     0.000000              NaT    408.084991
…                    …            …                …           …
2023-06-30  1.518077e+14     0.103348              NaT  30441.353516
2023-07-01  1.518189e+14     0.103282              NaT  30471.847656
2023-07-02  1.518220e+14     0.103264              NaT  30587.269531
2023-07-03  1.518752e+14     0.102949              NaT  30624.515625
2023-07-04  1.518736e+14     0.102958         812 days  31140.369141

                    High           Low         Close     Adj Close  \
Date
2014-09-17    468.174011    452.421997    457.334015    457.334015
2014-09-18    456.859985    413.104004    424.440002    424.440002
2014-09-19    427.834991    384.532013    394.795990    394.795990
2014-09-20    423.295990    389.882996    408.903992    408.903992
2014-09-21    412.425995    393.181000    398.821014    398.821014
…                    …             …             …             …
2023-06-30  31256.863281  29600.275391  30477.251953  30477.251953
2023-07-01  30641.289062  30328.865234  30590.078125  30590.078125
2023-07-02  30766.140625  30264.019531  30620.769531  30620.769531
2023-07-03  31375.613281  30586.513672  31156.439453  31156.439453
2023-07-04  31325.197266  30670.419922  30762.015625  30762.015625

                Volume
Date
2014-09-17     21056800
2014-09-18     34483200
2014-09-19     37919700
2014-09-20     36863600
2014-09-21     26580100
…                   …
2023-06-30  26387306197
2023-07-01   9086606733
2023-07-02  10533418042
2023-07-03  15271884873
2023-07-04  13607317504
```

```
[3213 rows x 9 columns]
```

[150]:
```
data.Close.max()*(10**chiffre)
```

[150]:
```
67566828125000.0
```

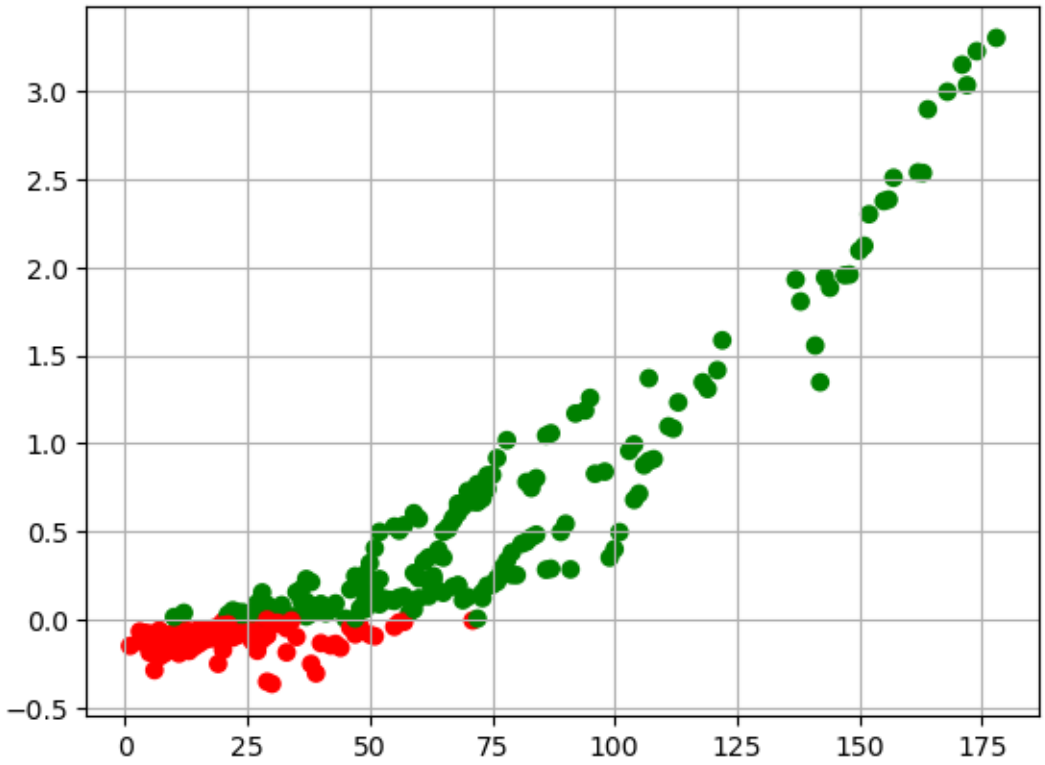
[151]:
```
plt.plot(equity_curve.Equity/equity_curve.Equity[0])
plt.title("Equity")
plt.show()
```



[152]:
```
plt.plot(data.Close/data.Close[0])
plt.title("Close Price")
plt.show()
```

[153]:
```
col = np.where(trades.ReturnPct<0, "red", "green")
```

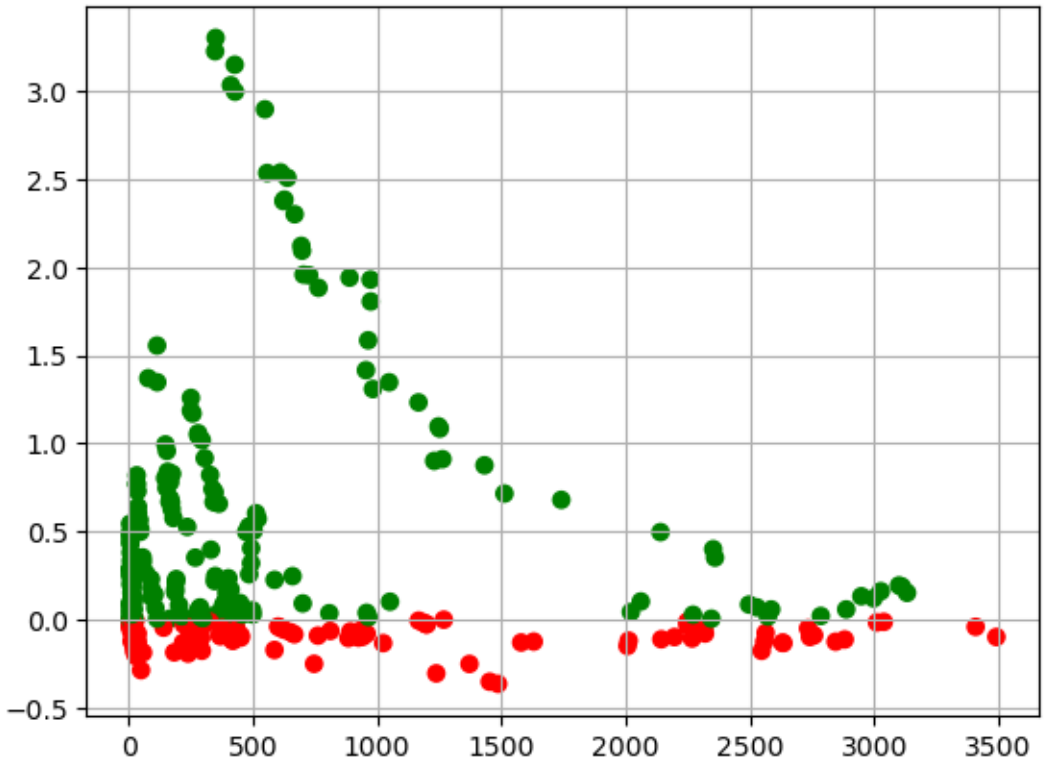
[154]:
```
plt.scatter(trades.Duration, trades.ReturnPct, c=col)
plt.grid()
plt.show()
```



[155]:
```
df = equity_curve[["Equity", "Close"]].rolling(365).apply(lambda x:␣
 ↪(x[-1]-x[0])/x[0])

fig = px.scatter(df, x=df.Close, y=df.Equity, text=[x.strftime("%d/%m/%Y") for␣
 ↪x in df.index], title="CVAM")
fig.update_traces(mode="markers")
fig.add_hline(0)
fig.show()
```

[177]:
```
# ATR
equity_curve["ATR"] = talib.ATR(equity_curve.High, equity_curve.Low,␣
 ↪equity_curve.Close)
```

[191]:
```
trades = pd.merge(trades, equity_curve.ATR, left_on="EntryTime",␣
 ↪right_index=True)
plt.scatter(trades.ATR, trades.ReturnPct, c=col)
plt.grid()
plt.show()
```


3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
0
500
1000
1500
2000
2500
3000
3500