In [8]:

```
import importlib

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import datetime

from model.Portfolio import Portfolio
from model.Optimizer import Optimizer
plt.rcParams["figure.figsize"] = 10, 15
```

In [3]:

```
names = ["C38U", "ND8U", "V01", "AGS", "S63", "CJLU"]

p = Portfolio()

# Set risk-free investment as 2%, approximately SSB's returns
p.rf = 0.02

# Add all assets
for name in names:
    p.addAsset(f"data/{name}.csv", name)

# Convert non SGD assets to SGD
p.addExchangeRate("data/forex/SGDEUR.csv", "EUR", True)
p.addExchangeRate("data/forex/USDSGD.csv", "USD", False)
```

In [4]:

```
currentWeight = [20, 20, 20, 10, 10, 20]

normalisedWeight = np.array(currentWeight)/np.sum(currentWeight)
normalisedWeight
```

Out[4]:

```
array([0.2, 0.2, 0.2, 0.1, 0.1, 0.2])
```

In [5]:

```
currentResult, currentBtPlot = p.backtest(normalisedWeight)
```

In [6]:

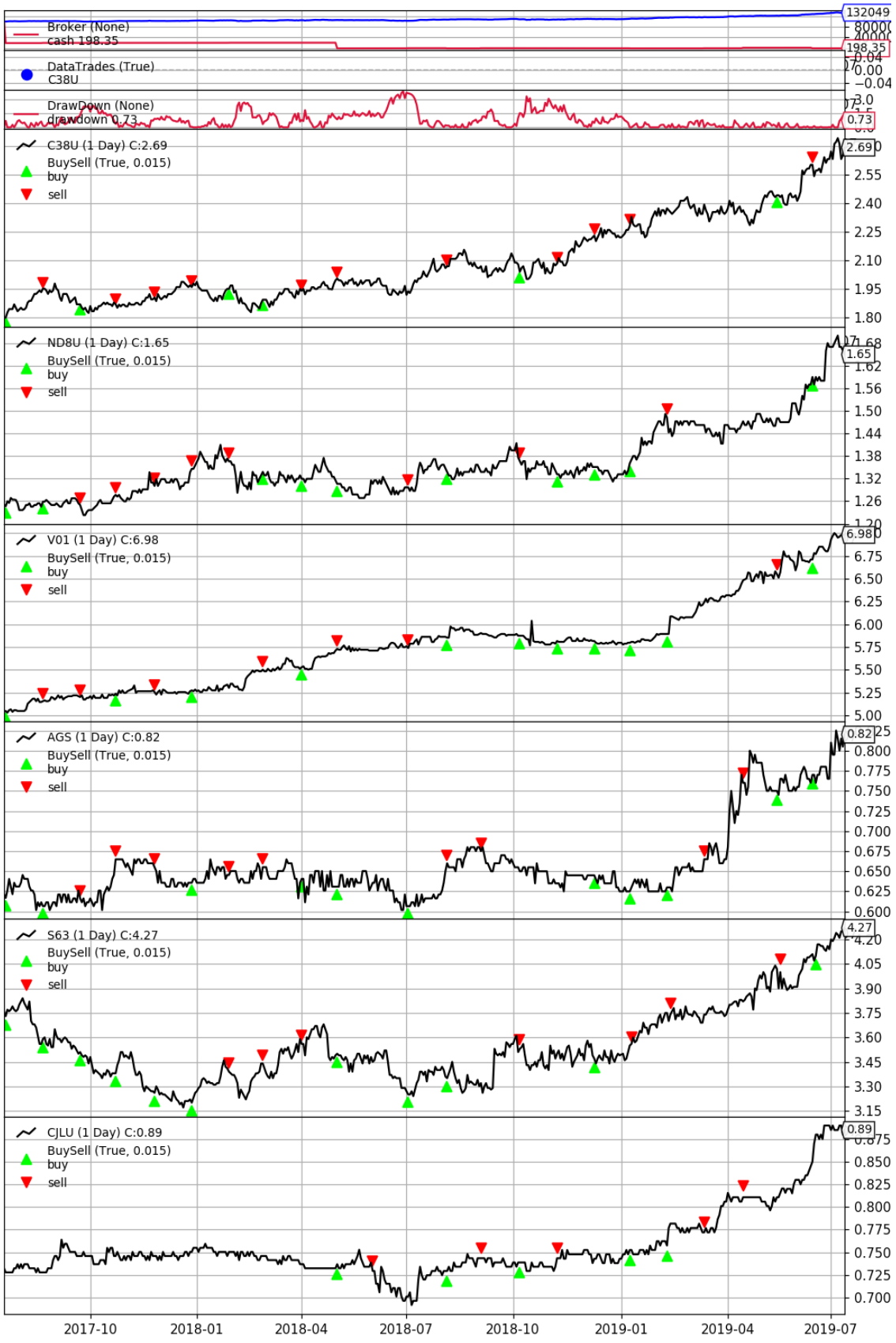
```
currentResult
```

Out[6]:

```
{'dateStart': Timestamp('2017-07-19 00:00:00'),
 'dateEnd': Timestamp('2019-07-11 00:00:00'),
 'days': 722,
 'valueStart': 100000.0,
 'valueEnd': 132049.092881,
 'sharpe': 0.9234066468402152,
 'drawdown': 0.7293828083956251,
 'drawdownPeriod': 4,
 'moneydown': 970.2200000000012,
 'maxDrawdown': 3.8642645770540343,
 'maxDrawdownPeriod': 63,
 'maxMoneydown': 4091.490180000008,
 'averageReturns': 0.10043109492243081,
 'standardDeviation': 0.08710257306210235,
 'positiveYears': 3,
 'negativeYears': 0,
 'noChangeYears': 0,
 'bestYearReturns': 0.22306415738581187,
 'worstYearReturns': 0.02905790697999988}
```

In [9]:

```
currentBtPlot()
```



Out[9]:

```
[[<Figure size 720x1080 with 9 Axes>]]
```

In [10]:

```
o = Optimizer(p)
optimisedWeight, tests = o.kfold(5)
```

In [11]:

```
optimisedWeight
```

Out[11]:

```
[0.14416416650393377,
 0.1123707125491497,
 0.4312060941113713,
 0.03956076349566985,
 0.028094159759921904,
 0.24460410358158907]
```

In [12]:

```
tests
```

Out[12]:

```
{'sharpeRaw': [24.099438196652933,
  20.44789686787606,
  18.535088709375337,
  16.940759283693644,
  89.02704291063445],
 'sharpeAvg': 33.81004519364649,
 'sharpeStd': 27.711574605986986,
 'weightsRaw': [array([0.13641491, 0.09920729, 0.41618581, 0.0424036 ,
0.05655936,
        0.24922903]),
  array([0.1541335 , 0.14841234, 0.3657246 , 0.05237152, 0.02548233,
        0.25387571]),
  array([0.13696991, 0.09833466, 0.38487679, 0.04215671, 0.02553504,
        0.3121269 ]),
  array([0.10838182, 0.12328995, 0.49823585, 0.03526044, 0.02463205,
        0.21019988]),
  array([0.18492069, 0.09260931, 0.49100743, 0.02561156, 0.00826202,
        0.19758899])],
 'weightsStd': 0    0.025110
 1    0.020876
 2    0.054275
 3    0.008854
 4    0.015678
 5    0.040152
 dtype: float64}
```

In [13]:

```
optimisedResult, optimisedBtPlot = p.backtest(optimisedWeight)
```

In [14]:

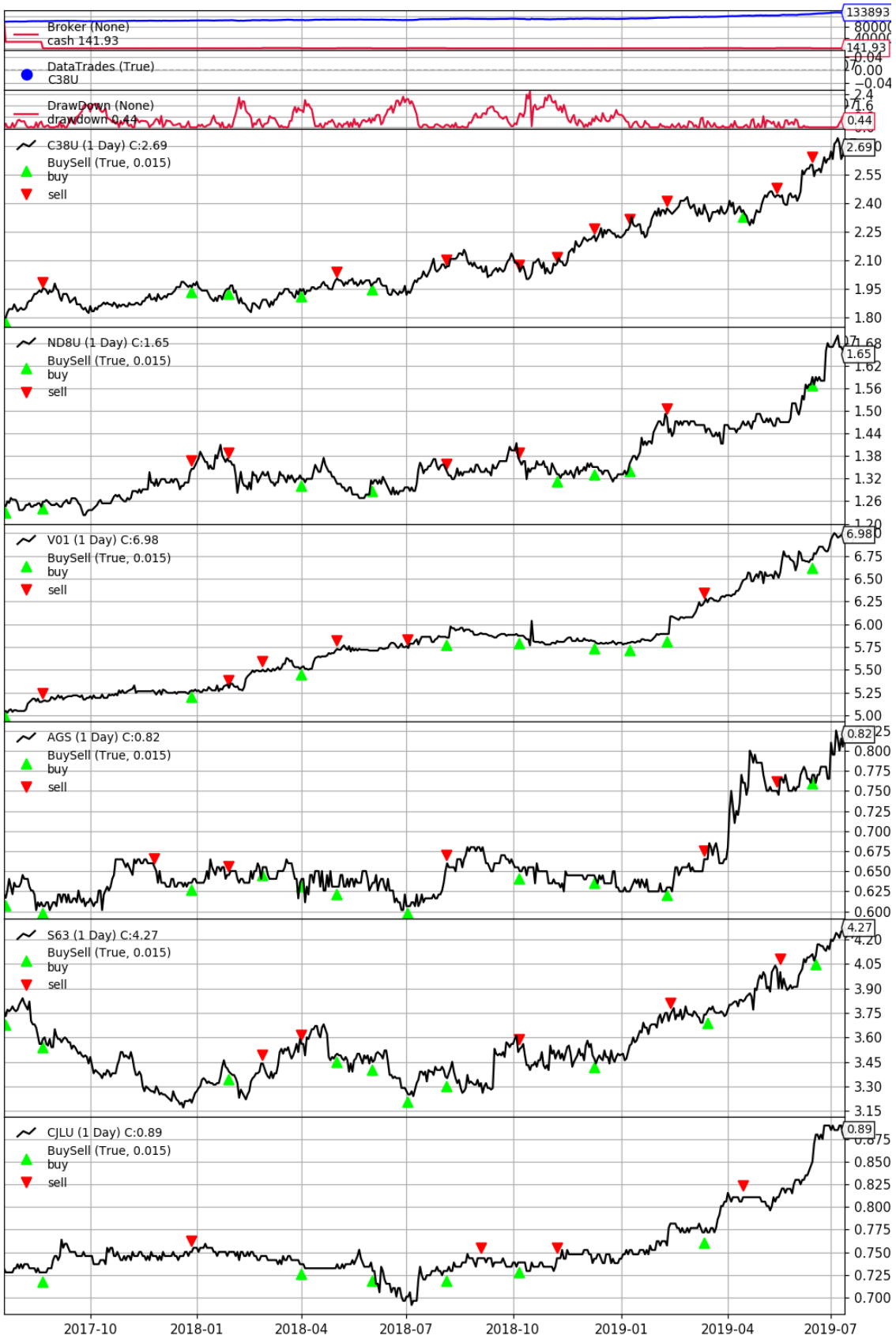
```
optimisedResult
```

Out[14]:

```
{'dateStart': Timestamp('2017-07-19 00:00:00'),
 'dateEnd': Timestamp('2019-07-11 00:00:00'),
 'days': 722,
 'valueStart': 100000.0,
 'valueEnd': 133893.480232,
 'sharpe': 1.1226714577528294,
 'drawdown': 0.43699431268727695,
 'drawdownPeriod': 5,
 'moneydown': 587.6749999999884,
 'maxDrawdown': 2.5993688555169028,
 'maxDrawdownPeriod': 68,
 'maxMoneydown': 2892.9447410000284,
 'averageReturns': 0.10467991911336035,
 'standardDeviation': 0.07542715950297518,
 'positiveYears': 3,
 'negativeYears': 0,
 'noChangeYears': 0,
 'bestYearReturns': 0.21127544076104554,
 'worstYearReturns': 0.04792807961000034}
```

In [15]:

```
optimisedBtPlot()
```



Out[15]:

```
[[<Figure size 720x1080 with 9 Axes>]]
```

In [16]:

```
dict(zip(names, np.array(optimisedWeight)*100))
```

Out[16]:

```
{'C38U': 14.416416650393376,
 'ND8U': 11.23707125491497,
 'V01': 43.12060941113713,
 'AGS': 3.956076349566985,
 'S63': 2.80941597599219,
 'CJLU': 24.460410358158907}
```

In [ ]:

```

```