Cascades

1. Select one portfolio as the cascade seed.
2. Add all activity of the seed portfolio until a period of quiescence to the cascade.
3. Add all stocks traded by the seed portfolio in this period to the cascade stock pool.
4. All other portfolios that hold a stock in the cascade stock pool are added to the cascade, and their activity until a period of quiescence is added to the cascade.
5. All new stocks traded by newly added portfolios are also added to the cascade stock pool which can reach previously non-overlapping portfolios.
6. Process repeats until all portfolio activity in the cascade reaches a period of quiescence.
7. Entire process repeats to create a new cascade.

This cascade stock pool links all the portfolios involved in the cascade, either directly overlapping or as “3rd stocks” like Siew Ann mentioned. However, since this pool is added to as I iterate over the portfolios, iterating over the portfolios a second time tends to result in all portfolios being added to the cascade. Maybe I’m missing a max time seperation for a stock to influence another portfolio?

Currently, portfolios can only undergo one period of activity per cascade and that activity is unique to the cascade it is classified into.

Cascade Algorithm

```
cascades = {}
numCascade = 0
cascadeStocks = {} - All stocks involved in the cascades
cascadePortfs = {} - All portfolios involved in the cascades
cascadeTime = {}   - The starting times of the cascades
cascadeTf = {}     - The final times of the cascades
cascades = {}      - The transactions involved in the cascades
while len(TtotalOrders) > 0:
#     totalTimeOrders = TtotalOrders[TtotalOrders['time']==t]
    # seed with first portfolio
    print("OG Orders left: ", len(TtotalOrders))
1.  seed = (TtotalOrders.iloc[0]['portfolio'])
    t0 = TtotalOrders.iloc[0]['time']
    seedCascade = findPortfOrderCascades(TtotalOrders, seed, t0, maxSep = 1)
    cascadeStocks[numCascade] = np.asarray(seedCascade['stock'])
    cascadePortfs[numCascade] = np.asarray(seed)
    cascadeTime[numCascade] = t0
    cascades[numCascade] = seedCascade
    assert len(seedCascade) > 0
    TtotalOrders = TtotalOrders[~TtotalOrders.isin(seedCascade)].dropna()
    # NEW VERSION (PORTFOLIO PERSPECTIVE)
    for childPortf in list(TtotalOrders['portfolio'].unique()):
        if any(np.isin(cascadeStocks[numCascade],traderIDs[childPortf].stocks)):
#           print("match: ", childPortf)
2.          if childPortf not in cascadePortfs[numCascade]:
                cascadePortfs[numCascade] = np.append(cascadePortfs[numCascade], childPortf)
                childPortfCascade = findPortfOrderCascades(TtotalOrders,childPortf,t0, maxSep = 1)
                cascades[numCascade] = pd.concat([cascades[numCascade],childPortfCascade])
                TtotalOrders = TtotalOrders[~TtotalOrders.isin(childPortfCascade)].dropna()
                for childStock in list(childPortfCascade['stock'].unique()):
                    if childStock not in cascadeStocks[numCascade]:
                        cascadeStocks[numCascade] = np.append(cascadeStocks[numCascade], childStock)
    print("Cascade length: ",len(cascades[numCascade]))
    print("New Orders left: ", len(TtotalOrders))
3.  cascadeTf[numCascade] = cascades[numCascade]['time'].max()
    print("making new cascade")
    numCascade += 1
```

1. Take the first row of the orders list as the cascade seed

	time	portfolio	stock	order
0	993	taukg	6	-1.0

Identify seedCascade as all portoflio activity that occurs at or after t0 and terminates after maxSep time duration of quiescence

Add all associated stocks in the seedCascade to cascadeStocks dictionary associated with the cascade ID

Add the seed portfolio to CascadePortfs dictionary associated with the cascade ID

Move all rows associated with seedCascade to cascades dictionary

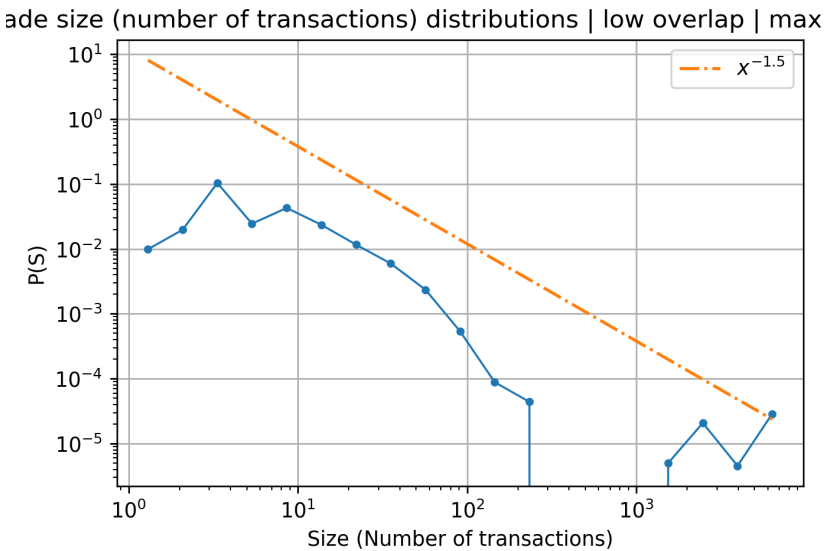
2. Go through each portfolio and see if it holds any stocks traded in this cascade

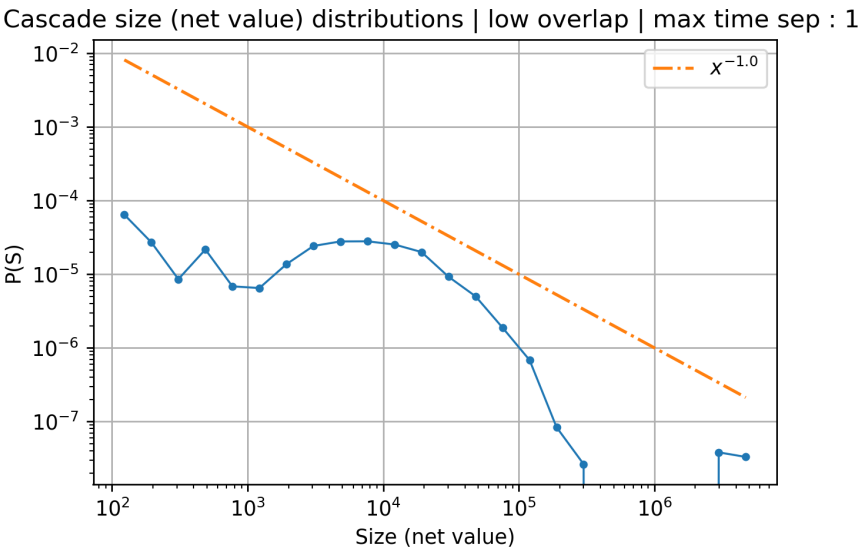
If match is found, above steps are repeated with this portfolio

3. Identify the cascade end time, and extend number of cascades

Case 1

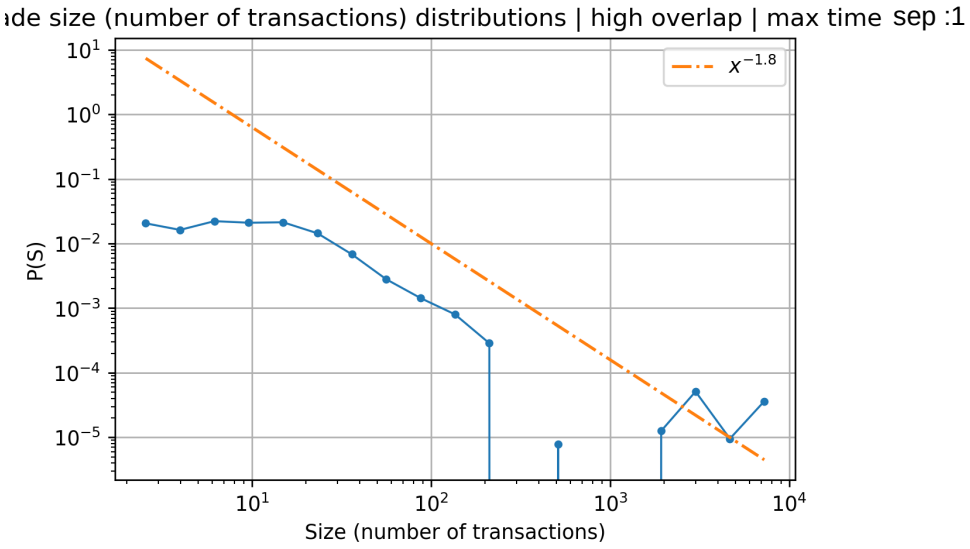
500 portfolios 1000 time steps
840 cascades

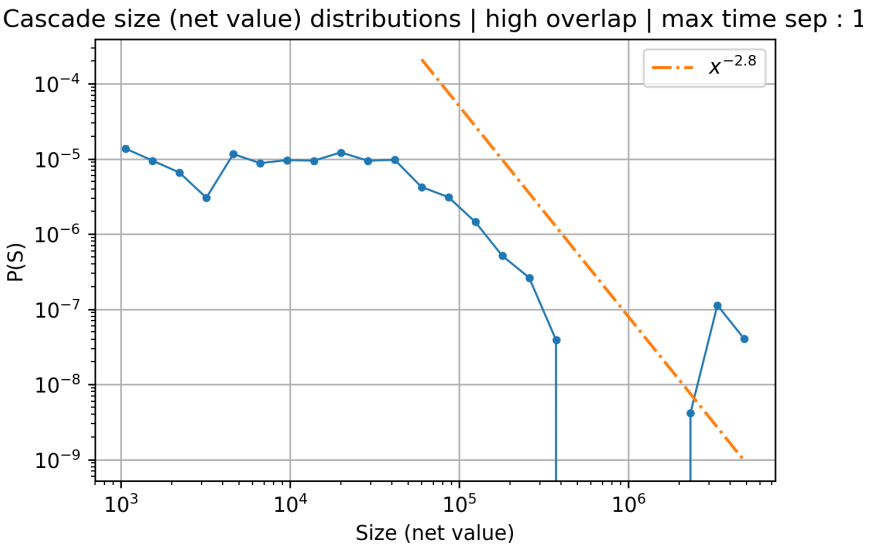




Case 2

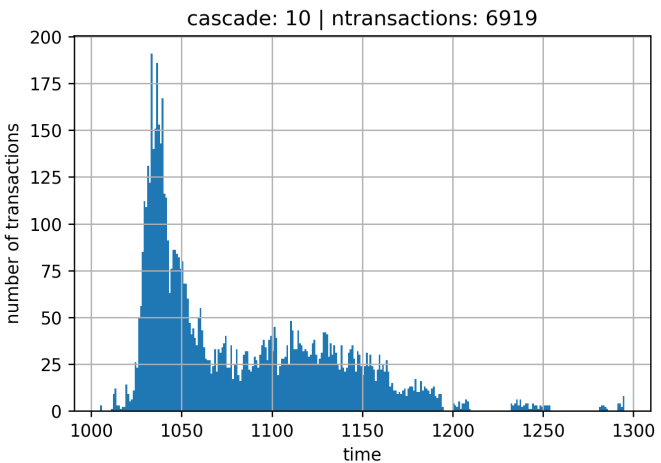
500 portfolios 1000 time steps
560 cascades



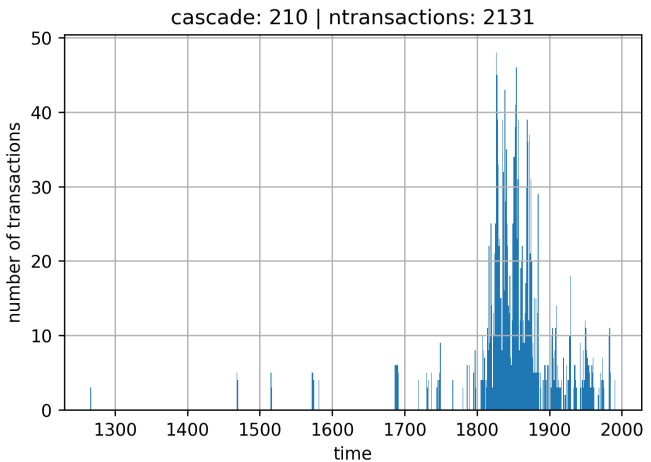


Cascade Shapes

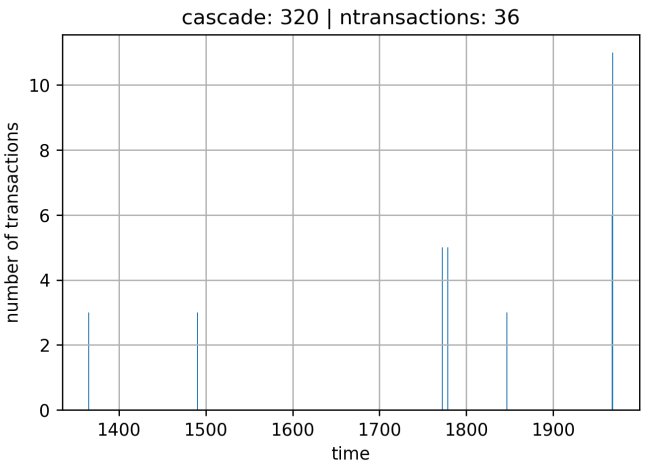
Large Cascade



Medium Cascade



Small Cascade



Adding a max time seperation for a stock to influence other portfolios could prevent spread out cascades like these