In [53]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
from sklearn.model_selection import train_test_split
```

In [54]:
```
df = pd.read_csv("trades_sorted_1m.csv")
```

In [55]:
```
df.head()
```

Out[55]:

	Unnamed: 0	TICKET	LOGIN	SYMBOL	DIGITS	CMD	VOLUME	OPEN_TIME	OPEN_PRICE
0	453515	2085443811	1162	USDJPY	2	1	10	2005-03-18 19:03:57	104.6600
1	453518	2085443814	1162	USDJPY	2	1	50	2005-03-21 05:15:39	104.9900
2	453517	2085443813	1162	GBPUSD	4	0	20	2005-03-21 05:37:03	1.9150
3	453516	2085443812	1162	USDJPY	2	1	20	2005-03-21 05:37:05	104.8700
4	453521	2085443817	1162	GBPUSD	4	0	20	2005-03-21 05:54:45	1.9138

5 rows × 31 columns

In [56]:
```
df= df[df['MAGIC'] == 0]
df= df[["TICKET", "LOGIN", "SYMBOL", "PROFIT", "OPEN_TIME"]]
df= df[df['SYMBOL'].notnull()]

traders_count= df.groupby(['LOGIN']).agg({'TICKET':'count'})
traders_count= traders_count[traders_count['TICKET'] >= 100].reset_index()
traders= traders_count['LOGIN'].tolist()
filter_df= df[df['LOGIN'].isin(traders)]
filter_df
```

Out[56]:

	TICKET	LOGIN	SYMBOL	PROFIT	OPEN_TIME
0	2085443811	1162	USDJPY	-70.21	2005-03-18 19:03:57
1	2085443814	1162	USDJPY	42.89	2005-03-21 05:15:39
2	2085443813	1162	GBPUSD	20.00	2005-03-21 05:37:03
3	2085443812	1162	USDJPY	17.17	2005-03-21 05:37:05
4	2085443817	1162	GBPUSD	20.00	2005-03-21 05:54:45
...	...	...	...	...	...
999991	2094826117	72067	EURUSD	-9.39	2011-08-17 14:31:31
999992	2094826123	72067	EURUSD	-10.23	2011-08-17 14:32:04
999993	2094826142	518298	USDCHF	1.63	2011-08-17 14:34:39
999997	2094814634	566832	EURUSD	-10.93	2011-08-17 14:37:44
999998	2094826180	565900	EURUSD	6.70	2011-08-17 14:37:46

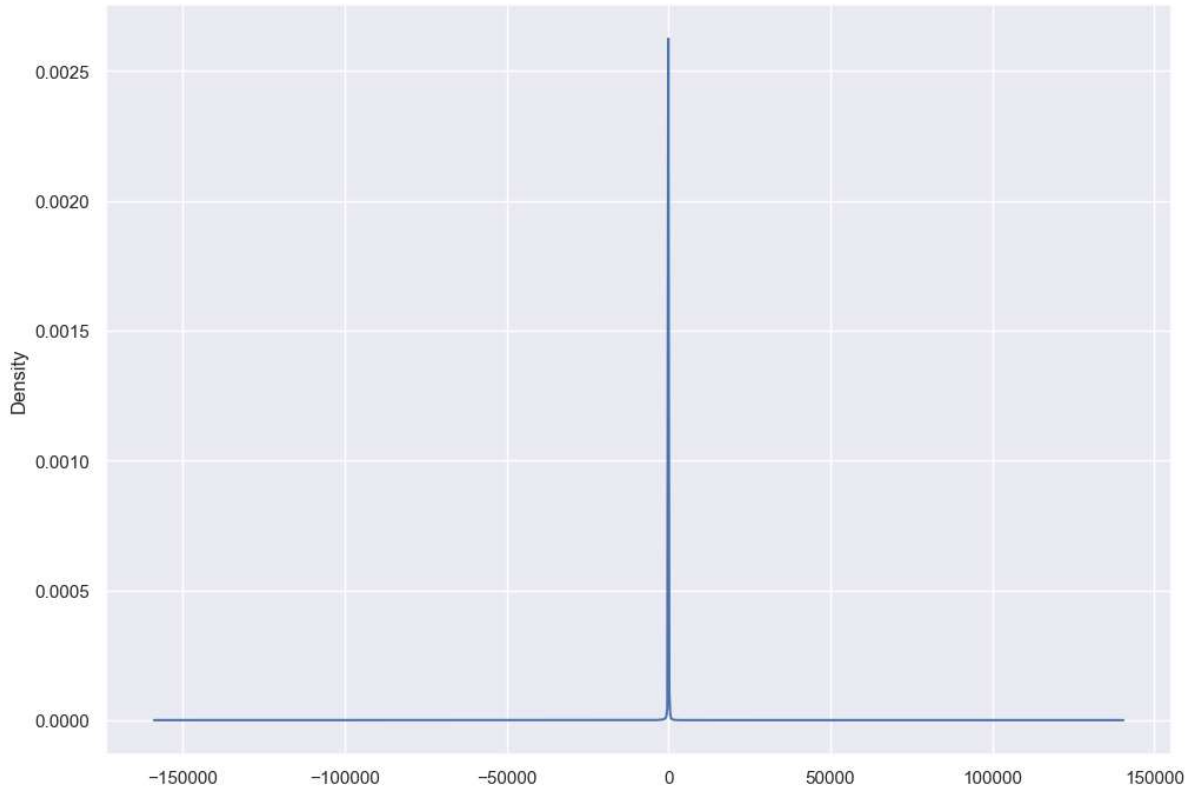
545747 rows × 5 columns

In [57]:
```
filter_df['loss'] = np.where(filter_df['PROFIT']<= 0, 1, 0)
filter_df['win'] = np.where(filter_df['PROFIT']> 0, 1, 0)
```

In [58]:
```
train, test = train_test_split(filter_df, test_size=0.2)
```

In [59]:
```
sns.set(rc={'figure.figsize':(11.7,8.27)})
train['PROFIT'].plot.kde()
```

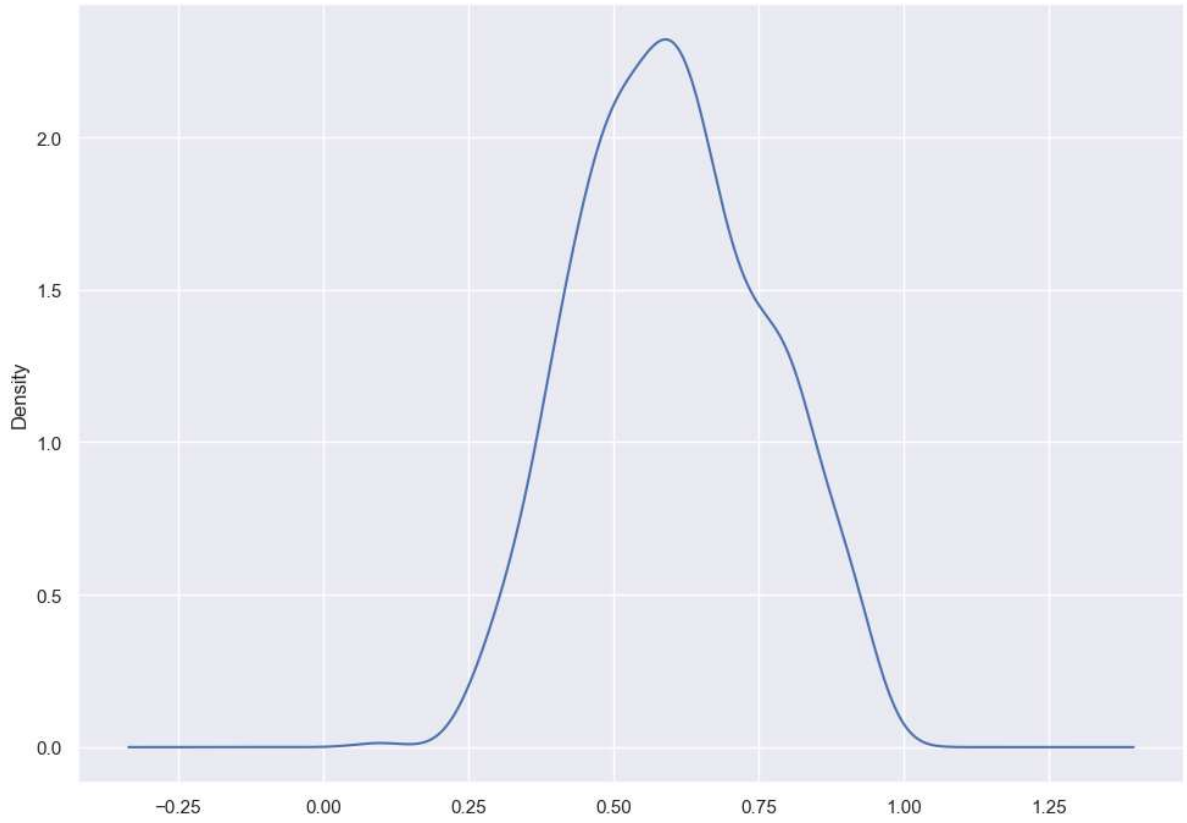
Out[59]: <AxesSubplot:ylabel='Density'>



In [60]:
```
trader_symbol_grouping= train.groupby(['LOGIN', 'SYMBOL']).agg({'TICKET':
                'loss': 'sum','win': 'sum'}).reset_index()

trader_symbol_grouping['win_rate']= trader_symbol_grouping['win'] / (trade
                                                                     trade
```

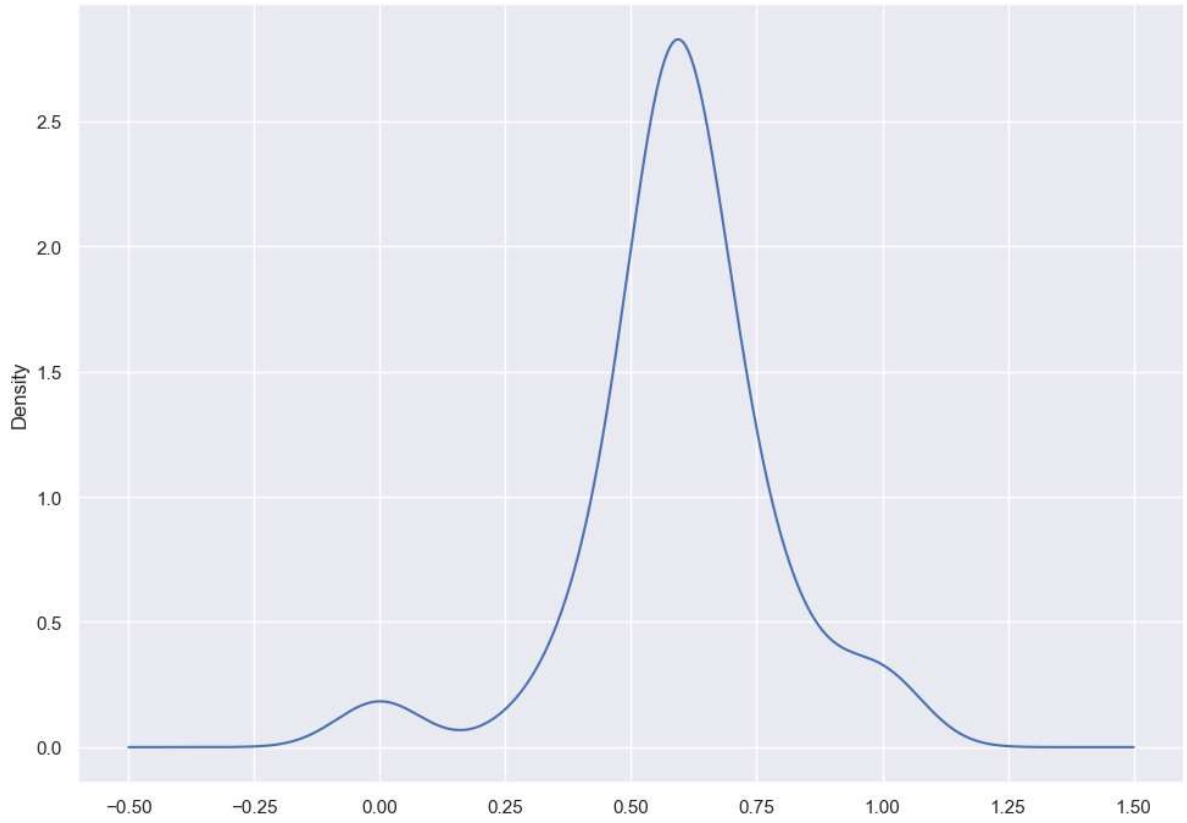
In [61]:
```
trader_grouping= trader_symbol_grouping.groupby(['LOGIN']).agg({'TICKET':
                'loss': 'sum','win': 'sum'}).reset_index()

trader_grouping['win_rate']= trader_grouping['win'] / (trader_grouping['lo
                                                                    trade

trader_grouping['win_rate'].plot.kde()
```

Out[61]: <AxesSubplot:ylabel='Density'>



In [62]:
```
symbol_grouping= train.groupby(['SYMBOL']).agg({'TICKET': 'count', 'PROFIT
                'loss': 'sum','win': 'sum'}).reset_index()

symbol_grouping['win_rate']= symbol_grouping['win'] / (symbol_grouping['lo
                                                                    symbo

symbol_grouping['win_rate'].plot.kde()
```

Out[62]: <AxesSubplot:ylabel='Density'>



Trader Performance

In [63]:
```
table1 = trader_symbol_grouping.pivot('LOGIN', 'SYMBOL', 'win_rate')
table1.to_csv('trader_symbol_performance')
```

In [64]:
```
train = train.sort_values(by=['LOGIN','SYMBOL','OPEN_TIME'], ascending=[Tr

unique_accid = train['LOGIN'].unique().tolist() #Getting the unique accoun

len(unique_accid)
```

Out[64]: 696

Back Testing Profit

In [65]:

```
agg_func_math = {
    'PROFIT':
    ['sum', 'mean', 'median', 'min', 'max', 'std', 'var', 'mad', 'prod', '
}
stats = train.groupby(['LOGIN', 'SYMBOL']).agg(agg_func_math)

stats
```

Out[65]:

		sum	mean	median	min	max	std	var	
LOGIN	**SYMBOL**								
	EURGBP	-47.29	-47.290000	-47.290	-47.29	-47.29	NaN	NaN	0.0
	EURUSD	-726.00	-10.835821	1.000	-640.00	76.00	85.642774	7334.684758	29.6
1162	**GBPJPY**	49.46	16.486667	18.620	2.90	27.94	12.655581	160.163733	9.0
	GBPUSD	-967.00	-29.303030	20.000	-980.00	100.00	201.742330	40699.967803	104.6
	USDCHF	8.12	8.120000	8.120	8.12	8.12	NaN	NaN	0.0
...	**...**	...	...	...	...	...	...	...	
	NZDUSD	-4.00	-4.000000	-4.000	-4.00	-4.00	NaN	NaN	0.0
	USDCAD	-12.95	-3.237500	-6.055	-9.71	8.87	8.265944	68.325825	6.0
6866968	**USDCHF**	-2.51	-0.627500	-0.115	-6.89	4.61	5.353008	28.654692	4.4
	USDJPY	3.01	0.130870	0.000	-61.05	48.33	21.929342	480.896026	13.3
	XAUUSD	-254.57	-84.856667	-75.600	-135.00	-43.97	46.215578	2135.879633	33.4

6603 rows × 10 columns

In [66]:
```
bad_traders = stats[(stats[('PROFIT', 'mean')] < 0) & (stats[('PROFIT', 'm
good_traders = stats[(stats[('PROFIT', 'mean')] > 0) & (stats[('PROFIT', '

bad_traders
```

Out[66]:

LOGIN	SYMBOL	sum	mean	median	min	max	std	var	
22093	EURUSD	-1432.00	-13.018182	-40.000	-490.00	820.00	175.307383	30732.678565	1(
31306	GBPUSD	-1243.00	-8.398649	-2.000	-330.00	475.00	101.628558	10328.363808	5
51023	EURUSD	-17791.84	-2.656291	-0.200	-3690.00	3258.00	67.444484	4548.758480	
	GBPUSD	-2468.60	-4.416100	-0.300	-400.00	118.00	31.046342	963.875332	
53050	EURJPY	-806.46	-6.556585	-49.500	-374.90	550.00	194.559766	37853.502388	16
...	...	...	...	...	...	...	...	...	
570406	GBPUSD	-1377.58	-7.871886	-13.100	-305.00	675.20	99.776247	9955.299522	5
571262	GBPUSD	-177.37	-1.597928	-1.570	-52.80	93.17	14.045798	197.284455	
571758	EURJPY	-4594.30	-21.468692	-1.265	-624.73	456.63	129.868649	16865.865926	7
	XAUUSD	-5356.52	-19.987015	-4.800	-1226.00	500.00	141.119402	19914.685712	8
572177	EURUSD	-833.34	-5.827552	-5.300	-125.00	34.50	21.192085	449.104479	

160 rows × 10 columns

In [67]:
```
1 good_traders
```

Out[67]:

		sum	mean	median	min	max	std	var	
LOGIN	SYMBOL								
1396	USDCAD	14769.93	11.797069	26.19	-16696.67	3513.45	588.294657	3.460906e+05	1
51040	EURJPY	2412.89	14.110468	41.58	-1031.19	224.98	165.694025	2.745451e+04	
	EURJPY	2068.84	8.879142	12.41	-746.62	164.19	57.920260	3.354757e+03	
51765	EURUSD	999.00	2.485075	6.00	-735.00	1180.00	112.824234	1.272931e+04	
	USDCHF	473.51	1.997932	15.26	-676.11	126.69	82.478057	6.802630e+03	
...	...	...	...	...	...	...	...	...	
570668	EURCHF	2206.40	21.013333	22.30	-1713.21	1306.29	395.941838	1.567699e+05	2
570943	EURUSD	93.20	0.733858	2.68	-76.92	39.20	14.853356	2.206222e+02	
571129	EURUSD	324.81	2.370876	1.63	-10.72	15.36	2.545644	6.480302e+00	
572015	EURUSD	24.23	0.031305	0.65	-67.86	15.85	6.569244	4.315497e+01	
572273	XAUUSD	10111.22	93.622407	152.00	-4940.00	5143.00	1188.681848	1.412965e+06	5

149 rows × 10 columns

In [68]:
```
1 badtraders = set()
2 for i, x in bad_traders.iterrows():
3     _id = f'{i[0]}/{i[1]}'
4     badtraders.add(_id)
5
6 badtraders
```

```
'559545/GBPUSD',
'559650/EURUSD',
'562329/GBPUSD',
'562692/GBPUSD',
'562692/NZDUSD',
'563772/AUDUSD',
'563772/EURJPY',
'563772/EURUSD',
'563772/GBPJPY',
'563772/USDCHF',
'563864/EURUSD',
'564290/EURUSD',
'564290/USDCAD',
'564301/EURUSD',
'564388/EURUSD',
'564741/USDJPY',
'564741/XAUUSD',
'565366/XAUUSD',
'566028/XAGUSD',
'566028/XAUUSD',
```

In [69]:
```
equity = 0
history = []
history.append(equity)

for i, x in test.iterrows():
    _id = f"{x['LOGIN']}/{x['SYMBOL']}"
    if _id in badtraders:
        equity += (-1 * x['PROFIT'])
        history.append(equity)
plt.plot(history)
```

Out[69]:
```
[<matplotlib.lines.Line2D at 0x1f2b4d55dc8>]
```

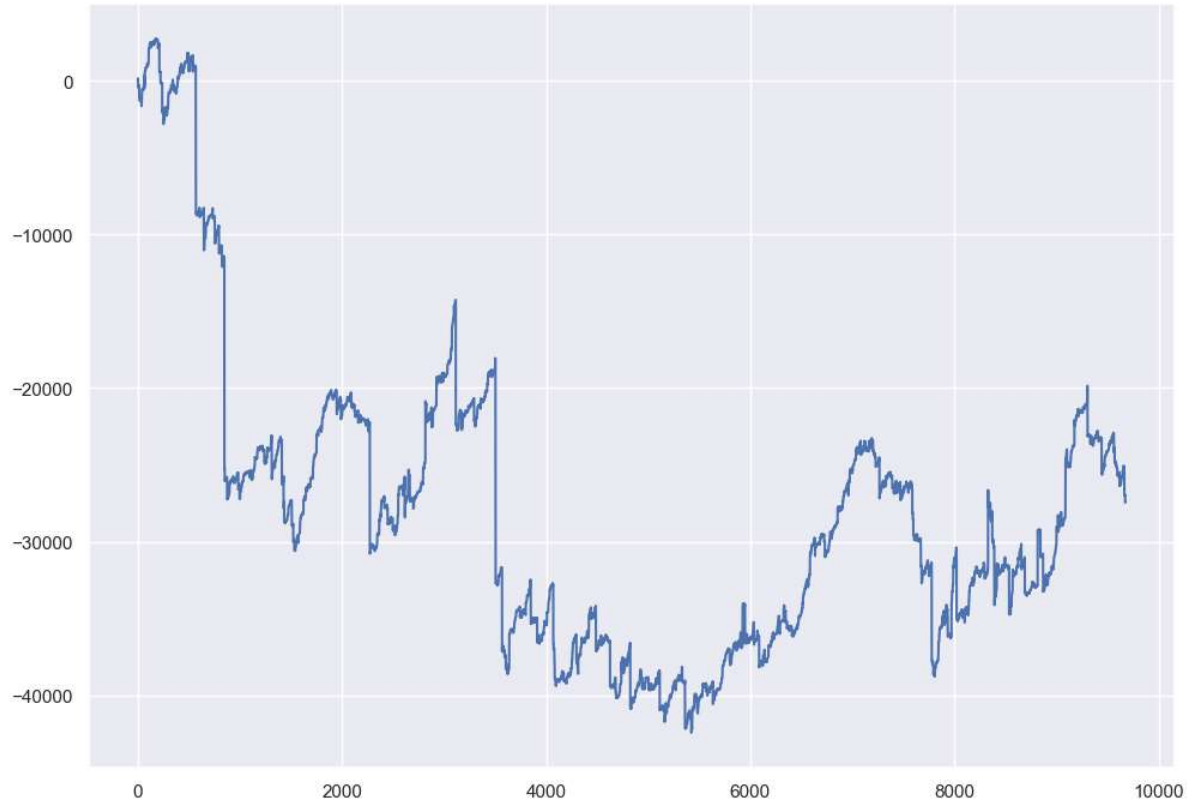
In [70]:
```
(len(history) / len(test)) * 100
```

Out[70]:
```
12.101694915254237
```

In [71]:
```
goodtraders = set()
for i, x in good_traders.iterrows():
    _id = f'{i[0]}/{i[1]}'
    goodtraders.add(_id)
```

In [72]:
```
equity = 0
history = []
history.append(equity)

for i, x in test.iterrows():
    _id = f"{x['LOGIN']}/{x['SYMBOL']}"
    if _id in goodtraders:
        equity += (1 * x['PROFIT'])
        history.append(equity)
plt.plot(history)
```

Out[72]:
```
[<matplotlib.lines.Line2D at 0x1f2b5189c88>]
```



In [73]:
```
(len(history) / len(test)) * 100
```

Out[73]:
```
8.866697205680255
```

Back Test for Win Rate

In [74]:
```
train.head()
```

Out[74]:

	TICKET	LOGIN	SYMBOL	PROFIT	OPEN_TIME	loss	win
3119	2085460744	1162	EURGBP	-47.29	2006-09-27 10:08:37	1	0
45	2085444035	1162	EURUSD	-4.00	2005-07-13 08:45:04	1	0
90467	2085502218	1162	EURUSD	5.00	2008-08-06 15:32:08	0	1
98570	2085502819	1162	EURUSD	2.00	2008-09-03 03:04:27	0	1
99801	2085502930	1162	EURUSD	40.00	2008-09-05 11:23:11	0	1

In [75]:
```
agg_func_math = {
    'win_rate':
    ['sum', 'mean', 'median', 'min', 'max', 'std', 'var', 'mad', 'prod', '
}
stats1 = trader_symbol_grouping.groupby(['LOGIN', 'SYMBOL']).agg(agg_func_

stats1
```

Out[75]:

		w								
		sum	mean	median	min	max	std	var	mad	prod
LOGIN	**SYMBOL**									
	EURGBP	0.000000	0.000000	0.000000	0.000000	0.000000	NaN	NaN	0.0	0.000000
	EURUSD	0.641791	0.641791	0.641791	0.641791	0.641791	NaN	NaN	0.0	0.641791
1162	**GBPJPY**	1.000000	1.000000	1.000000	1.000000	1.000000	NaN	NaN	0.0	1.000000
	GBPUSD	0.787879	0.787879	0.787879	0.787879	0.787879	NaN	NaN	0.0	0.787879
	USDCHF	1.000000	1.000000	1.000000	1.000000	1.000000	NaN	NaN	0.0	1.000000
...	**...**	...	...	...	...	...	...	...	...	...
	NZDUSD	0.000000	0.000000	0.000000	0.000000	0.000000	NaN	NaN	0.0	0.000000
	USDCAD	0.250000	0.250000	0.250000	0.250000	0.250000	NaN	NaN	0.0	0.250000
6866968	**USDCHF**	0.500000	0.500000	0.500000	0.500000	0.500000	NaN	NaN	0.0	0.500000
	USDJPY	0.478261	0.478261	0.478261	0.478261	0.478261	NaN	NaN	0.0	0.478261
	XAUUSD	0.000000	0.000000	0.000000	0.000000	0.000000	NaN	NaN	0.0	0.000000

6603 rows × 10 columns

In [76]:
```
bad_traders1 = stats1[(stats1[('win_rate', 'sum')] < 0.5)]# & (stats1[('PR
good_traders1 = stats1[(stats1[('win_rate', 'sum')] >= 0.5)]# & (stats1[('
```

In [85]:
```
1 bad_traders1
```

Out[85]:

		w								
		sum	mean	median	min	max	std	var	mad	prod
LOGIN	SYMBOL									
1162	EURGBP	0.000000	0.000000	0.000000	0.000000	0.000000	NaN	NaN	0.0	0.000000
1396	AUDJPY	0.000000	0.000000	0.000000	0.000000	0.000000	NaN	NaN	0.0	0.000000
	EURUSD	0.487805	0.487805	0.487805	0.487805	0.487805	NaN	NaN	0.0	0.487805
22002	GBPUSD	0.434783	0.434783	0.434783	0.434783	0.434783	NaN	NaN	0.0	0.434783
	USDCHF	0.400000	0.400000	0.400000	0.400000	0.400000	NaN	NaN	0.0	0.400000
...	...	...	...	...	...	...	...	...	...	...
	GBPJPY	0.369231	0.369231	0.369231	0.369231	0.369231	NaN	NaN	0.0	0.369231
	NZDUSD	0.000000	0.000000	0.000000	0.000000	0.000000	NaN	NaN	0.0	0.000000
6866968	USDCAD	0.250000	0.250000	0.250000	0.250000	0.250000	NaN	NaN	0.0	0.250000
	USDJPY	0.478261	0.478261	0.478261	0.478261	0.478261	NaN	NaN	0.0	0.478261
	XAUUSD	0.000000	0.000000	0.000000	0.000000	0.000000	NaN	NaN	0.0	0.000000

2494 rows × 10 columns

In [86]:
```
1 good_traders1
```

Out[86]:

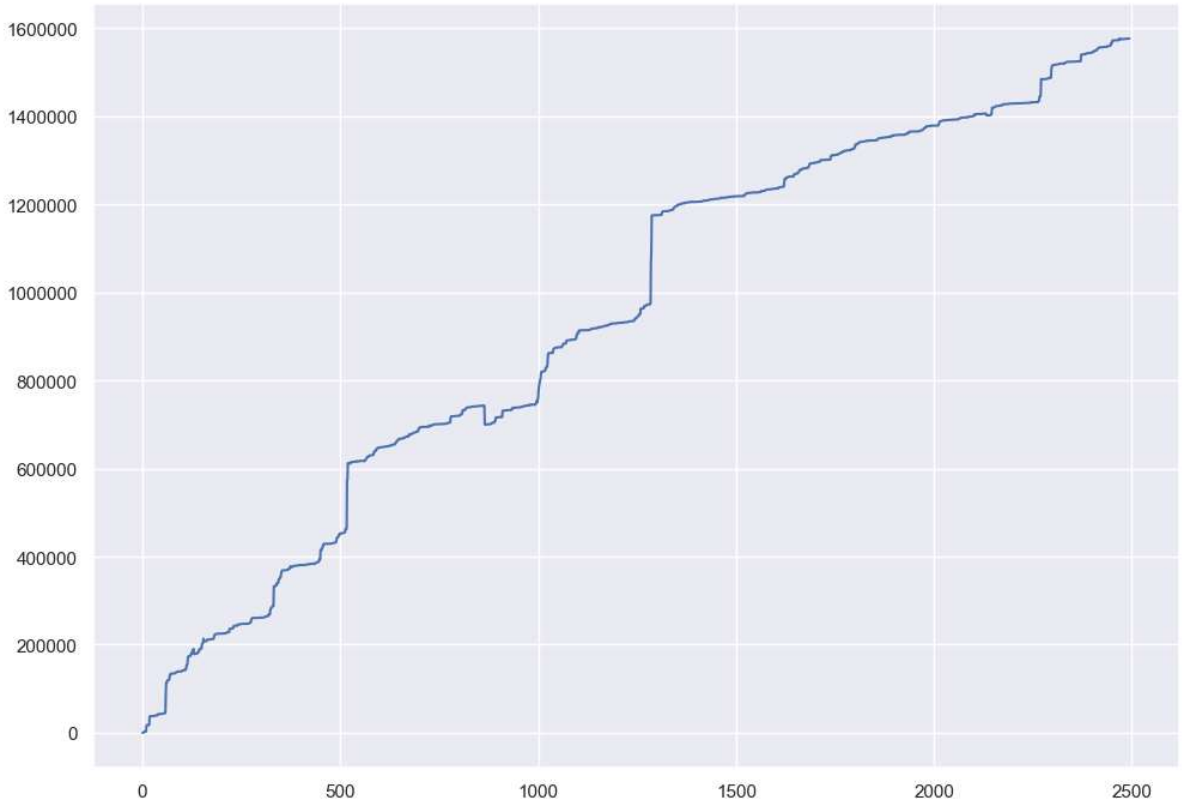
		w								
		sum	mean	median	min	max	std	var	mad	prod
LOGIN	SYMBOL									
	EURUSD	0.641791	0.641791	0.641791	0.641791	0.641791	NaN	NaN	0.0	0.641791
	GBPJPY	1.000000	1.000000	1.000000	1.000000	1.000000	NaN	NaN	0.0	1.000000
1162	GBPUSD	0.787879	0.787879	0.787879	0.787879	0.787879	NaN	NaN	0.0	0.787879
	USDCHF	1.000000	1.000000	1.000000	1.000000	1.000000	NaN	NaN	0.0	1.000000
	USDJPY	0.750000	0.750000	0.750000	0.750000	0.750000	NaN	NaN	0.0	0.750000
...	...	...	...	...	...	...	...	...	...	...
	AUDUSD	0.636364	0.636364	0.636364	0.636364	0.636364	NaN	NaN	0.0	0.636364
	CHFJPY	0.500000	0.500000	0.500000	0.500000	0.500000	NaN	NaN	0.0	0.500000
6866968	EURGBP	0.500000	0.500000	0.500000	0.500000	0.500000	NaN	NaN	0.0	0.500000
	GBPUSD	0.583333	0.583333	0.583333	0.583333	0.583333	NaN	NaN	0.0	0.583333
	USDCHF	0.500000	0.500000	0.500000	0.500000	0.500000	NaN	NaN	0.0	0.500000

4109 rows × 10 columns

In [88]:
```
badtraders1 = set()
for i, x in bad_traders1.iterrows():
    _id = f'{i[0]}/{i[1]}'
    badtraders1.add(_id)

badtraders1
```

Out[88]:
```
{'123248/USDJPY',
 '518093/AUDJPY',
 '53863/NZDUSD',
 '72452/NZDUSD',
 '523784/EURUSD',
 '79873/USDCAD',
 '554463/AUDUSD',
 '567376/GBPUSD',
 '1162/EURGBP',
 '517057/CADJPY',
 '6866968/XAUUSD',
 '63885/USDCHF',
 '54084/EURUSD',
 '562329/EURCHF',
 '556628/AUDJPY',
 '68305/AUDUSD',
 '520817/AUDCAD',
 '562202/USDCHF',
 '63073/CADJPY',
```

In [92]:
```
equity = 0
history = []
history.append(equity)

for i, x in trader_symbol_grouping.iterrows():
    _id = f"{x['LOGIN']}/{x['SYMBOL']}"
    if _id in badtraders1:
        equity += (-1 * x['PROFIT'])
        history.append(equity)

fig, ax = plt.subplots()
ax.plot(history)
ax.ticklabel_format(style='plain')
plt.show()
```

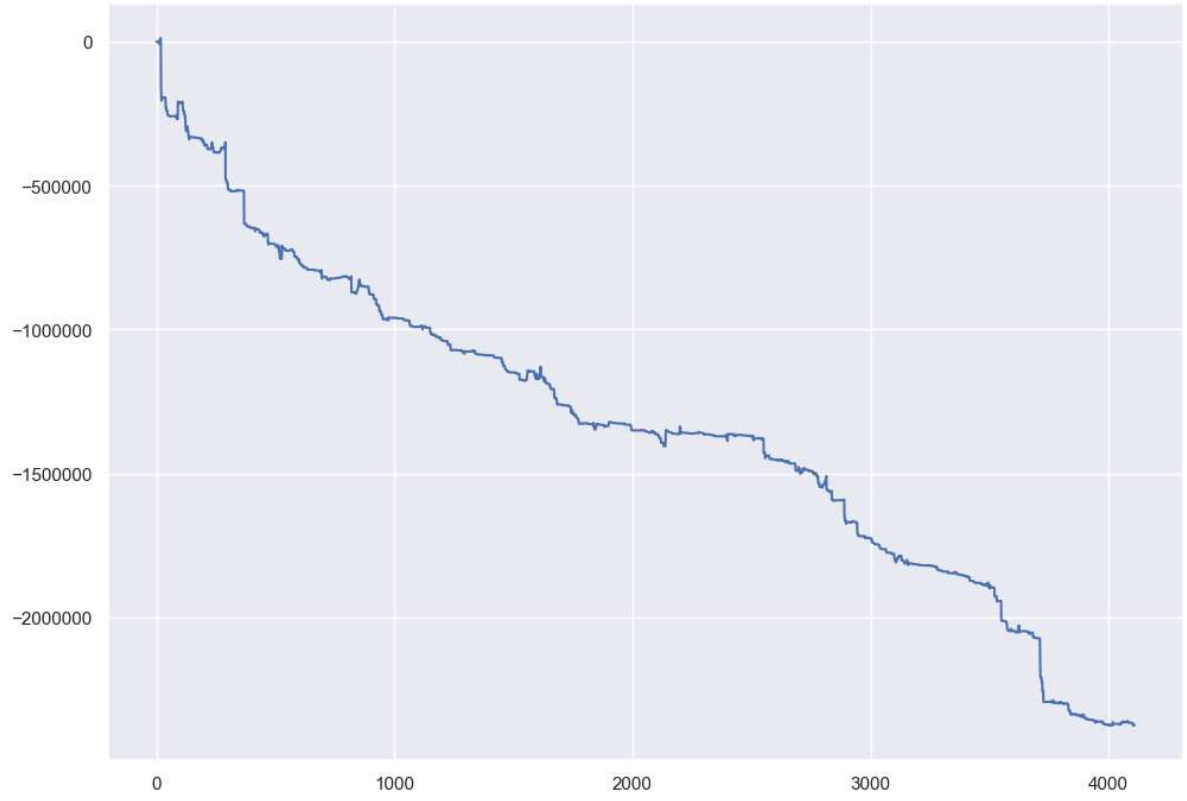


In [79]:
```
(len(history) / len(test)) * 100
```

Out[79]: 2.2858451672010993

In [80]:
```
goodtraders1 = set()
for i, x in good_traders1.iterrows():
    _id = f'{i[0]}/{i[1]}'
    goodtraders1.add(_id)
```

In [95]:
```
equity = 0
history = []
history.append(equity)

for i, x in trader_symbol_grouping.iterrows():
    _id = f"{x['LOGIN']}/{x['SYMBOL']}"
    if _id in goodtraders1:
        equity += (1 * x['PROFIT'])
        history.append(equity)
fig, ax = plt.subplots()
ax.plot(history)
ax.ticklabel_format(style='plain')
plt.show()
```



In [82]:
```
(len(history) / len(test)) * 100
```

Out[82]: 3.7654603756298672

In [ ]:
```

```