Modelling Asset Returns with Ehrenfest Urns

Cass Bivens, Terry Cheng, Kankshat Patel, Michael Sack, Vance Sine

12/14/2021

Introduction

Empirical observations have detected excess kurtosis in stock returns, suggesting that stock prices in reality are more volatile than in traditional models. We speculate these bursts in volatility come from human tendencies to follow the investment patterns of much of the public, or herd. These tendencies lead us to hypothesize that human psychological tendencies, unrelated to the influx of viable information, drive stock price distributions to be characterized by fatter tails and excess kurtosis. In order to explain some of these phenomena observed in the empirical data, we create a modified Ehrenfest urn to model asset prices in the presence of herd behavior tendencies.

The problems in this field, as discussed above, motivate us to perform a detailed study of the impact of herd-like tendencies on price distributions by the following approaches:

Most current theoretical models for showing stock prices to be normally distributed, however, Empirical data of the S&P 500 suggests that distributions of financial returns are fat-tailed with excess kurtosis (Cont (2001)). Many believe this to be a result of large bursts of volatility due to human tendencies to follow the pack observed in the market.

Ehrenfest Urn Model

The Ehrenfest urn model is one of the most instructive and popular models in all of probability, stochastic processing, and physical statistics. The basic Ehrenfest urn model consists of 2 urns and N balls, with n balls in the first urn and $N-n$ balls in the second. At each discrete time step t, a natural number from 1 to N is chosen at random and the corresponding ball moves from one urn to the other. The discrete random process X_t tracks the number of balls in the first urn at time t, which is all you need to describe the state of the system at a given time (as the number of balls in the second urn is simply $N - X_t$). A more generalized model allows the selected ball to change urns with a probability that depends on the number of balls present in the urns.

Cont and Bouchaud (2000) apply this generalized model to simulate the number of bull and bear traders for some asset:

Let X_t be the number of bulls at time t, with transition probabilities

$$
\begin{aligned}
X_{n,n-1} &= \frac{n}{N} * \frac{\beta + N - n}{\alpha + \beta + N - 1} && (1)\\
X_{n,n} &= \frac{n}{N} * \frac{\alpha + N - n - 1}{\alpha + \beta + N - 1} + \frac{N-n}{N} * \frac{\beta + N - n - 1}{\alpha + \beta + N - 1} && (2)\\
X_{n,n+1} &= \frac{N-n}{N} * \frac{\alpha + N - n}{\alpha + \beta + N - 1} && (3)
\end{aligned}
$$

where n is the number of bull traders, N is the total number of traders, and α, β are parameters representing market preference.

```
def Path(N, T, a, b):
    bull = sps.betabinom.rvs(N,a,b)
    bear = N-bull
    Bull_Path = np.zeros([T])
    Bear_Path = np.zeros([T])
    Bull_Path[0] = bull
    Bear_Path[0] = bear
    total = bear + bull
    for i in range(1,T):
        p1 = (bull/total)*((b + bear)/(a + b + total - 1))
        p2 = (bull/total)*((a + bull - 1)/(a + b + total - 1)) + (bear/total)*((b + bear - 1)/(a + b +
        p3 = (bear/total)*((a + bull)/(a + b + total - 1))
        x = np.random.choice(3, p = [p1,p2,p3])
        if x == 0:
            bull = bull - 1
            bear = bear + 1
        if x == 1:
            bull = bull
            bear = bear
        if x == 2:
            bull = bull + 1
            bear = bear - 1

        Bull_Path[i] = bull
        Bear_Path[i] = bear
    return Bull_Path
```

```
def Simulator(N,T,a,b,M):
    data = np.zeros([M,T])
    for i in range(M):
        case = Path(N,T,a,b)
        for j in range(T):
            data[i][j] = case[j]
    return data
```

```
pickle.dump(Simulator(100,9000,.8,.2,10000),open( "sim_1.p", "wb" ))
pickle.dump(Simulator(100,9000,4000,6000,10000),open( "sim_2.p", "wb" ))
pickle.dump(Simulator(100,9000,100,100,10000),open( "sim_3.p", "wb" ))
```

Parameters α and β

Within our model, we defined two parameters, α and β, that represent market preferences. These parameters control the market preference for bulls and bears and dictate the sensitivity of the process to n.

We describe α as the parameter controlling the bull preference, since increasing α relative to β increases the probability that a selected bear becomes a bull. Likewise, we describe β as the parameter controlling the bear preference, since increasing β relative to α increases the probability that a selected bull becomes a bear.

In addition, the overall magnitude of α and β control the transition probabilities sensitivity to n, the current amount of bull traders. Essentially, if the sum of α and β is relatively small compared to N, traders have an increased tendency to move towards the urn with more of his fellow traders. The inverse is true as well: when the sum of α and β is relatively large compared to N, traders have an decreased tendency to follow the herd.

Microstates

To support the definitions above, we define three cases in a manner inspired by Cont and Bouchaud (2000), then simulate 10,000 microstates of each case and make qualitative assessments of the behavior.

Since we want to analyze how the parameters affect long-run behavior, we start all simulations in the stationary distribution. Cont and Bouchaud (2000) lists this as the beta-binomial distribution with parameters N, α, β.

Note: For each case, three random microstates are shown.

1. Bull preference, high sensitivity ($0 < \beta < \alpha << N$)

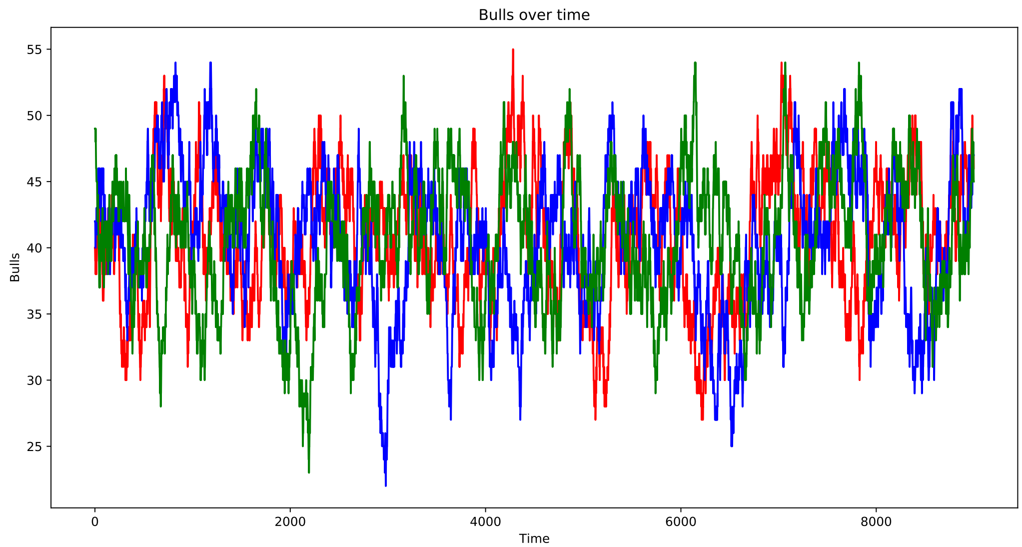
This case is characterized by a long stretches of time spent around the barriers, with rapid fluctuations across the central region. This is consistent with our claim that small α, β values lead to stronger herd behavior. The economy will often reach the all-bull (100) or all-bear (0) states. Occasionally, a few traders shift to the group containing less individuals, but quickly return to the original edge. Less often, the market will shift, with the traders flipping from one state to the other in a short number of steps. Overall, the economy will rarely stay evenly split, and whichever group has more traders tends to draw in the rest.

```
# Case 1: 0 < b < a < 1:

N = 100
T = 9000
a = 0.8
b = 0.2

x = np.arange(0,T)
y1 = pickle.load( open( "sim_1.p", "rb" ))[0]
y2 = pickle.load( open( "sim_1.p", "rb" ))[1]
y3 = pickle.load( open( "sim_1.p", "rb" ))[2]

plt.figure(figsize=(14,7))
plt.title("Bulls over time")
plt.xlabel('Time')
plt.ylabel('Bulls')
plt.plot(x, y1, color="red")
plt.plot(x, y2, color="blue")
plt.plot(x, y3, color="green")

plt.show()
```

2. Bear preference, low sensitivity ($N << \alpha < \beta$)

In this case, the process behaves similarly to the basic Ehrenfest urn model, with little time spent around the barriers and long fluctuations across the central region. Since $\alpha < \beta$, there should be a slight preference for bear traders. However, the economy is fairly evenly distributed, as they are less sensitive to the behavior of the herd than in the first model.

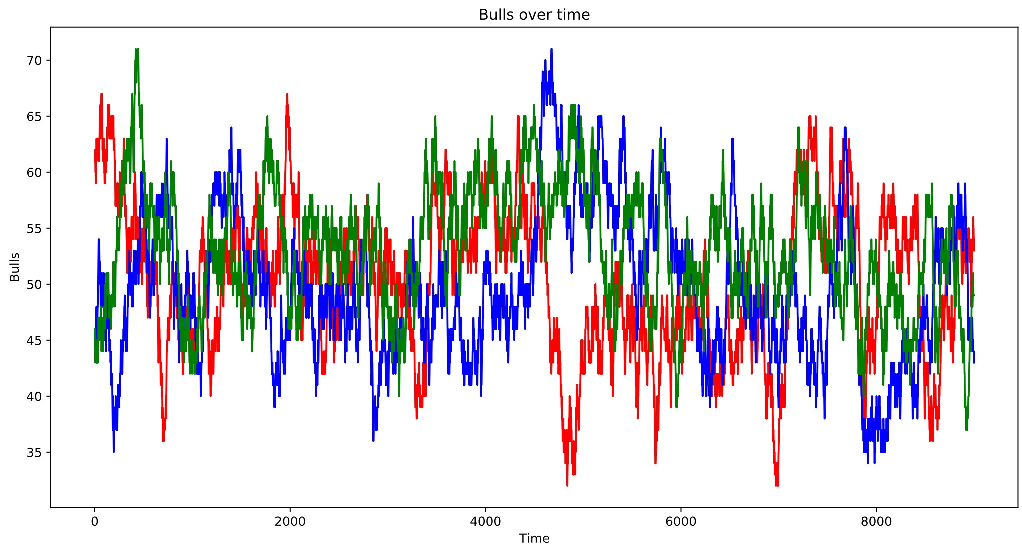
```
# Case 2: N << a < b:

N = 100
T = 9000
a = 4000
b = 6000

x = np.arange(0,T)
y1 = pickle.load( open( "sim_2.p", "rb" ))[0]
y2 = pickle.load( open( "sim_2.p", "rb" ))[1]
y3 = pickle.load( open( "sim_2.p", "rb" ))[2]

plt.figure(figsize=(14,7))
plt.title("Bulls over time")
plt.xlabel('Time')
plt.ylabel('Bulls')
plt.plot(x, y1, color="red")
plt.plot(x, y2, color="blue")
plt.plot(x, y3, color="green")

plt.show()
```



3. No preference, middling sensitivity ($\alpha = \beta = 100$)

These parameters settings are the most middling possible. We may see some time spent around the barriers, with some fluctuation across the central region. Since $\alpha = \beta$, we expect the process to reach and hover near an even split between bulls and bears (50). Compared to case 2, we expect to see a slightly higher average value and slightly more erratic movement.

```
# Case 2: a = b = 100

N = 100
T = 9000
a = 100
b = 100

x = np.arange(0,T)
y1 = pickle.load( open( "sim_3.p", "rb" ))[0]
y2 = pickle.load( open( "sim_3.p", "rb" ))[1]
y3 = pickle.load( open( "sim_3.p", "rb" ))[2]

plt.figure(figsize=(14,7))
plt.title("Bulls over time")
plt.xlabel('Time')
plt.ylabel('Bulls')
plt.plot(x, y1, color="red")
plt.plot(x, y2, color="blue")
plt.plot(x, y3, color="green")

plt.show()
```



Macrostates

With the simulated microstates of the economy, we can now simulate the macrostate of the model: the price history.

We define the process S_t, the stock price at time t: First, set S_0. At each time step, determine the price of the next increment by adding the excess demand (the number of bulls less the number of bears). If the price would go below 0, instead set to 0.

$$S_{t+1} - S_t = D(t) = X_t - (N - X_t) = 2X_t - N$$

Note that the maximum single-step increment is N, the total number of traders.

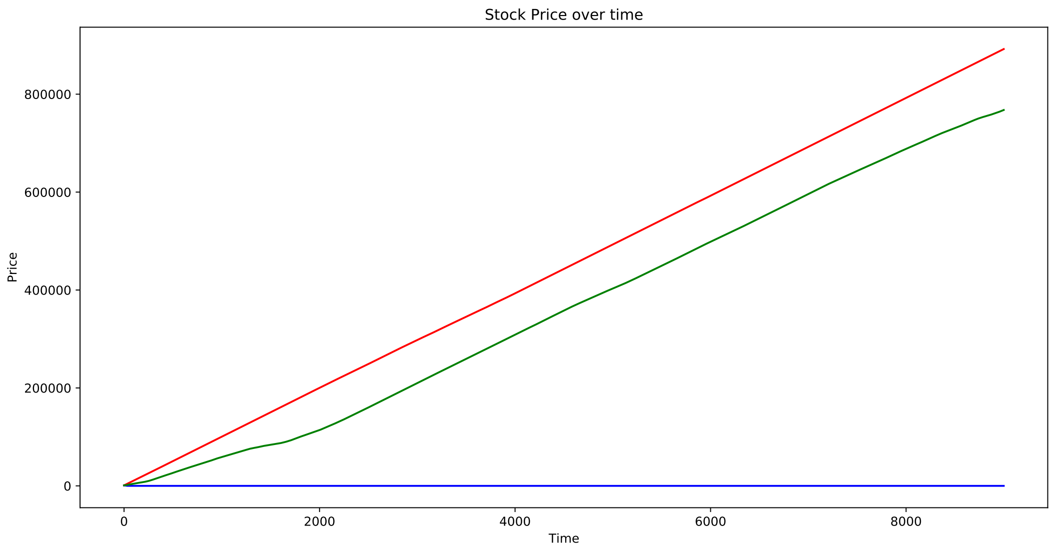
```
def price_sim(N, sim, S0):
    D_t = np.zeros([len(sim),len(sim[0])])
    prices = np.zeros([len(sim),len(sim[0])])
    for i in range(0,len(sim)):
        for j in range(1,len(sim[0])):
            prices[i][0] = S0
            D_t[i,j] = 2*sim[i][j-1] - N
            prices[i,j] = prices[i][j-1] + D_t[i][j]
            if prices[i,j]<0:
                prices[i,j] = 0
    return prices
```

```
N = 100
S0= 1000

sim = pickle.load( open( "sim_1.p", "rb" ))
pickle.dump(price_sim(N, sim, S0),open( "sim_1_price.p", "wb" ))

sim = pickle.load( open( "sim_2.p", "rb" ))
pickle.dump(price_sim(N, sim, S0),open( "sim_2_price.p", "wb" ))
```

```
sim = pickle.load( open( "sim_3.p", "rb" ))
pickle.dump(price_sim(N, sim, S0),open( "sim_3_price.p", "wb" ))
```

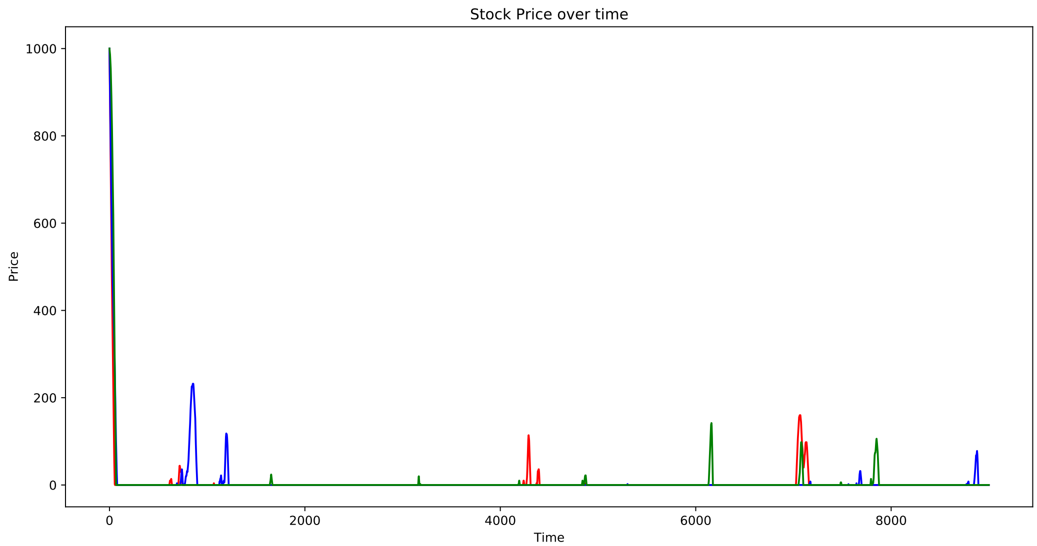
For each case, the price implied by the same three paths are shown.

```
T = 9000
x = np.arange(0,T)
y1 = pickle.load( open( "sim_1_price.p", "rb" ))[0]
y2 = pickle.load( open( "sim_1_price.p", "rb" ))[1]
y3 = pickle.load( open( "sim_1_price.p", "rb" ))[2]

plt.figure(figsize=(14,7))
plt.title("Stock Price over time")
plt.xlabel('Time')
plt.ylabel('Price')
plt.plot(x, y1, color="red")
plt.plot(x, y2, color="blue")
plt.plot(x, y3, color="green")

plt.show()
```

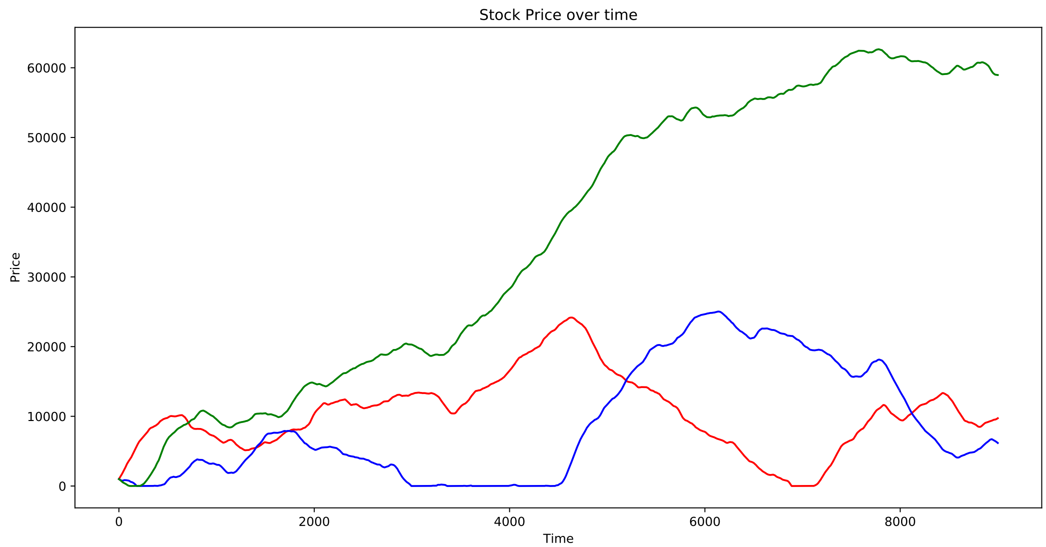



```
T = 9000
x = np.arange(0,T)
y1 = pickle.load( open( "sim_2_price.p", "rb" ))[0]
y2 = pickle.load( open( "sim_2_price.p", "rb" ))[1]
y3 = pickle.load( open( "sim_2_price.p", "rb" ))[2]

plt.figure(figsize=(14,7))
plt.title("Stock Price over time")
plt.xlabel('Time')
plt.ylabel('Price')
plt.plot(x, y1, color="red")
```

```
plt.plot(x, y2, color="blue")
plt.plot(x, y3, color="green")

plt.show()
```




```
T = 9000
x = np.arange(0,T)
y1 = pickle.load( open( "sim_3_price.p", "rb" ))[0]
y2 = pickle.load( open( "sim_3_price.p", "rb" ))[1]
y3 = pickle.load( open( "sim_3_price.p", "rb" ))[2]

plt.figure(figsize=(14,7))
plt.title("Stock Price over time")
plt.xlabel('Time')
plt.ylabel('Price')
plt.plot(x, y1, color="red")
plt.plot(x, y2, color="blue")
plt.plot(x, y3, color="green")

plt.show()
```



We also define the process R_t, the return at time t: First, set $R_0 = 0$. At each time step, determine the return of the increment by dividing the difference between the current and previous prices by the previous price. Returns can be negative, which occurs when the price decreases. If the price does not change, the return is undefined.

$$R_t = \frac{R_t - R_{t-1}}{R_{t-1}}$$

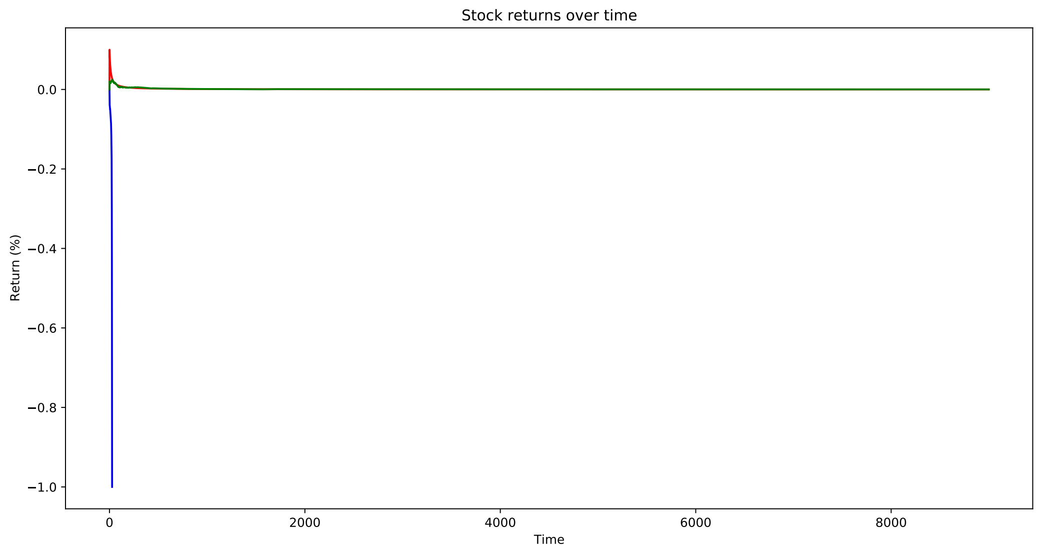
```
def return_sim(sim):
    returns = np.zeros([len(sim),len(sim[0])])
    for i in range(0,len(sim)):
        for j in range(1,len(sim[0])):
            if sim[i][j-1] != 0:
                returns[i,j] = (sim[i][j] - sim[i][j-1])/sim[i][j-1]
            else:
                returns[i,j] = None

    return returns
```

```
sim = pickle.load( open( "sim_1_price.p", "rb" ))
pickle.dump(return_sim(sim),open( "sim_1_return.p", "wb" ))

sim = pickle.load( open( "sim_2_price.p", "rb" ))
pickle.dump(return_sim(sim),open( "sim_2_return.p", "wb" ))

sim = pickle.load( open( "sim_3_price.p", "rb" ))
pickle.dump(return_sim(sim),open( "sim_3_return.p", "wb" ))
```

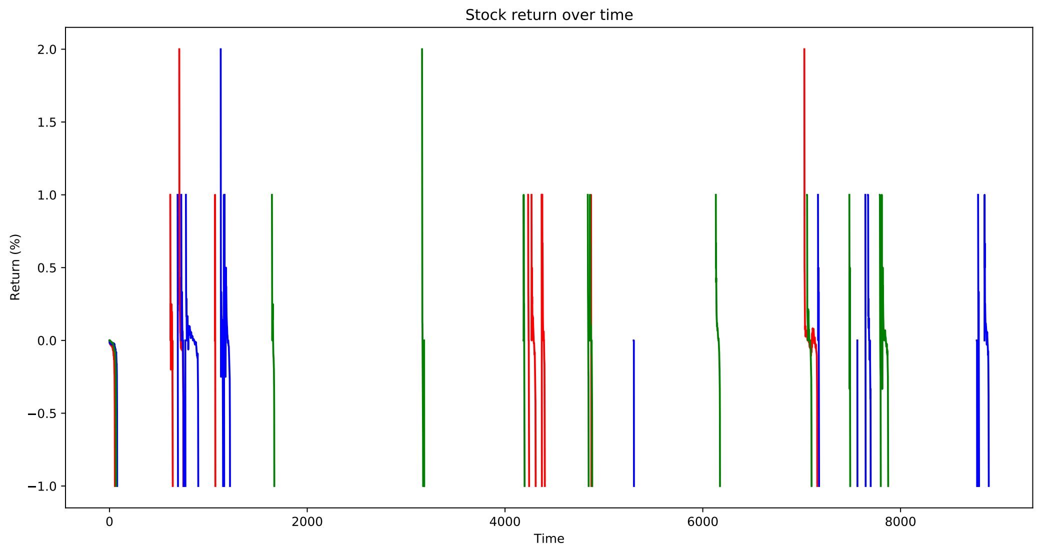
For each case, the returns implied by the same three paths are shown.

```
T = 9000
x = np.arange(0,T)
y1 = pickle.load( open( "sim_1_return.p", "rb" ))[0]
```

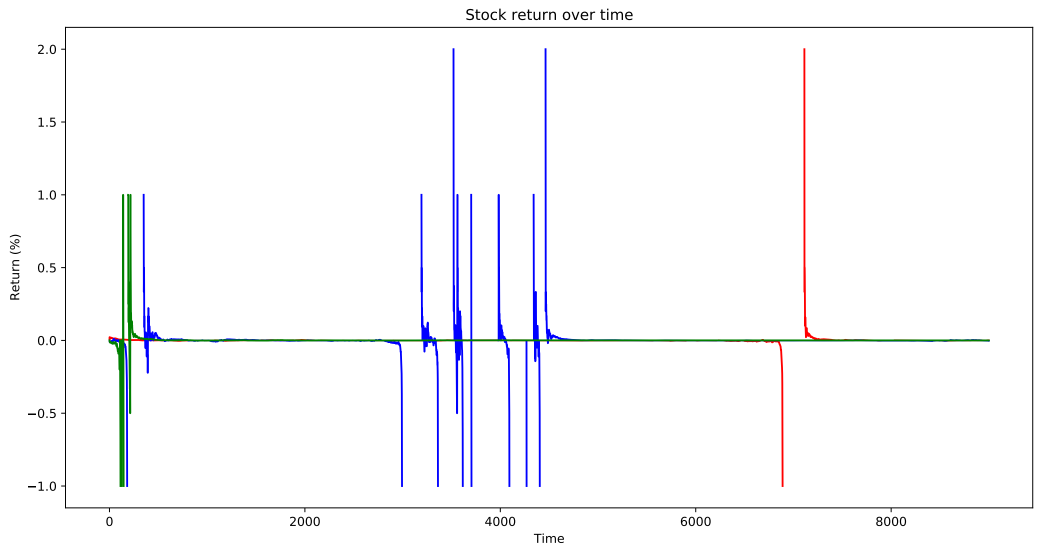
```
y2 = pickle.load( open( "sim_1_return.p", "rb" ))[1]
y3 = pickle.load( open( "sim_1_return.p", "rb" ))[2]

plt.figure(figsize=(14,7))
plt.title("Stock returns over time")
plt.xlabel('Time')
plt.ylabel('Return (%)')
plt.plot(x, y1, color="red")
plt.plot(x, y2, color="blue")
plt.plot(x, y3, color="green")

plt.show()
```




```
T = 9000
x = np.arange(0,T)
y1 = pickle.load( open( "sim_2_return.p", "rb" ))[0]
y2 = pickle.load( open( "sim_2_return.p", "rb" ))[1]
y3 = pickle.load( open( "sim_2_return.p", "rb" ))[2]

plt.figure(figsize=(14,7))
plt.title("Stock return over time")
plt.xlabel('Time')
plt.ylabel('Return (%)')
plt.plot(x, y1, color="red")
plt.plot(x, y2, color="blue")
plt.plot(x, y3, color="green")

plt.show()
```




```
T = 9000
x = np.arange(0,T)
y1 = pickle.load( open( "sim_3_return.p", "rb" ))[0]
y2 = pickle.load( open( "sim_3_return.p", "rb" ))[1]
y3 = pickle.load( open( "sim_3_return.p", "rb" ))[2]

plt.figure(figsize=(14,7))
plt.title("Stock return over time")
plt.xlabel('Time')
plt.ylabel('Return (%)')
plt.plot(x, y1, color="red")
plt.plot(x, y2, color="blue")
plt.plot(x, y3, color="green")

plt.show()
```



Model Analysis

We will extract the prices returns at steps 3000, 6000, and 9000 from all 10000 simulations and plot them in histograms to help us understand the distribution of returns over time.

Predictions

We attempt predict the results of the analysis.

Case 1 ($0 < \beta < \alpha < 1$)

The first case we analyzed with the goal of mirroring extreme herd sensitivity, with a preference to be bullish. We set our parameters to $\alpha = 0.8$, $\beta = 0.2$ with our initial bull and bear distribution being randomly drawn from the stationary distribution. We will make a few predictions about what we expect our return and price distributions to look like after running the simulations.

Our first prediction is regarding the mean of the return distributions. We believe the mean of the distributions will be roughly equal to the alpha parameter. This is because the herd sensitivity aspect of the economy works the same in both directions, the only difference leading to one urn being preferred to the other is in the preference parameters. Thus, whichever of the two preference parameters is larger will draw the mean of the return distribution to that side of 0. The reason we hypothesize the mean to be equal to the alpha parameter specifically in this case, is that the alpha parameter is actually equal to the fraction of the sum of the parameters which represents the dominant preference. Thus I hypothesize that on average, roughly 80% of the market will be bullish leading to returns centered around 0.8.

Our second prediction is with regard to the tails of the distribution. We hypothesize that the distribution will innately have heavy tails and as time progresses will develop heavier and heavier tails. The cause of these extremely heavy tails is the 'domino effect' discussed earlier in this paper. As the majority of traders slip into one state, the state begins to draw in more and more traders with the increasing probability. Eventually, we will have a very small fraction or simulations which will remain in an all bull or all bear state for extended periods of time, thus generating extremely high or low price values which will create large swings in returns.

We believe that overall the distributions will have a very sharp peak around the mean for the simulations which acted in a more 'predictable' manner, surrounded by a very long drawn out tails, representing any of the simulations where the 'domino effect' of the herd played a major role. We believe this attribute of the first case to be the most interesting aspect of our study with regard to explaining real world phenomena.

The third prediction which we make about our first case we analyze is that it will be left skewed. The reason for this left skew is again, the underlying herd sensitivity. When we have alpha larger than beta, the mean of the distribution will obviously be pushed in the positive direction. However, when the herd sensitivity comes into play, we occasionally will have a group of traders who flip to the bearish stance despite the overall market preference to act bullish. Once this happens we will have a period of time where the herd mentality dominates and traders swing to become bears. We then have periods of a bearish majority, driving returns down and skewing the return distribution to the left.

Case 2 ($\alpha < \beta << 1$)

The second case was used to contextualize the influence our α and β parameters played in return distributions. We initialized the model parameters to $\alpha = 4000$, $\beta = 6000$, and used a random generator to choose our starting bear/bull values from the analytically calculated stationary distribution.

Our first prediction regarding the second case is that the distribution of returns will be much more Gaussian than in the first case. We believe this to be true because the large preference parameters will, for the most part, dominate the underlying herd tendencies of the economy. Specifically, we expect the tails of the second case to be much more condensed than the first. This will lead to a much more gradually sloping mean peak. Our economy will behave much more similarly to a normal Ehrenberg urn than in the first case, with a slight preference to one side being the main exception to the standard Ehrenberg model. While we expect the distribution of returns to be somewhat similar to a Gaussian distribution we expect there to be a right skew to the data. This is for the same reason as the skew in the first case, this time it is in the opposite direction because the economy is bear preferring for these parameters. Specifically, although the majority of the time we will have a bear-heavy market, occasionally a group of traders will be pulled to the bullish urn and the 'domino effect' will occur. It is important to note however, that the skew should be much less apparent in this case than in the first. We believe this to be true due to suppressed relative herd sensitivity.

Case 3 ($\alpha = \beta = 100$)

The third case is created with the goal of trying to make an economy where herd behavior and position preference were obsolete. This will allow us to use this state as a basis to contextualize the effects of the herd behavior and preference parameter alone. We initialize the parameters to be $\alpha = \beta = 100$ and again randomly select our initial bull and bear values from the stationary distribution.

Our hypothesis is that the distribution of returns will be almost perfectly normal, with very small standard deviation and a mean of 0. We predict this to be the case because the economy actually prefers to be in a sort of 'neutral state.' We see that our parametric selections drive the bulls to want to be bears and the bears to want to be bulls. With such large parameters driving away from the extremes of all bears or all bulls, we will see tight groupings around a 50-50 split in the data and the price to hover around the starting value the majority of the time.

Results

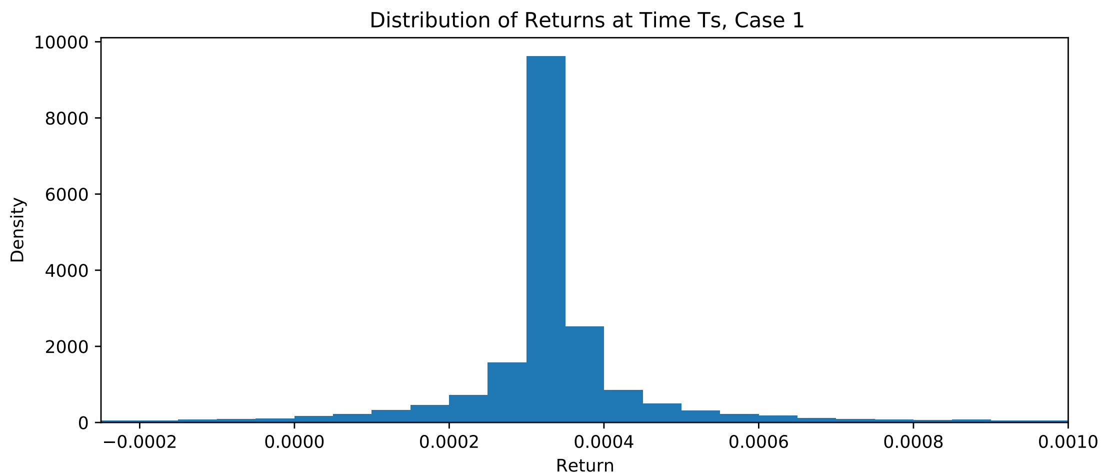
```
Ts = 3000
Tm = 6000
Th = 9000
M = 10000
```

```
Return_case_1 = pickle.load( open( "sim_1_return.p", "rb" ))

# CASE 1
R_Ts1 = []
R_Tm1 = []
R_Th1 = []
for i in range(0,M):
    R_Ts1.append(Return_case_1[i][Ts-1])
    R_Tm1.append(Return_case_1[i][Tm-1])
    R_Th1.append(Return_case_1[i][Th-1])

#Graph 1
fig, ax = plt.subplots(figsize=(10, 4))
n, bins, patches = plt.hist(R_Ts1, density = True, bins = 30000)
plt.xlim(xmin=-0.00025, xmax = 0.001)
plt.title("Distribution of Returns at Time Ts, Case 1")
plt.xlabel('Return')
plt.ylabel('Density')
plt.show()

#Graph 2
fig, ax = plt.subplots(figsize=(10, 4))
n, bins, patches = plt.hist(R_Tm1, density = True, bins = 10000)
plt.xlim(xmin=-0.00025, xmax = 0.0006)
plt.title("Distribution of Returns at Time Tm, Case 1")
plt.xlabel('Return')
plt.ylabel('Density')
plt.show()

#Graph 3
fig, ax = plt.subplots(figsize=(10, 4))
n, bins, patches = plt.hist(R_Th1, density = True, bins = 70000)
plt.xlim(xmin=-0.00025, xmax = 0.00045)
plt.title("Distribution of Returns at Time Th, Case 1")
plt.xlabel('Return')
plt.ylabel('Density')
plt.show()

Return_case_2 = pickle.load( open( "sim_2_return.p", "rb" ))

#CASE 2
R_Ts2 = []
R_Tm2 = []
R_Th2 = []
for i in range(0,M):
    R_Ts2.append(Return_case_2[i][Ts-1])
    R_Tm2.append(Return_case_2[i][Tm-1])
    R_Th2.append(Return_case_2[i][Th-1])

#Graph 4
fig, ax = plt.subplots(figsize=(10, 4))
n, bins, patches = plt.hist(R_Ts2, density = True, bins = 500)
plt.xlim(xmin=-0.04, xmax = 0.04)
```

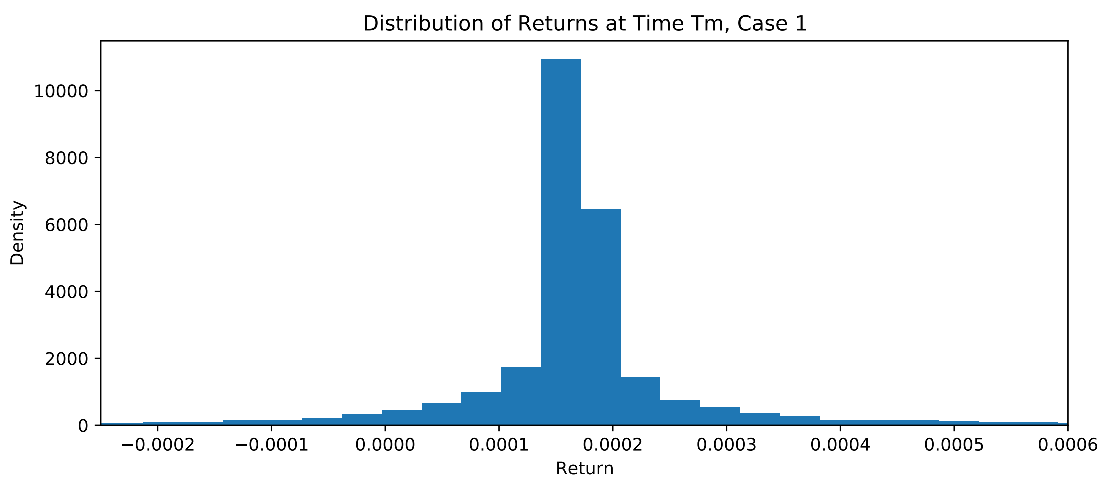
```
plt.title("Distribution of Returns at Time Ts, Case 2")
plt.xlabel('Return')
plt.ylabel('Density')
plt.show()



#Graph 5
fig, ax = plt.subplots(figsize=(10, 4))
n, bins, patches = plt.hist(R_Tm2, density = True, bins = 400)
plt.xlim(xmin=-0.04, xmax = 0.04)
plt.title("Distribution of Returns at Time Tm, Case 2")
plt.xlabel('Return')
plt.ylabel('Density')
plt.show()

#Graph 6
fig, ax = plt.subplots(figsize=(10, 4))
n, bins, patches = plt.hist(R_Th2, density = True, bins = 500)
plt.xlim(xmin=-0.05, xmax = 0.05)
plt.title("Distribution of Returns at Time Th, Case 2")
plt.xlabel('Return')
plt.ylabel('Density')
plt.show()

Return_case_3 = pickle.load( open( "sim_3_return.p", "rb" ))

#CASE 3
R_Ts3 = []
R_Tm3 = []
R_Th3 = []
for i in range(0,M):
    R_Ts3.append(Return_case_3[i][Ts-1])
    R_Tm3.append(Return_case_3[i][Tm-1])
    R_Th3.append(Return_case_3[i][Th-1])

#Graph 7
fig, ax = plt.subplots(figsize=(10, 4))
n, bins, patches = plt.hist(R_Ts3, density = True, bins = 800)
plt.xlim(xmin=-0.05, xmax = 0.05)
plt.title("Distribution of Returns at Time Ts, Case 3")
plt.xlabel('Return')
plt.ylabel('Density')
plt.show()

#Graph 8
fig, ax = plt.subplots(figsize=(10, 4))
n, bins, patches = plt.hist(R_Tm3, density = True, bins = 800)
plt.xlim(xmin=-0.05, xmax = 0.05)
plt.title("Distribution of Returns at Time Tm, Case 3")
plt.xlabel('Return')
plt.ylabel('Density')
plt.show()
```

```
#Graph 9
fig, ax = plt.subplots(figsize=(10, 4))
n, bins, patches = plt.hist(R_Th3, density = True, bins = 1000)
plt.xlim(xmin=-0.02, xmax = 0.02)
plt.title("Distribution of Returns at Time Th, Case 3")
plt.xlabel('Return')
plt.ylabel('Density')
plt.show()
```

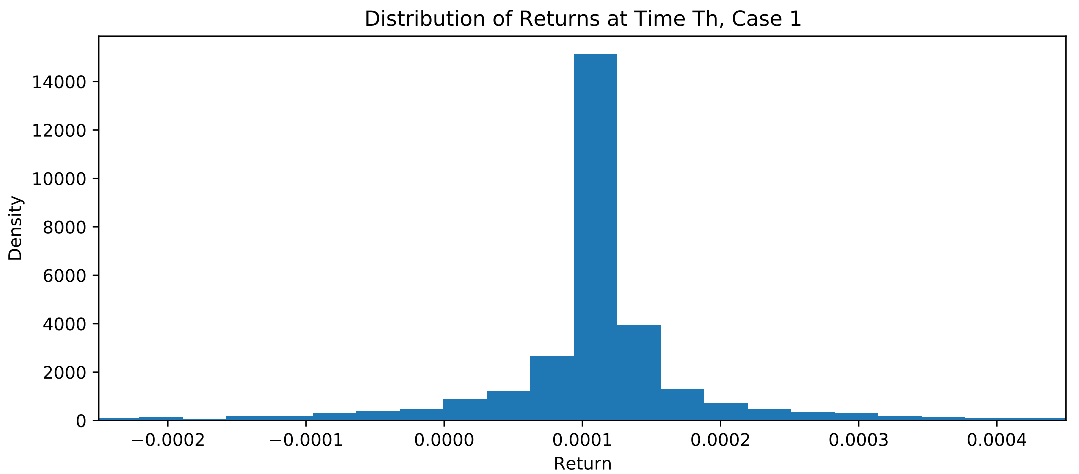
```
## (-0.00025, 0.001)
```



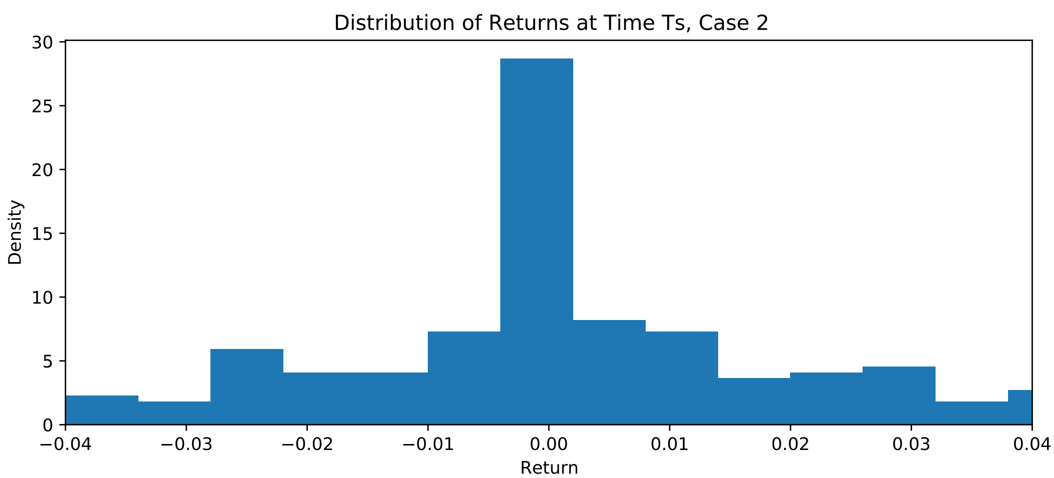

```
## (-0.00025, 0.0006)
```



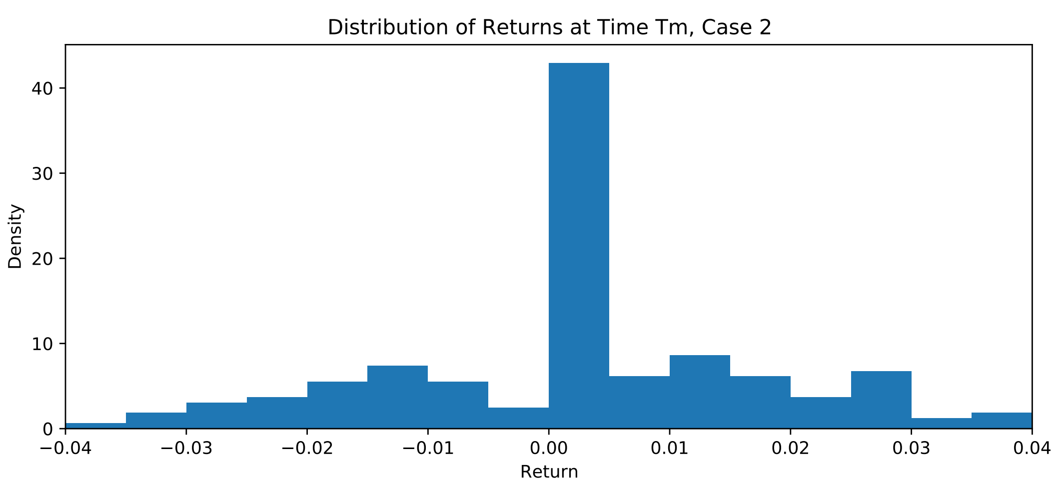

```
## (-0.00025, 0.00045)
```



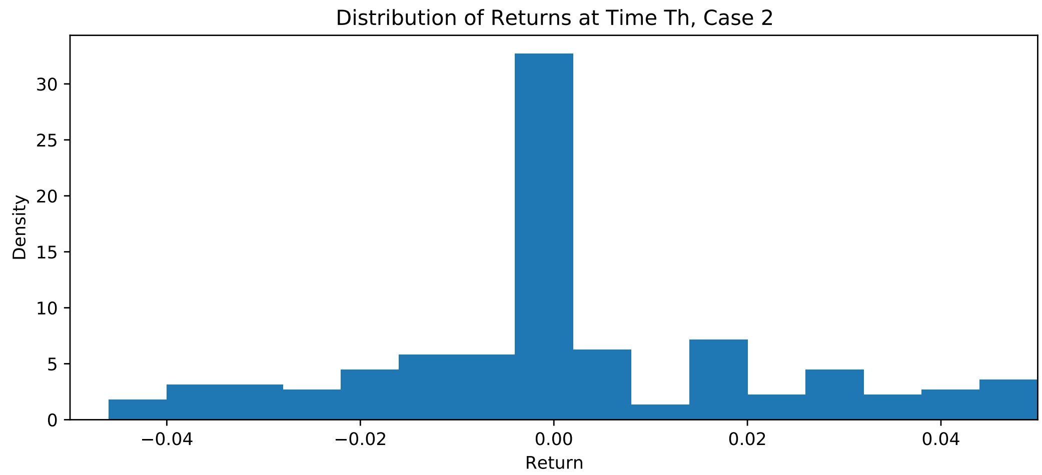

```
## (-0.04, 0.04)
```



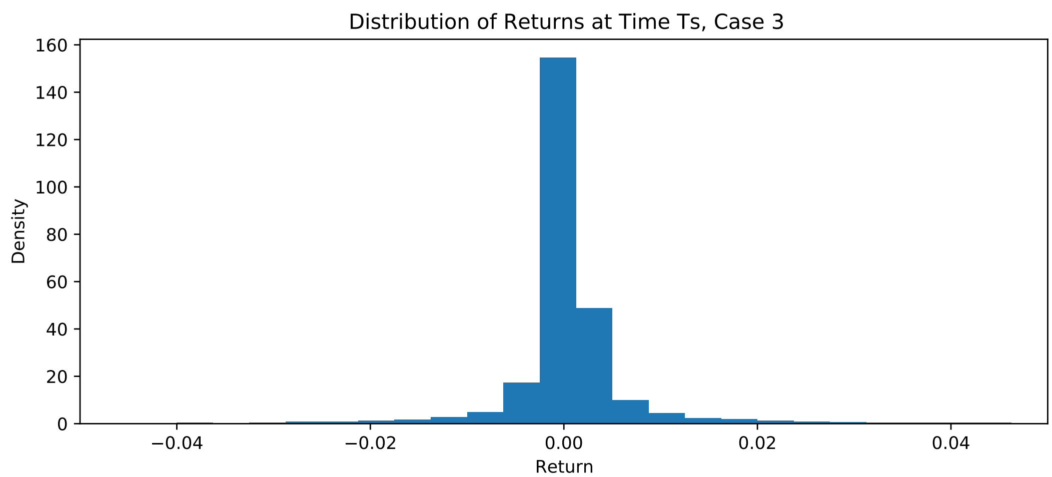

```
## (-0.04, 0.04)
```



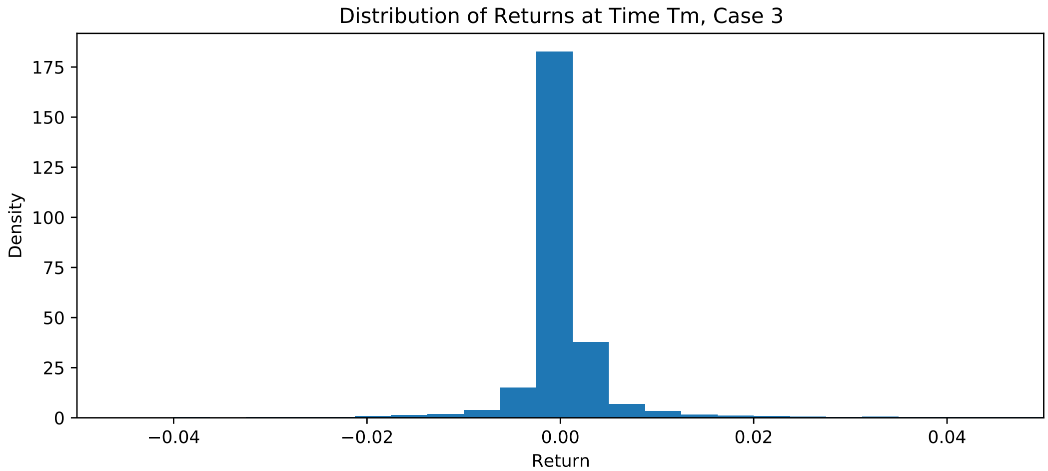
```
## (-0.05, 0.05)
```



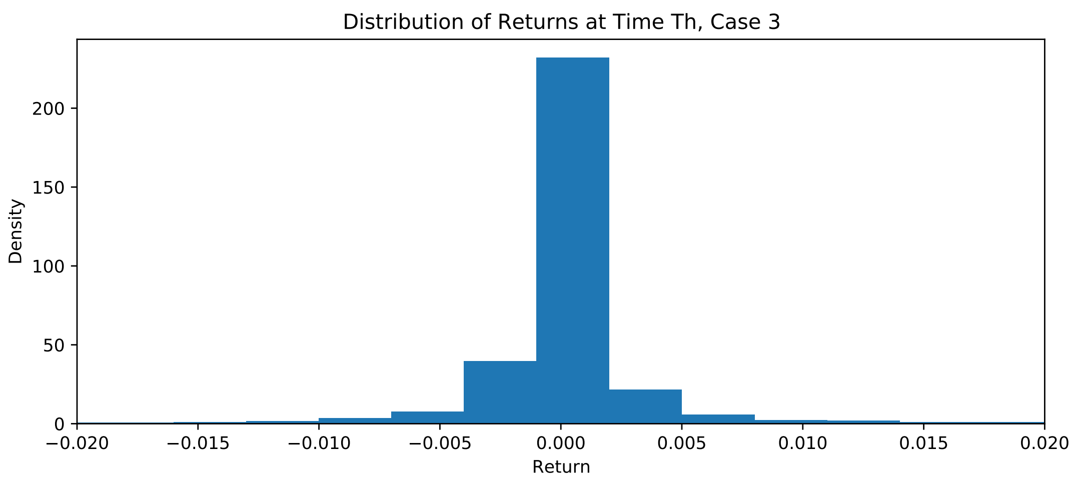

```
## (-0.05, 0.05)
```




```
## (-0.05, 0.05)
```




```
## (-0.02, 0.02)
```



Case 1 $(0 < \beta < \alpha < 1)$

In the first case we analyzed, we wanted to simulate extreme herd sensitivity with a preference to be bullish. Setting the parameters $\alpha = 0.8$, $\beta = 0.2$ with with our initial bull and bear distribution being randomly drawn from the stationary distribution.

Our initial prediction stated that we expect the tails of the distribution to exhibit heavier tails as time progresses. Judging from the tails of the 3 graphs below, our predictions match our observed results as the tails of the distributions become heavier as time increases due to the 'domino effect' discussed earlier in this paper. As our prediction states, there is an extremely sharp peak around the mean indicating our distribution has high kurtosis. This high kurtosis can be attributed to the fact that as the majority of traders tend to slip into one state, that state attracts other traders resulting in large and heavy swings of returns.

In addition, our third prediction states we expect the graph to be left skewed. As our histogram depicts, the distributions at each time step are left skewed which can be attributed to herd sensitivity. More specifically, as one group of bearish traders go against the bullish trends of the market, there exists a period of time where the herd mentality dominates and an increasing number of traders swing from a bullish to bearish stance causing a period of bearish majority. This drives returns down and skews the distribution to the left and contributes to an increase in kurtosis as conveyed in the figures above.

Case 2 ($\alpha < \beta << 1$)

Our goal with this simulation is to contextualize the role our α and β parameters play in determining return distributions. We initialize the model with parameters to $\alpha = 4000$, $\beta = 6000$ and used a random generator to choose our starting bear/bull values from the analytically calculated stationary distribution. Our first prediction states that the resulting distributions will be Gaussian with lower kurtosis compared to the first case due to the expectation of the large parameters to dominate the underlying herd tendencies of the economy.

Our results indicate that the first case produces more of a Gaussian distribution than the second case as shown by the figures below. We observe that the distribution at the first time step Ts is more of a Gaussian distribution compared to our first case, yet the distribution at time Tm is less of a Guassian distribution. The second distribution displays a very sharp peak as well as non-uniformity centered around the mean signifying an excess level of kurtosis. As our prediction states, the large parameters dominate the underlying herd tendencies of the market with a slightly right skewed distribution around the mean. This is as expected since our β corresponds to bullish tendencies and it is larger than α. However, we notice a higher level of kurtosis in this case than in the previous. In addition, it can be seen in the figure (right) that there are numerous local peaks throughout all time increments. These local peaks and cliffs are attributed to the 'domino effect' mentioned above where a group of bearish traders will be pulled towards bearish tendencies. This is a phenomenon known as suppressed relative herd sensitivity.

Case 3 ($\alpha = \beta = 100$)

The third case is created with the goal of trying to make an economy where herd behavior and position preference were obsolete, allowing us to use this state as a basis to contextualize the effects of the herd behavior and preference parameter alone. We initialize the parameters to be $\alpha = \beta = 100$ and again randomly select our initial bull and bear values from the stationary distribution. We believe the distribution of returns will be completely normal with minimal standard deviation and 0 mean. As depicted in the 3 figures below, we have strayed from our initial hypothesis and notice a non-gaussian distribution at each time increment. Similar to the first case, the distributions display the 'domino effect' of traders slipping in an out of one trading tendency. The only difference is whether traders begin with bearish or bullish investment tendencies. As they are randomly sampled, the initial investment tendency is also randomly decided. Similar to the first case, the distributions across time are left skewed, indicating herd sensitivity. This contributes to traders' tendency to flip from bearish to bullish and vice versa. Lastly, each distribution displays excess kurtosis with sharp peaks and heavy tails with the tails becoming heavier throughout time.

Discussion

As early as the 1960s, Mandelbrot pointed out the insufficiency of the normal distribution for modeling the marginal distribution of asset returns and their heavy-tailed character. Since then, the character of the distribution of price variations has been observed in various markets. One can quantify the distribution through examining the kurtosis of the distribution. From Cont (2001), the distribution of returns contains a few stylized facts which are present within our data:

1. Heavy Tails: the distribution follows a sharp peak with a steep fall off.
2. Aggregated Time Scale: as Ts, Tm, and Th increase, the distribution looks more and more like a Gaussian distribution.
3. Local peaks: at any given time scale, a high degree of variability can be seen. This is quantified by the presence of irregular bursts in time series of a wide variety of volatility estimators.
4. Volume/Volatility Correlation: trading volume is directly correlated with volatility.

According to the paper, we see that the 30 minute density's of the S&P 500 follow a sharp peaked Gaussian distribution with high kurtosis and heavy tails.

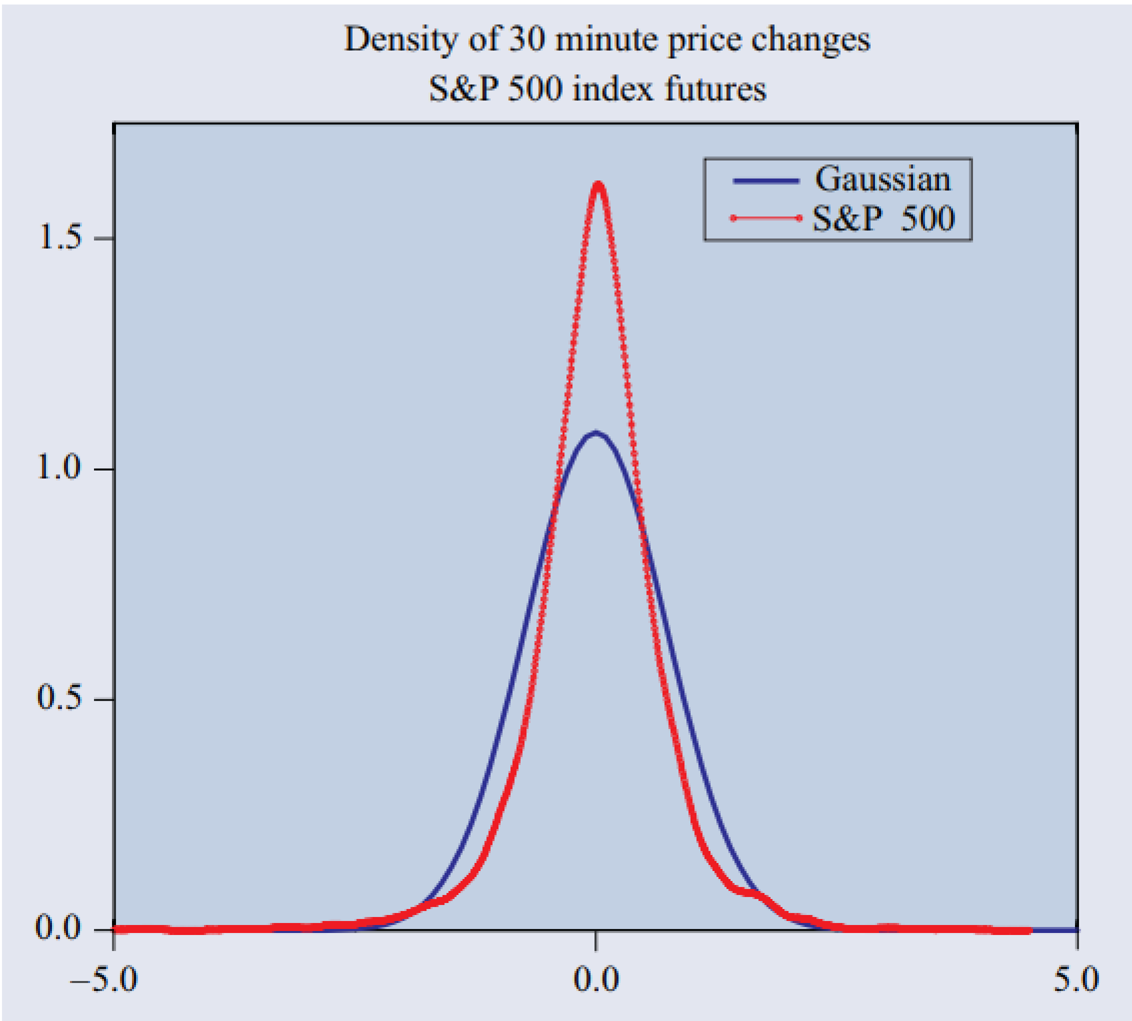


Figure 1: Excess Kurtosis in the S&P 500

Conclusion

We successfully implemented and simulated the Ehrenfest model described by Cont and Bouchaud (2000). Our analysis of the model parameters α and β supports their original definitions in all sets of parameters tested. Additionally, we generalize their original interpretations past the original three cases they present.

When trying to contextualize the influence α and β played in resulting distributions, we found that mirroring extreme herd sensitivity was possible. Setting α to 0.8 and β to 0.2 indicates a preference to be bullish led to a distribution of sharp peaks, heavy tails, large kurtosis, as well as a left skew. Setting traders to have bullish tendencies led to a distribution similar to that of the S&P 500 distribution as time persists. Herd sensitivity is noticed least when traders are equally split to have bullish and bearish investment tendencies as kurtosis is observably lowest when such parameters are set. Lastly, when β is larger than α , we discover a sharp peak with a large drop off and interestingly, local peaks where only a select few traders exhibit herd sensitivity. This high kurtosis can be attributed to the fact that as the majority of traders tend to slip into one state and remain within that state over time. However, it can be seen that when the sample of investors are equally split to be bullish and bearish it is less likely for the market to remain in one state for an extended period of time.

References

Cont, Rama. 2001. "Empirical Properties of Asset Returns: Stylized Facts and Statistical Issues." *Quantitative Finance* 1 (2): 223–36. https://doi.org/10.1080/713665670.

Cont, Rama, and Jean-Philipe Bouchaud. 2000. "Herd Behavior and Aggregate Fluctuations in Financial Markets." *Macroeconomic Dynamics* 4 (2): 170–96. https://doi.org/10.1017/S1365100500015029.