Jerry Cheng

Graphing Project

This graph compares several different crypto-currencies and their relative ages. Overall, over the last few months, crypto-currency prices have increased. Therefore, I theorized that crypto-currencies that are older have had more time to increase in value, resulting in a more valuable currency. Also, I hypothesized that older crypto-currencies, due to having stayed in the market for a longer period of time, would also have the factor of being more attractive to prospective crypto investors, due to it potentially having more security. And this appears to be true, at least for more popular currencies such as Bitcoin and Ethereum. However, this does not seem to be the case for less popular currencies, such as Ark and Ardor. Therefore, I will have to conclude that the age of crypto-currencies do not necessarily have an effect on the value of said currency.

Because some crypto-currencies have an incredibly low value and others have a comparably high value, a second subplot was created in order to better detail crypto-currencies with a relatively low monetary value.

Code:

import matplotlib.pyplot as plt, csv  
from pylab import plot  
  
first\_line = True  
with open('C:/Users/jerry/Desktop/crypto/bitcoin.csv') as csvfile:  
 readCSV = csv.reader(csvfile, delimiter=',')  
 bitcoin\_dates = []  
 bitcoin\_close = []  
  
 for row in readCSV:  
  
 if first\_line == True:  
 first\_line = False  
 continue  
  
  
 date = row[0]  
 close = row[4]  
 bitcoin\_close.append(float(close))  
  
with open('C:/Users/jerry/Desktop/crypto/bitcoin-cash.csv') as csvfile:  
 readCSV = csv.reader(csvfile, delimiter=',')  
 bitcoincash\_dates = []  
 bitcoincash\_close = []  
  
 for row in readCSV:  
  
 if first\_line == True:  
 first\_line = False  
 continue  
  
 date = row[0]  
 close = row[4]  
 bitcoincash\_close.append(float(close))  
  
with open('C:/Users/jerry/Desktop/crypto/ethereum.csv') as csvfile:  
 readCSV = csv.reader(csvfile, delimiter=',')  
 ethereum\_dates = []  
 ethereum\_close = []  
  
 for row in readCSV:  
  
 if first\_line == True:  
 first\_line = False  
 continue  
 date = row[0]  
 close = row[4]  
 ethereum\_close.append(float(close))  
  
with open('C:/Users/jerry/Desktop/crypto/ardor.csv') as csvfile:  
 readCSV = csv.reader(csvfile, delimiter=',')  
 ardor\_dates = []  
 ardor\_close = []  
  
 for row in readCSV:  
  
 if first\_line == True:  
 first\_line = False  
 continue  
  
 date = row[0]  
 close = row[4]  
 ardor\_close.append(float(close))  
  
with open('C:/Users/jerry/Desktop/crypto/ark.csv') as csvfile:  
 readCSV = csv.reader(csvfile, delimiter=',')  
 ark\_dates = []  
 ark\_close = []  
  
 for row in readCSV:  
  
 if first\_line == True:  
 first\_line = False  
 continue  
  
 date = row[0]  
 close = row[4]  
 ark\_close.append(float(close))  
  
ardor\_x\_values = range(len(ardor\_close))  
ethereum\_x\_values = range(len(ethereum\_close))  
bitcoincash\_x\_values = range(len(bitcoincash\_close))  
bitcoin\_x\_values = range(len(bitcoin\_close))  
ark\_x\_values = range(len(ark\_close))  
  
plt.subplots\_adjust(hspace=0.4)  
  
plt.subplot(211)  
plot(bitcoin\_x\_values, bitcoin\_close, bitcoincash\_x\_values, bitcoincash\_close, ethereum\_x\_values, ethereum\_close,  
 ardor\_x\_values, ardor\_close, ark\_x\_values, ark\_close)  
plt.yticks([0, 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000, 5500, 6000, 6500, 7000, 7500, 8000])  
plt.legend(["Bitcoin Price", "Bitcoin-Cash Price", "Ethereum Price", "Ardor Price", "Ark Price"], fontsize=20)  
plt.title('Crpyto-Currency Age vs Price', fontsize=40)  
plt.xlabel('Lifespan (in Days)', fontsize=30)  
plt.ylabel('Prices (USD)', fontsize=30)  
plt.grid(True)  
  
plt.subplot(212)  
plot(ardor\_x\_values, ardor\_close, ark\_x\_values, ark\_close)  
#plt.yticks([0, 0.3, 0.6, 0.9])  
plt.legend(["Ardor Price", "Ark Price"], fontsize=20)  
plt.title('Crpyto-Currencies Under $5', fontsize=40)  
plt.xlabel('Lifespan (in Days)', fontsize=30)  
plt.ylabel('Prices (USD', fontsize=30)  
plt.grid(True)  
  
plt.show()

Graph:

