Analyzing the Stock Market Using Chaos Theory

**Abstract**

The program I designed takes a set of price data for a stock or stock market and produces a graph displaying the price, Hurst exponent, Lyapunov exponent, and predictability over time.  This graph can be used to analyze the stock's rate of chaos and predictability to determine when or where to invest.  I used different sample data sets to analyze the effect of price on the Hurst and Lyapunov exponents, and thus the predictability of the market.

**Introduction**

My program generates graphs for the analysis of Chaos Theory for stock prices over time. By analyzing the Hurst and Lyapunov exponents of a stock, the trends the stock undergoes are showcased and the stock can be given a value pertaining to its predictability. The less the variance in the Hurst and Lyapunov exponents over time, the more predictable a stock is and thus the less likely the stock is to fluctuate. Specifically, the Hurst exponent determines the rate of chaos of a stock price and is related to the effect memory has on the stock market. If the Hurst exponent = 0.5, memory has no adverse effect on the stock price, however, if the Hurst exponent is > 0.5, if the price is going up, it is most likely going to keep increasing, and if the price is going down, it is most likely going to keep decreasing. If the Hurst Exponent is < 0.5, if the stock price is going up, it is most likely going to fall in the coming time period and if the stock price is falling, it is most likely to gain in the next time period. Because of this trend, the Hurst exponent is a very important for potentially predicting the future price of a stock. The Lyapunov exponent is related to the predictability of the stock, where the predictability p = 1 / L, where L is the Lyapunov exponent for a time period. The more predictable a stock is, the more value that can be placed into the accuracy of the Hurst exponent.

**Procedure**

To develop the Hurst (H(tau)) and Lyapunov (L(t)) functions I programmed a series of helper functions to make the design easier. Of the most basic helper functions, one is called data(x), which simply returns the value of the data at a given value of x, and the second is called avg(vals), which returns the average of all the values in the given list vals. My first helper function is x(tau), which takes a final time value, tau, and averages the values of the data from time t = 1 to t = tau. The next helper function X(t, tau) takes an initial time value, t, and final time value, tau, and calculates the sum of the deviations of each data point from the average value across the entire time interval from time t = 1 to t = tau. The function compiles these deviation sums in a list for each time interval starting at time t = t until t = tau, where t increases by one until it reaches the end of the interval, t = tau, and returns this list. The S(tau) function takes a final time value, tau, and calculates the sum of the squared deviations of the data from its average value across the time interval from time t = 1 to t = tau and returns the square root of the average of the calculated sum. The next helper function, R(t, tau) takes an initial time value, t, and final time value, tau, and returns the difference between the maximum and minimum value in the list returned by X(t, tau), which is the range of the deviations. Using the above helper functions the Hurst exponent can finally be calculated. The Hurst function, H(tau) takes a final time, tau, and calculates the log(R(tau)/(S(tau)\*c))/log(tau), where c is a constant that influences the values of the Hurst function and here equals 1.0. The other function for analyzing stocks with chaos theory, the Lyapunov function, L(t), takes a time value, t, and computes the sum of the log base 2 of the ratio between the data at t +1 and the ratio at t for t in the range of t = 1 to t = t. Then, it returns the average of the sum across the time period. In summary:

,

,

,

,

,

**Results**

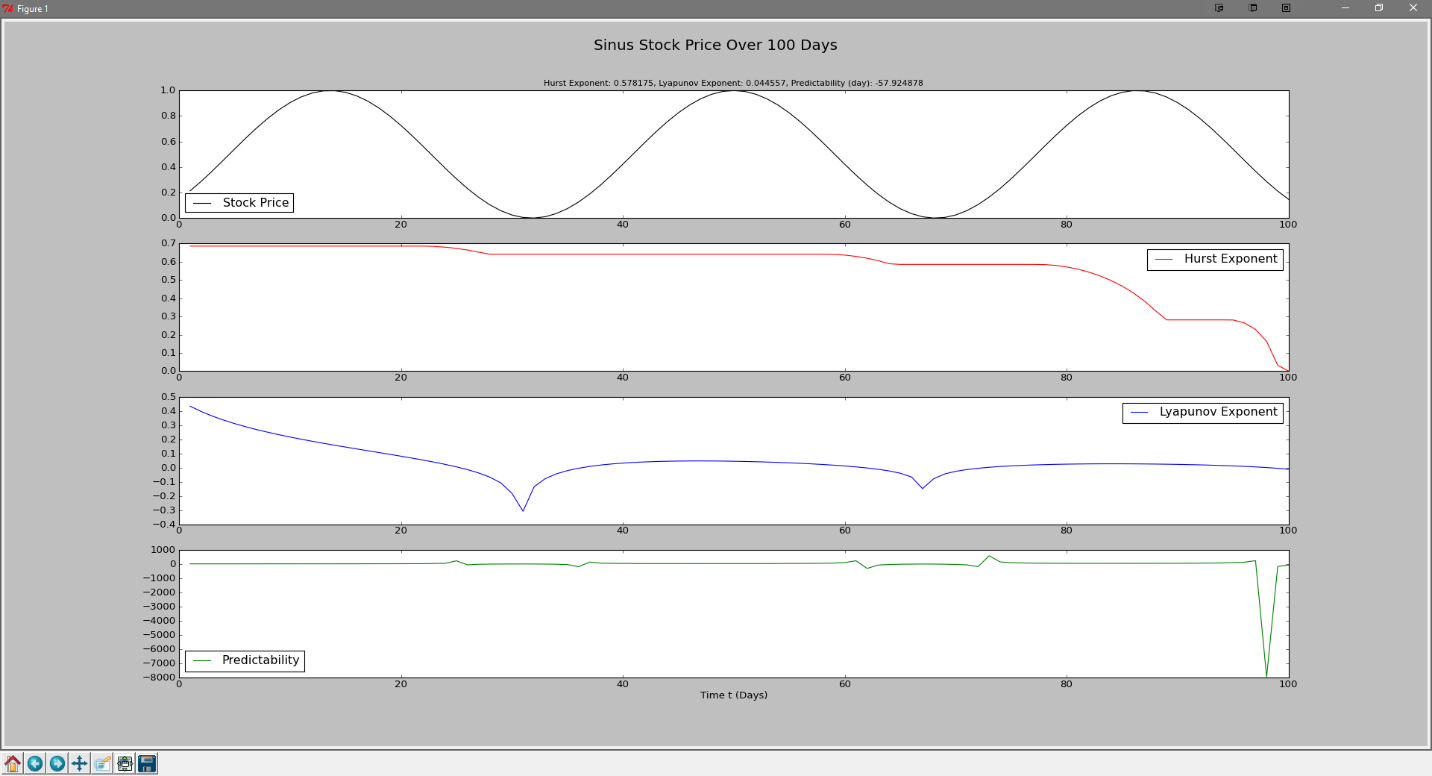


Figure 1 - Graphing the equation (sin(5.5\*pi\*x/100.0 + 5.5) + 1.0)/2.0 over 100 days

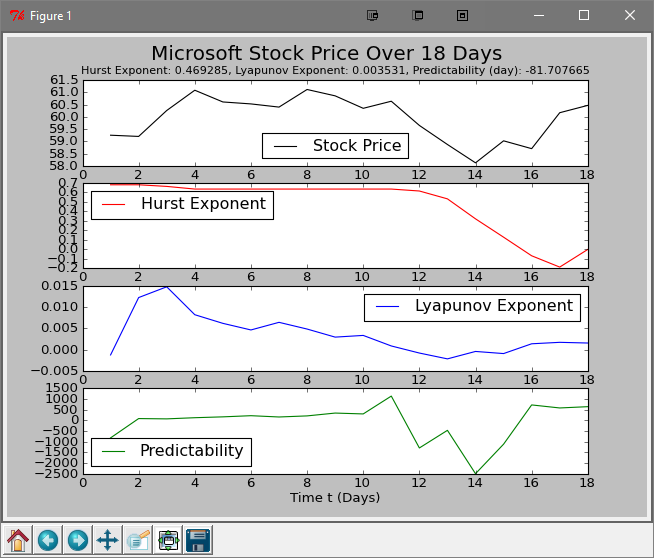


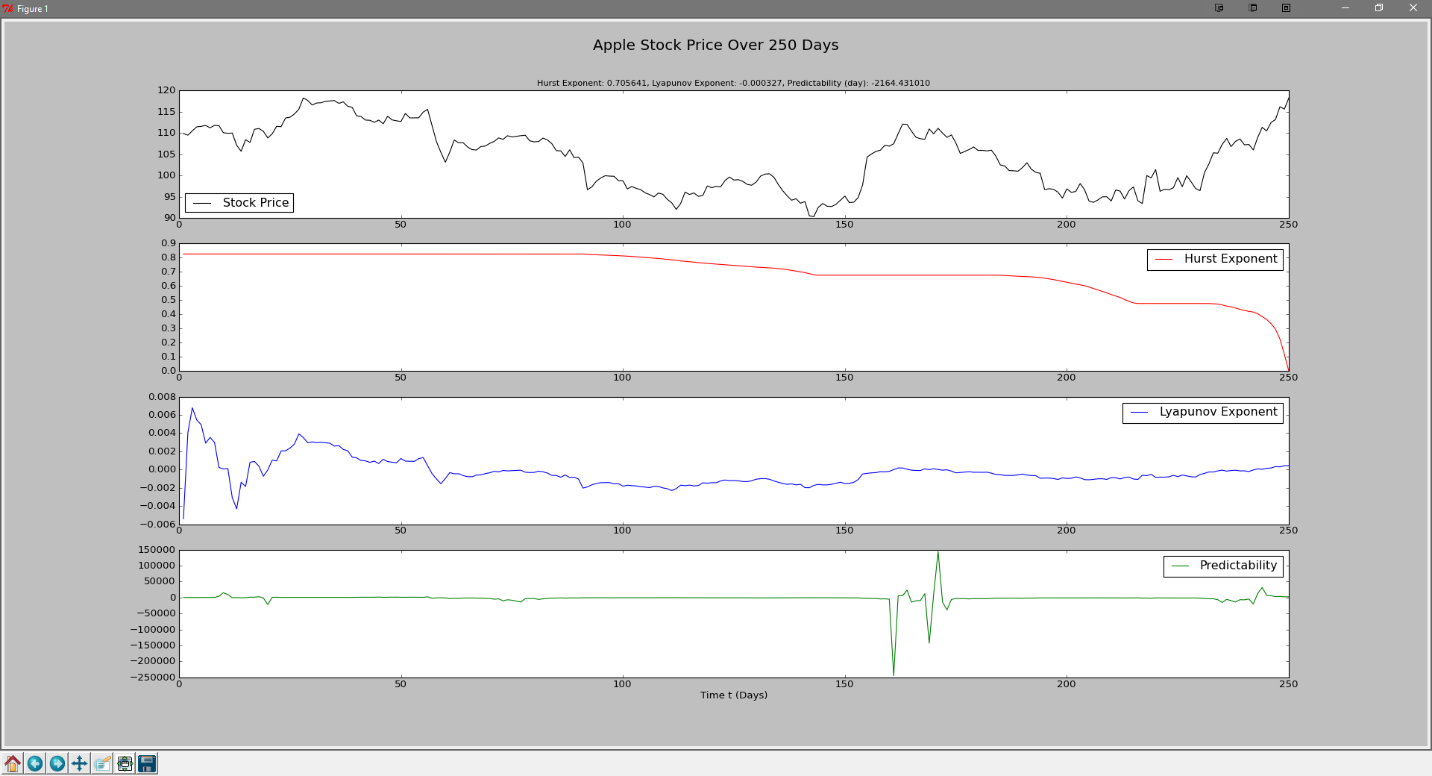
Figure 2- Graphing the MSFT stock over 18 days

Figure 3- Graphing the APPL stock over 250 days.

The results show that the predictability of the stock is highest (or lowest), and therefore most unpredictable at places where the stock fluctuates widely over a short period of time. On the sinus curve, where the fluctuations are small and uniform, the predictability is relatively constant and close to 0. However, in the real life stock graphs, the predictability is much more variant, to the point where it is unpredictable where the stock prices suddenly leap or drop. As far as the Hurst exponent shows, the overall trend is that the markets are expected to continue in their current form. The Hurst exponent appears to be off on the sinus curve however. Another noticeable trait of the graphs is the spike at the end of the Predictability and Hurst exponent graphs; this spike is not accurate and should be ignored, it is almost always part of the last time period.

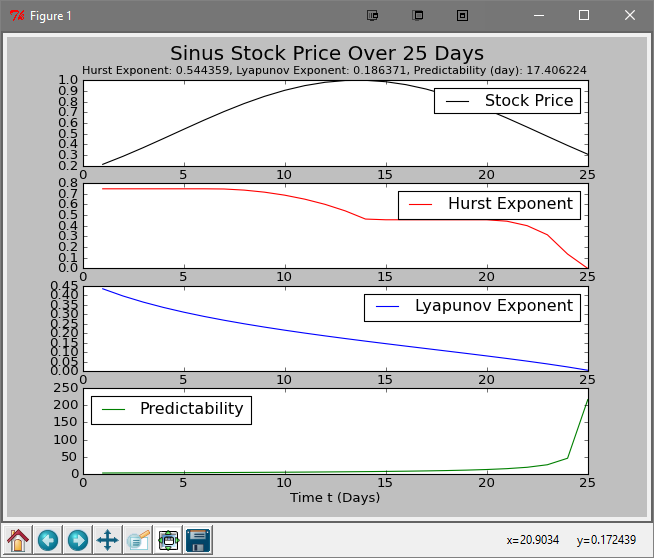
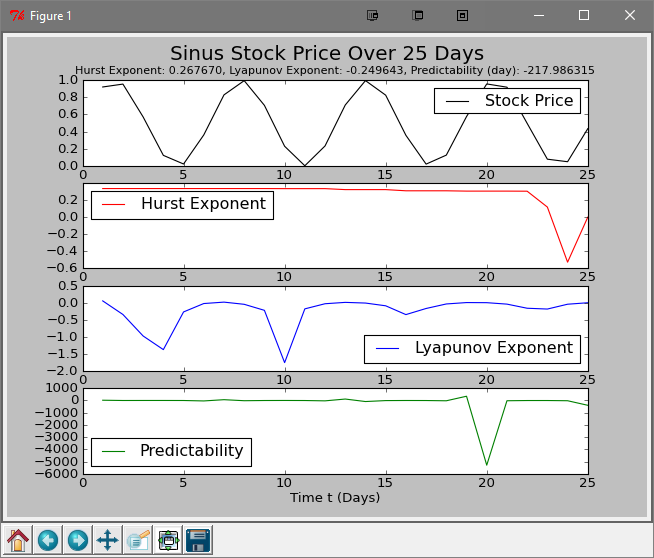
 

Figure 4 - Comparing the regular sinus curve with an adjusted sinus curve.

In the above comparison between the adjusted sinus curve ((sin(5.5\*pi\*x/100.0 + 5.5) + 1.0)/2.0) and the regular normalized sinus curve ((sin(x) + 1.0)/2.0), the Hurst and Lyapunov exponents are quite different. Both of the exponents are slowly decreasing over time across one sinus deviation and are relatively stable with the exception of spikes in the Lyapunov exponent when the curve is changing direction on the normal sinus curve.

Some average data over short durations of time:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stock | Duration | Hurst Exponent | Lyapunov Exponent | Predictability |
| Sinus (normal) | 25 Days | 0.26767 | -0.249643 | -217.986315 |
| Sinus (adjusted) | 25 Days | 0.544359 | 0.186371 | 17.406224 |
| APPL (Apple) | 25 Days | 0.531523 | 0.000987 | 558.274305 |
| MSFT (Microsoft) | 18 Days | 0.469285 | 0.003531 | -81.707665 |

This data shows that taking the values of the exponents over an extended period of time tells very little about the specifics of a stock without breaking up the time period into smaller portions. It does however accurately portray the overall picture of a stock’s price history. The adjusted sinus and Apple stocks were on a sharp uptick at the end of the time period, and thus their Hurst exponents were larger than 0.5; the normal sinus curve was nearing the end of its downward turn and the Microsoft stock was nearing the end of its upward tick, and thus, both stock’s Hurst exponents were less than 0.5. The Lyapunov exponent and predictability and hard to account for over a large period of time because the spikes are averaged in with the stable data. In order to find the days of instability and unpredictability, the graphs must be looked at, but the average gives you a general overall picture. The normal sinus curve and Apple stock were in general unpredictable over the 25-day period. The adjusted sinus and Microsoft stocks however, were relatively predictable.

**Conclusion**

The results of this simulation are quite impressive, the Hurst and Lyapunov exponents are shown to be reliable predictors of a stock’s price and trajectory. Because they are reliable predictors, brokers can use them to predict when to buy and sell stocks based on current and past trends in the stock’s closing price. If the stock’s Hurst exponent is > 0.5, its price is rising, and the predictability of the stock is stable, then it is a good time to invest or look towards selling stocks as the price rises because the stock is expected to continue rising. If it is falling when its Hurst exponent is > 0.5, then it may be a good time to hold off buying or selling because it is expected to go the way of the bear. If a stock’s Hurst exponent is < 0.5 and its price is rising while the predictability is stable, then it’s price is expected to fall soon. If it was falling while H < 0.5, it will hopefully rise in time for a broker to invest.

The Hurst and Lyapunov functions are great predictors of a market, but unfortunately my algorithm for calculating the Hurst exponent seems to be inaccurate over long stretches of time. I could not pinpoint the error in the Hurst function, but on short timeframes the Hurst graphs are accurate. However, the Lyapunov exponent, and thus the predictability, is accurate over a large stretch of time and a very reliable predictor.

**References**

Dostál, Petr. "Chaos and Stock Market." Economy & Management – Decision Making. Brno University of Technology. Web. 1 Dec. 2016. <http://www.petrdostal.eu/papers/cla15.pdf>.

**Appendix**

I downloaded the stock data from Yahoo finance as a .csv file.

*"""  
FinalProject.py  
Jack Baumann's final project analyzing the stock market with Chaos Theory  
11/28/16  
"""***from** math **import** sqrt, log, sin, pi  
**import** matplotlib.pyplot **as** plt  
**from** numpy **import** loadtxt  
  
  
**def data**(x):  
 **return** prices[x-1]  
 # return (sin(5.5\*pi\*x/100.0 + 5.5) + 1.0)/2.0  
 # return (sin(x) + 1.0)/2.0  
  
  
**def x**(tau):  
 sum = 0  
 i = 1  
 **while** i <= tau:  
 sum += data(i)  
 i += h  
 **return** 1.0/tau \* sum  
  
  
**def X**(t, tau):  
 rvs = []  
 # t = 0  
 **while** t <= tau:  
 sum = 0  
 i = 1  
 **while** i <= t:  
 sum += data(i) - x(tau)  
 i += h  
 rvs.append(sum)  
 t += h  
 **return** rvs  
  
  
**def S**(tau):  
 sum = 0  
 i = 1  
 **while** i <= tau:  
 sum += (data(i) - x(tau)) \*\* 2  
 i += h  
 **return** sqrt(1.0/tau \* sum)  
  
  
**def R**(tau):  
 # 1 <= t <= tau  
 **return** max(X(t, tau)) - min(X(t, tau))  
  
  
**def H**(tau):  
 c = 1.0 # constant that influences the value of H, default 1.0  
 **if** R(tau) == 0:  
 **return** 0  
 **return** log(R(tau)/(S(tau)\*c))/log(tau)  
  
  
**def L**(t):  
 sum = 0  
 i = 1  
 **while** i <= t:  
 **if** data(i) != 0 **and** data(i+h)/data(i) != 0:  
 sum += log(data(i+h)/data(i), 2)  
 i += h  
 **return** 1.0/t \* sum  
  
  
**def avg**(vals):  
 **return** sum(vals)/len(vals)  
  
  
t = 1  
tau = 25  
h = 1  
xs = []  
ts = []  
Hs = []  
Ls = []  
ps = []  
prices = loadtxt("Final Project Apple Stock Prices.csv", skiprows=1, delimiter=',')[:, 4]  
  
**while** t <= tau:  
 xs.append(data(t))  
 ts.append(t)  
 Hs.append(H(tau))  
 Lt = L(t)  
 Ls.append(Lt)  
 ps.append(1.0/Lt)  
 t += h  
  
sub = "Hurst Exponent: {0:f}, Lyapunov Exponent: {1:f}, Predictability (day): {2:f}".format(avg(Hs), avg(Ls), avg(ps))  
**print** sub  
plt.figure(1)  
plt.subplot(411)  
plt.title(sub, fontsize=10)  
plt.suptitle('Apple Stock Price Over 25 Days', y=0.97, fontsize=18)  
plt.plot(ts, xs, 'k-', label='Stock Price')  
plt.legend(loc='best')  
  
plt.subplot(412)  
plt.plot(ts, Hs, 'r-', label='Hurst Exponent')  
plt.legend(loc='best')  
  
plt.subplot(413)  
plt.plot(ts, Ls, 'b-', label='Lyapunov Exponent')  
plt.legend(loc='best')  
  
plt.subplot(414)  
plt.plot(ts, ps, 'g-', label='Predictability')  
plt.legend(loc='best')  
plt.xlabel('Time t (Days)')  
plt.show()