

# June 28 Update

- Portfolio Network Generation—3 different methods
- Price Update with Hurst Index
- Thresholding with Sigmoid Function
- Data format and analysis

# Rolling method

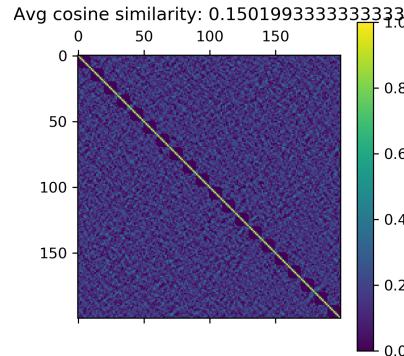
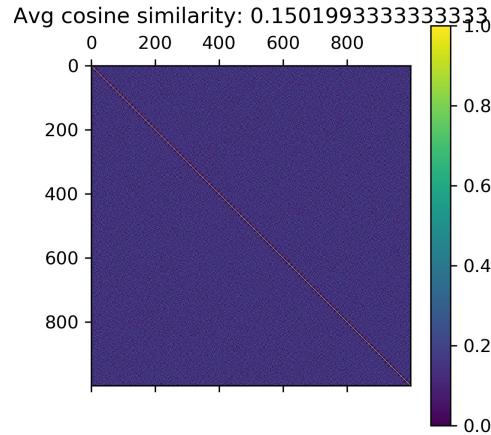
5 overlap in size 15 portfolios

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14]  
[10 11 12 13 14 15 16 17 18 19 20 21 22 23 24]  
[20 21 22 23 24 25 26 27 28 29 30 31 32 33 34]  
[30 31 32 33 34 35 36 37 38 39 40 41 42 43 44]  
[40 41 42 43 44 45 46 47 48 49 0 1 2 3 4] _____ shuffle  
[ 3  8 41  1 32  7  6 16 42 44 36 21 48 12 24]  
[36 21 48 12 24 49 39 46 45 13 35 18 31 33 28]  
[35 18 31 33 28 17 26 37 25 14 4 0 11 38 23]  
[ 4  0 11 38 23 34 43 29 15 10 20 9 27 30 47]  
[20  9 27 30 47 22 40 5 19 2 3 8 41 1 32] _____ shuffle  
[ 3  1 37 16 13 19 5 48 14 0 21 15 27 20 24]  
[21 15 27 20 24 44 38 18 29 6 9 34 2 39 42]
```

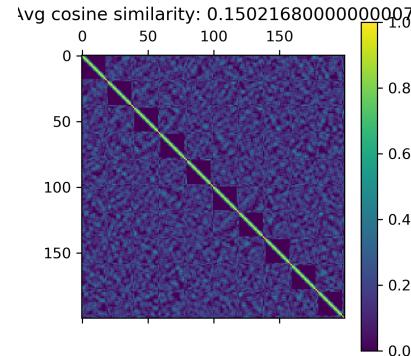
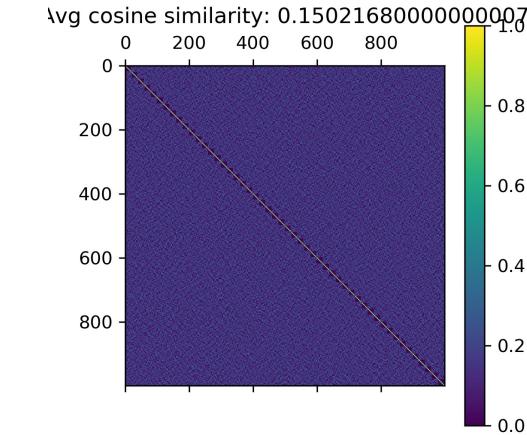
# Rolling Method cosine similarities

1000 portfolios of size 15

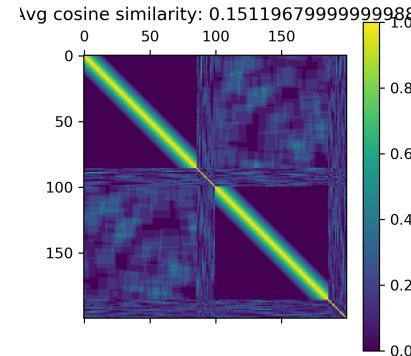
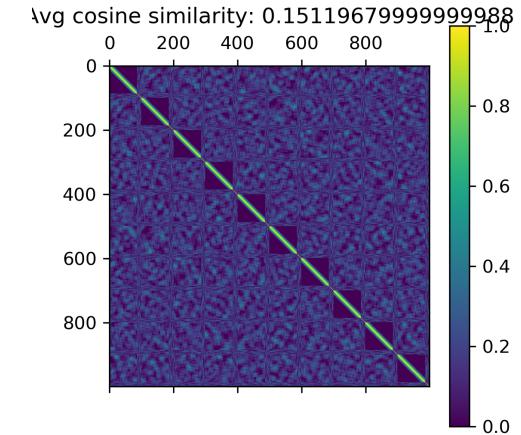
Overlap window size: 5



Overlap window size: 10



Overlap window size: 14



# Rolling with random choice

5 overlap in size 15 portfolios

[ 7 45 4 29 1 10 14 26 15 31 49 39 36 19 37]
[49 39 36 19 37 6 3 38 35 41 30 21 12 42 25]
[30 21 12 42 25 0 32 1 24 5 7 2 9 31 44]
[ 7 2 9 31 44 28 37 48 40 4 26 11 13 39 49]
[26 11 13 39 49 47 14 19 8 22 27 24 46 15 12]
[27 24 46 15 12 37 44 25 21 48 31 2 33 36 5]
[31 2 33 36 5 6 11 49 18 28 13 35 43 26 9]
[13 35 43 26 9 29 23 47 40 41 32 19 0 30 42]
[32 19 0 30 42 12 1 14 3 4 16 49 21 46 36]
[16 49 21 46 36 23 17 39 13 34 8 6 5 25 43]
[ 8 6 5 25 43 10 12 41 48 27 28 38 14 37 42]

First 5 overlap last 5 of  
previous portfolio

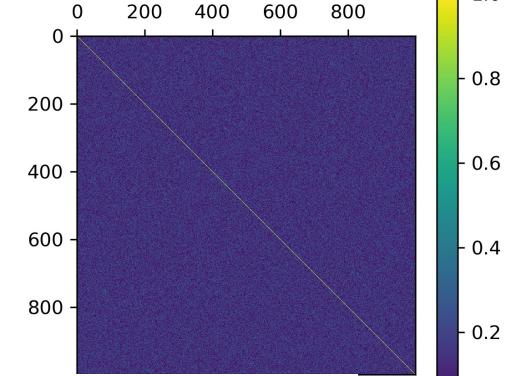
Last 10 chosen randomly  
from remaining stocks

# Rolling w/ Random cosine similarities

1000 portfolios of size 15

Overlap window size: 5

Avg cosine similarity: 0.15109320000000007

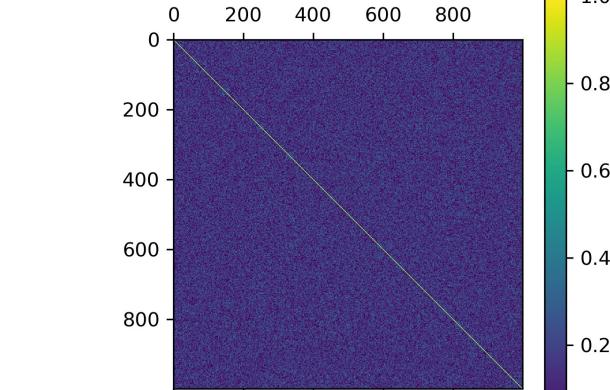


Avg cosine similarity: 0.15109320000000007

Overlap window size: 10

Overlap window size: 10

Avg cosine similarity: 0.1523554666666658

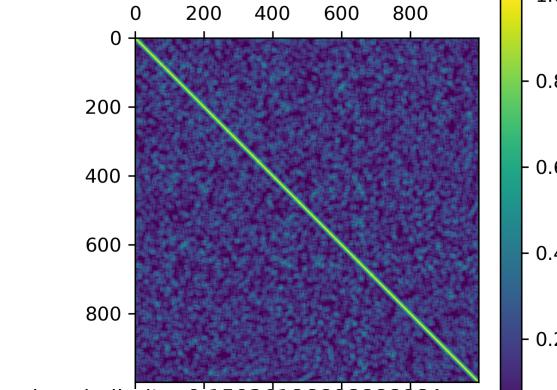


Avg cosine similarity: 0.1523554666666658

Overlap window size: 14

Overlap window size: 14

Avg cosine similarity: 0.1592618666666664

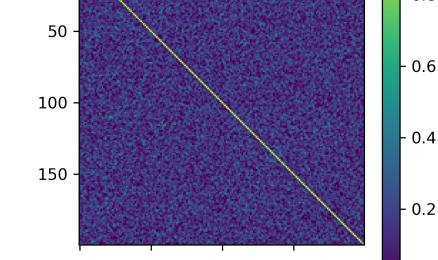


Avg cosine similarity: 0.1592618666666664

Overlap window size: 14

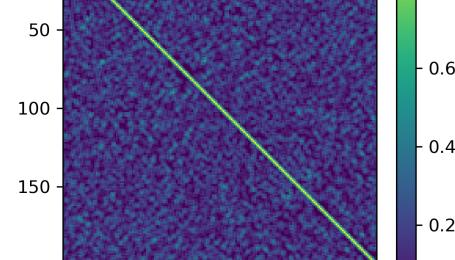
Overlap window size: 5

Avg cosine similarity: 0.15109320000000007



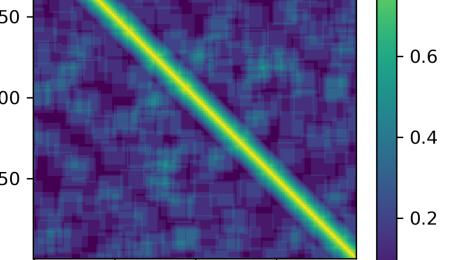
Overlap window size: 10

Avg cosine similarity: 0.1523554666666658



Overlap window size: 14

Avg cosine similarity: 0.1592618666666664



# Random method

5 overlap in size 15 portfolios

[	4	29	76	80	71	72	45	62	82	2	78	63	40	88	13]
[	76	13	4	72	63	47	20	95	16	36	19	61	54	17	90]
[	90	17	54	61	16	86	12	28	78	45	94	3	71	69	52]
[	71	86	90	52	54	59	99	35	79	6	58	41	5	24	37]
[	90	37	79	5	54	11	49	2	23	19	32	67	63	78	70]
[	11	37	63	32	23	53	9	98	51	29	58	84	68	40	18]
[	58	29	32	11	40	45	2	83	42	38	36	72	71	3	75]
[	83	32	3	40	11	77	82	80	79	7	95	65	24	81	9]

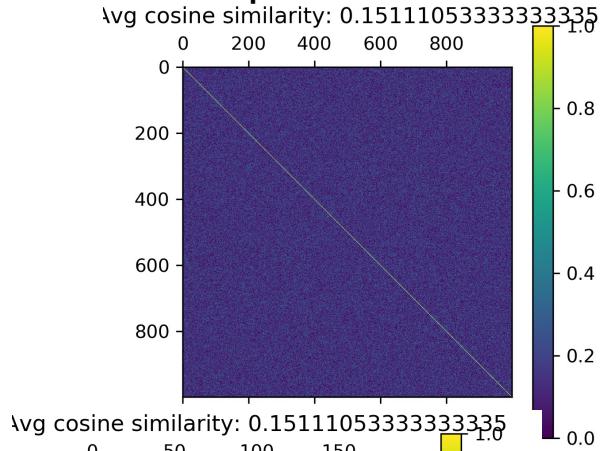
First 5 overlap chose  
randomly from previous  
portfolio

Last 10 chosen randomly  
from remaining stocks

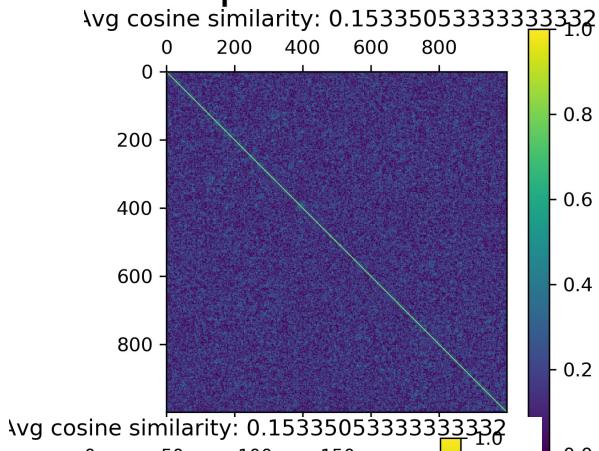
# Rolling w/ Random cosine similarities

1000 portfolios of size 15

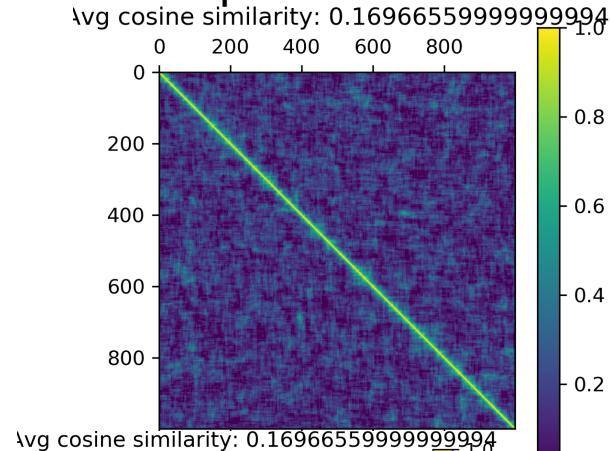
Overlap window size: 5



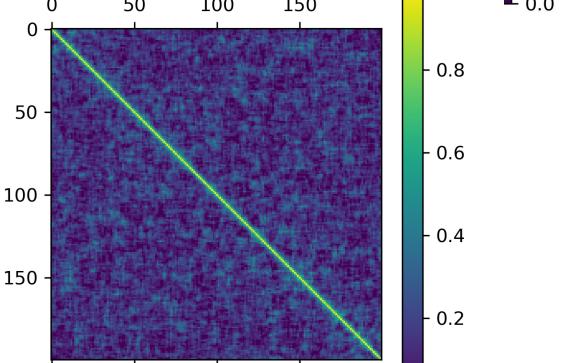
Overlap window size: 10



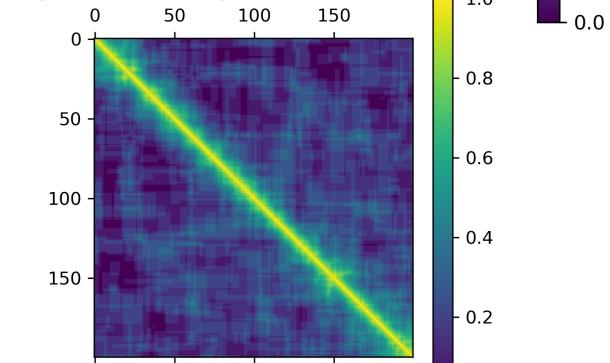
Overlap window size: 14



Avg cosine similarity: 0.1533505333333332



Avg cosine similarity: 0.1696655999999994



# Price Update

```
def priceChange(stock,time, sell=True, increment = 0.001):
    """
    Updates the stockPool and hurstPool as price change
    """
    global stockPool, hurstPool

    h0 = hurstPool[stock][time]
    numberNewPrices = len(stockPool[stock][time:])
    p0 = stockPool[stock][time]

    if sell:
        h1 = h0-increment
        if h1<0.4:
            h1=0.4
        hurstPool[stock][time:] = h1
    else:
        h1 = h0+increment
        if h1>0.8:
            h1=0.8
        hurstPool[stock][time:] = h1

    ffbmNew = ffbm(h1,2**14,2**14)
    ffbmNew = abs(ffbmNew[:numberNewPrices]+p0)
    print('stock ', stock, ' original H', h0, ' to ', h1)
    stockPool[stock][time:]=ffbmNew
```

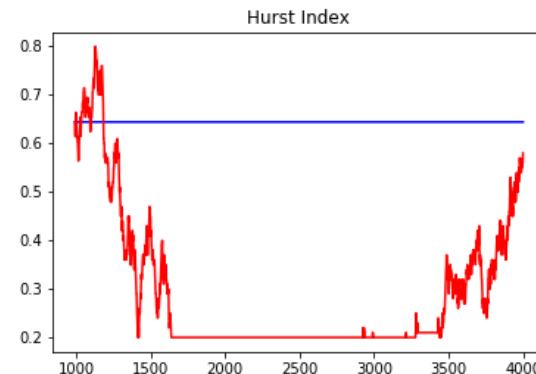
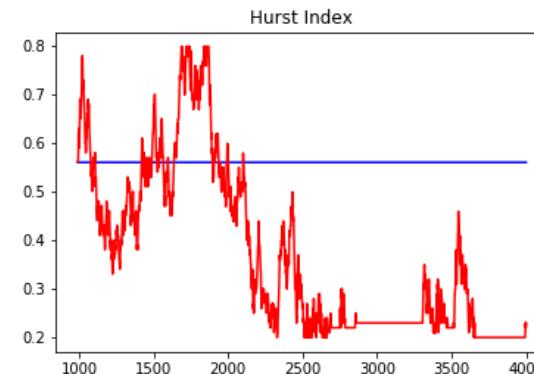
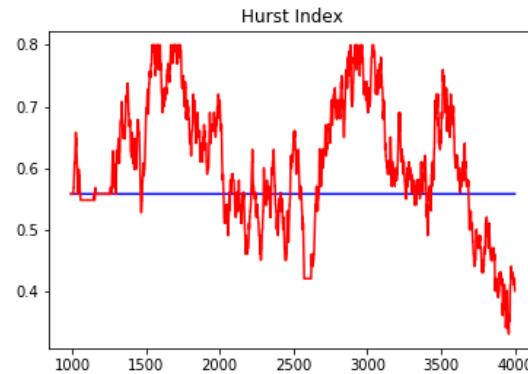
If selling a stock, decrease H index

If buying, increase H index

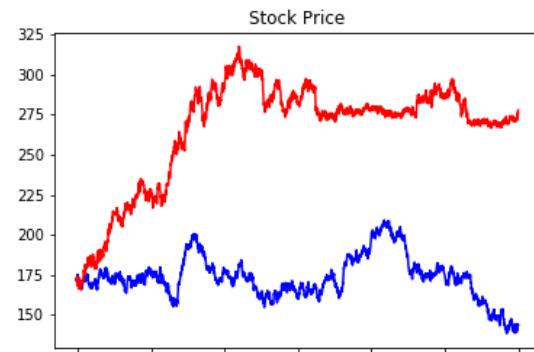
Generates new fBM path and replaces the remaining stock prices from that point forward

Absolute value of the new stock prices

# Increment of 0.01



Stock 4



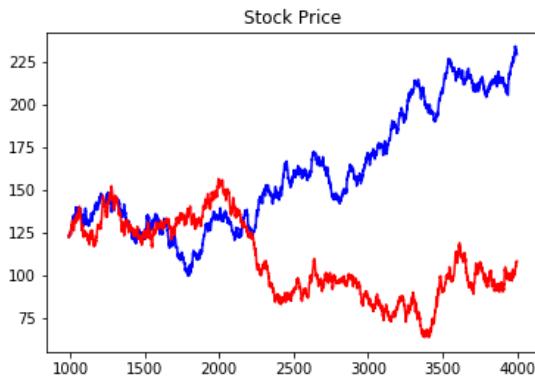
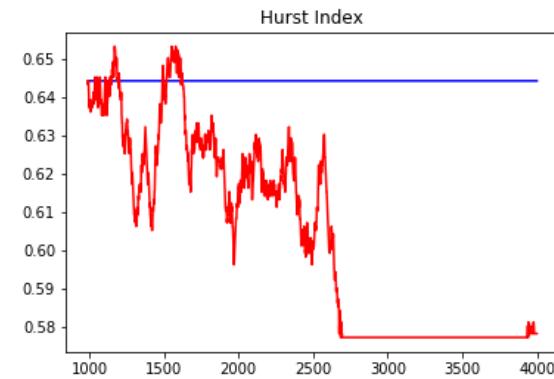
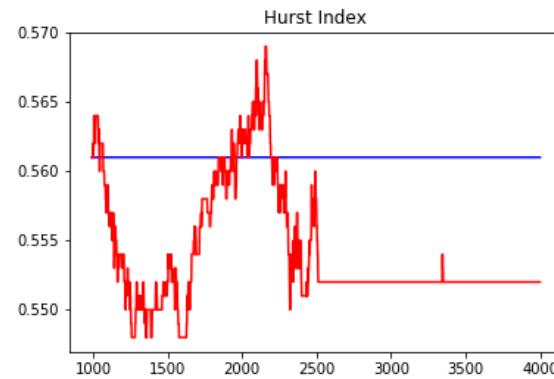
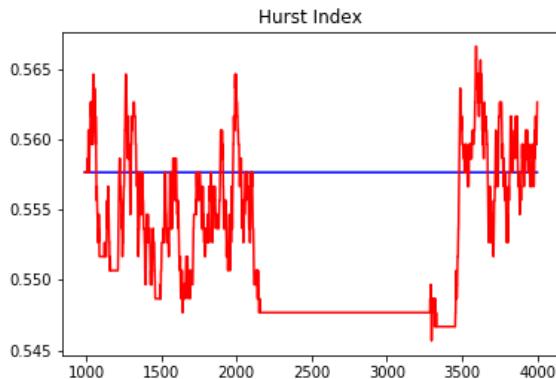
Changing Index Original Index



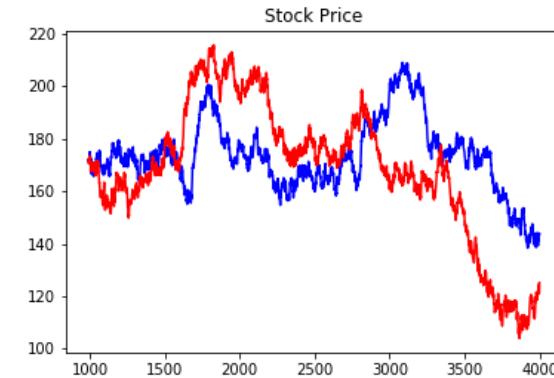
Stock 2

Hurst indices (top) and corresponding stock price (bottom)

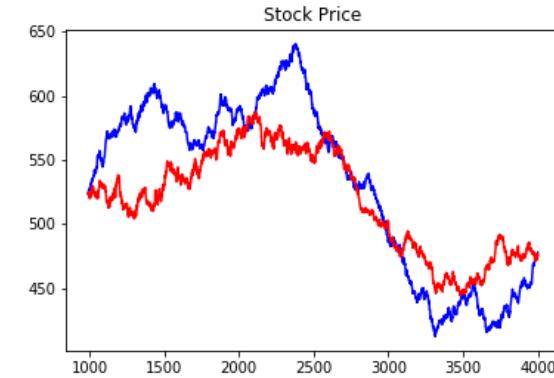
# Increment of 0.001



Stock 4



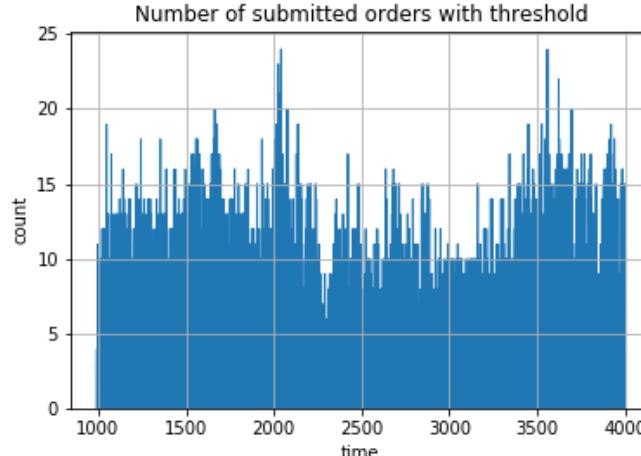
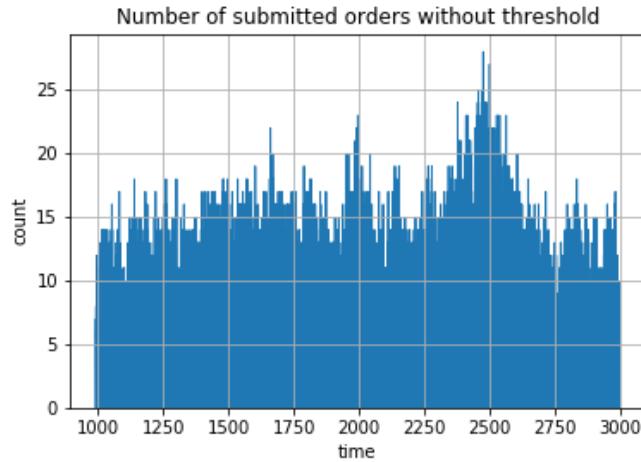
Changing Index Original Index



Stock 2

Hurst indices (top) and corresponding stock price (bottom)

# Thresholding with Sigmoid Function



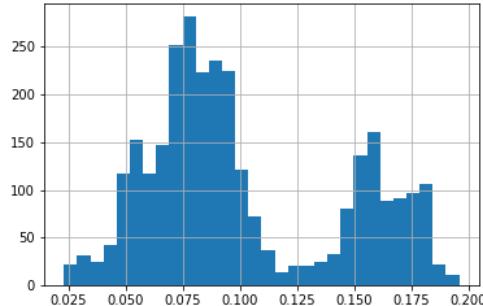
1<sup>st</sup> run: 100% rebalance probability each turn

- compile all the Sharpe ratios at each point in time of each portfolio
- Take a top percentile of these ratios for sigmoid function

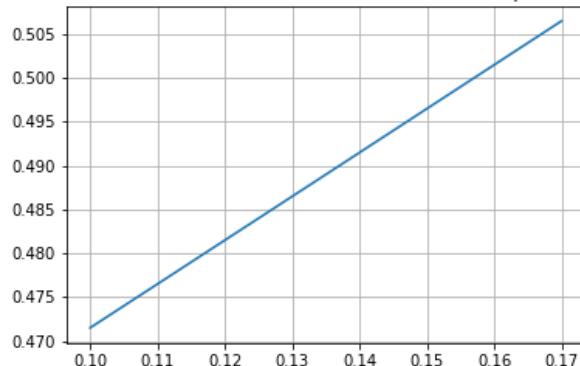
2<sup>nd</sup> run: varying probability based on sigmoid function

# Sigmoid functions

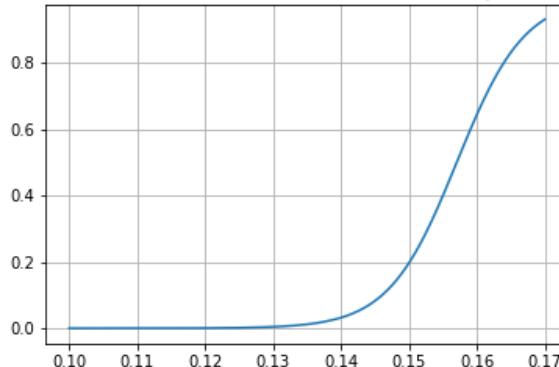
Distribution of all sharpe ratios in example portfolio before thresholding



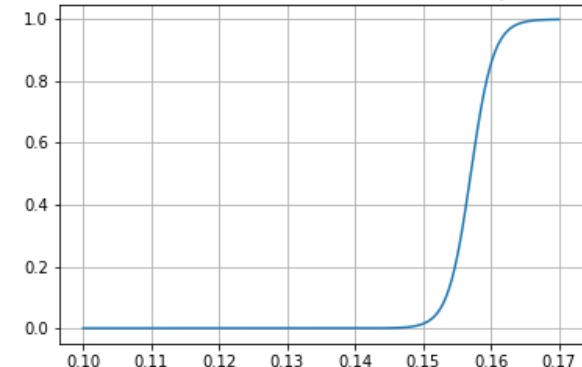
85th Ptile (0.15705378424399136) with k slope: 2



85th Ptile (0.15705378424399136) with k slope: 200



85th Ptile (0.15705378424399136) with k slope: 600



Small slopes hover around 50% at all ratios

# Data tables

	time	portfolio	stock	order
0	993	jkitd	2	-1.0
1	993	jkitd	0	3.0
2	993	jkitd	18	-1.0
3	994	gdwoa	8	1.0
4	994	jkitd	2	1.0
...	...	...	...	...
19375	3998	jkitd	14	-1.0
19376	3999	acdjr	6	1.0
19377	3999	acdjr	15	-1.0
19378	3999	acdjr	14	-1.0
19379	3999	acdjr	9	1.0

19380 rows × 4 columns

## Transactions list

- Volumes are very small, should increase once I use a steeper sigmoid function or increase portfolio total volumes

	time	stock	mean	var	std
0	992	0	0.089316	0.502124	0.708607
0	992	1	-0.027688	0.155857	0.394788
0	992	2	0.017721	0.049307	0.222053
0	992	3	-0.032374	0.190581	0.436555
0	992	4	-0.002152	0.644203	0.802623
...	...	...	...	...	...
0	3999	15	0.017759	0.060449	0.245863
0	3999	16	0.011629	0.189428	0.435234
0	3999	17	-0.092593	0.463182	0.680575
0	3999	18	0.010118	0.511986	0.715532
0	3999	19	0.017805	0.261123	0.511002

## Stock Characteristics

- Characteristics of each stock at each time point taken based on the window of points beforehand

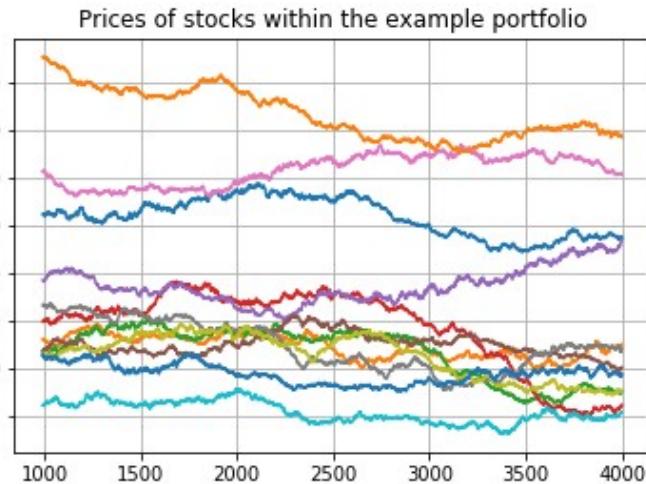
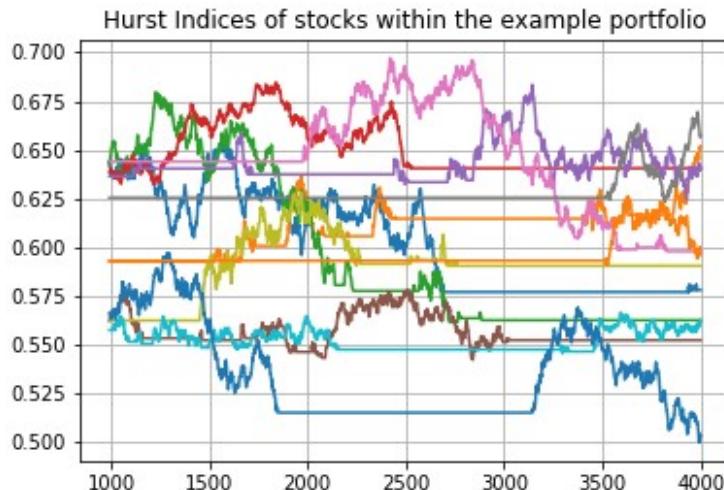
	ID	time	stock	weight
0	edgla	992	12	18.0
0	edgla	992	17	27.0
0	edgla	992	6	37.0
0	edgla	992	3	0.0
0	edgla	992	7	0.0
...	...	...	...	...
0	edgla	1023	8	49.0
0	edgla	1023	2	67.0
0	edgla	1024	12	8.0
0	edgla	1024	17	27.0
0	edgla	1024	6	36.0

## Portfolio weights

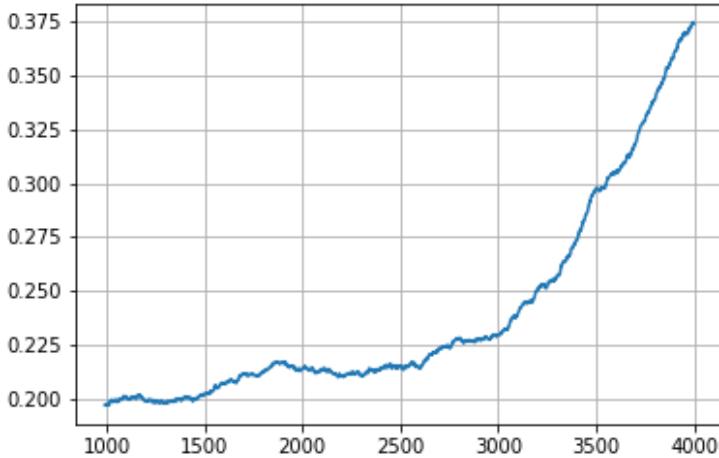
- Each portfolio has a table tracking its weights over time

# Example Portfolio

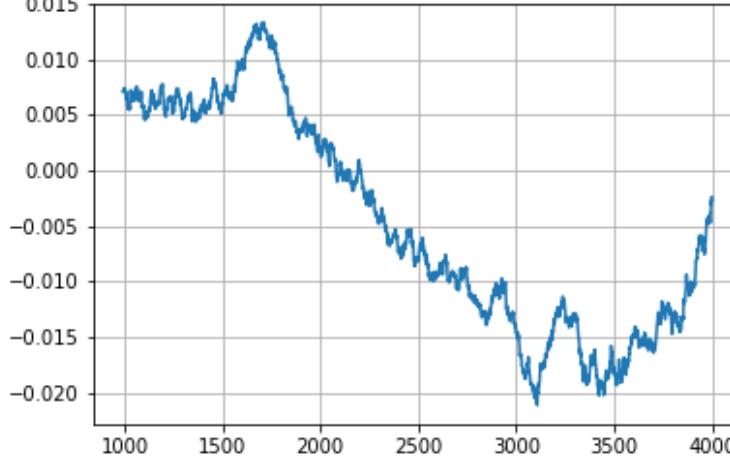
- Contains 12 stocks



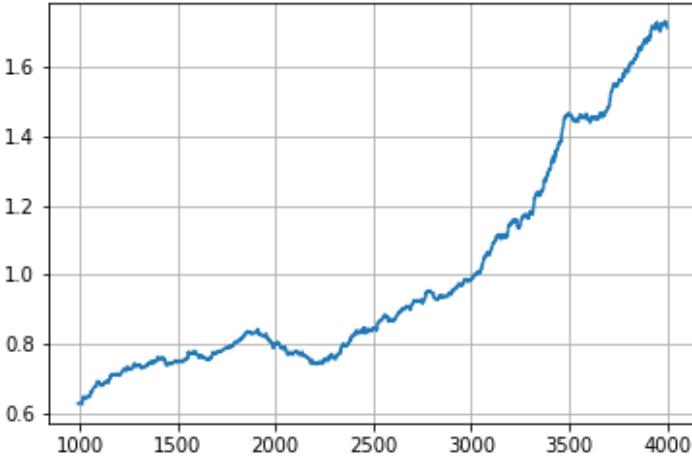
Variance of expected percent returns of an example portfolio



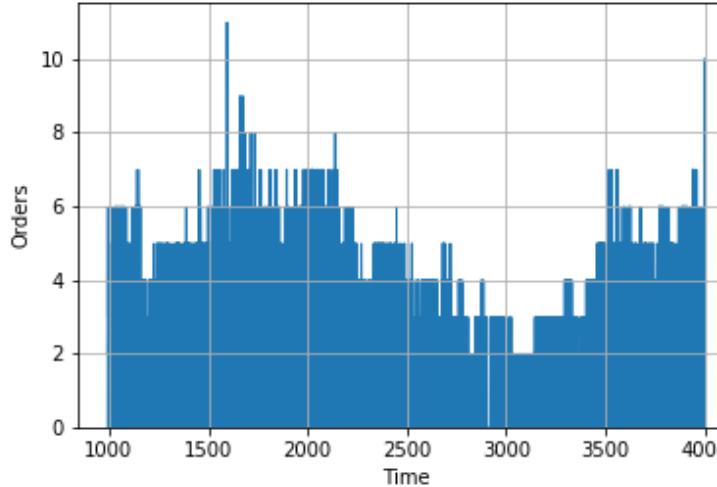
Average expected percent returns of an example portfolio



Variance gap of returns of an example portfolio



Number of submitted orders of an example portfolio



Haven't accounted for weights in these figures