"""I will be using 4 datasets on bitcoin daily trading for the years 2016-2019. I will be looking to find if there is a

relationship (if any) between the Open price, High, Low, Closing price, Volume of bitcoins, or the Market Cap. I will

be using simple linear regression analysis.

The orange plots are for the year 2016, the silver plots are for the year 2017, and the green plots are for the year 2018."""

Out[438]:

'I will be using 4 datasets on bitcoin daily trading for the years 2016-2019. I will be looking to find if there is a\nrelationship (if any) between the Open price, High, Low, Closing price, Volume of bitcoins, or the Market Cap. I will\nbe using simple linear regression analysis.\nThe orange plots are for the year 2016, the silver plots are for the year 2017, and the green plots are for the year 2018.'

In [359]:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

from statsmodels.formula.api import ols as sm

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

from scipy.stats import linregress

from sklearn import datasets, linear\_model

import seaborn as sns

In [360]:

#Reading the file for bitcoin daily trading for the year 2016.

#Renaming some of the columns so it will be easier to call later on.

bit2016 = pd.read\_csv('2016bitcointrading.csv')

bit2016.rename(columns = {'Market Cap':'Market\_Cap', 'Close\*\*':'Close', 'Open\*':'Open'}, inplace = True)

bit2016.head()

Out[360]:

Date Open High Low Close Volume Market\_Cap

0 31-Dec-16 960.63 963.74 947.24 963.74 99,135,104 15,492,554,222

1 30-Dec-16 972.53 972.53 934.83 961.24 187,474,000 15,450,507,055

2 29-Dec-16 975.13 979.40 954.50 973.50 199,320,000 15,645,617,392

3 28-Dec-16 934.83 975.92 934.83 975.92 236,630,000 15,682,769,405

4 27-Dec-16 908.35 940.05 904.26 933.20 167,308,000 14,994,286,713

In [361]:

#Here we are removing any commas in our data so we are able to call it later on.

bit2016['Market\_Cap'] = bit2016['Market\_Cap'].astype(str).str.replace(',','')

bit2016['Volume'] = bit2016['Volume'].astype(str).str.replace(',','')

bit2016['High'] = bit2016['High'].astype(str).str.replace(',','')

bit2016['Open'] = bit2016['Open'].astype(str).str.replace(',','')

In [362]:

bit2016.head()

Out[362]:

Date Open High Low Close Volume Market\_Cap

0 31-Dec-16 960.63 963.74 947.24 963.74 99135104 15492554222

1 30-Dec-16 972.53 972.53 934.83 961.24 187474000 15450507055

2 29-Dec-16 975.13 979.4 954.50 973.50 199320000 15645617392

3 28-Dec-16 934.83 975.92 934.83 975.92 236630000 15682769405

4 27-Dec-16 908.35 940.05 904.26 933.20 167308000 14994286713

In [402]:

#Returns some statistics on our dataframe.

bit2016.describe()

Out[402]:

Open High Low Close Volume Market\_Cap

count 366.000000 366.000000 366.000000 366.000000 3.660000e+02 3.660000e+02

mean 567.141503 574.302295 560.453415 568.492213 8.592451e+07 8.929758e+09

std 137.909543 140.461459 135.860670 139.255688 5.034484e+07 2.335454e+09

min 365.070000 374.950000 354.910000 364.330000 2.851400e+07 5.496598e+09

25% 431.967500 434.687500 427.197500 432.205000 5.633540e+07 6.574682e+09

50% 582.070000 588.960000 575.315000 582.555000 7.076340e+07 9.201876e+09

75% 663.332500 673.192500 653.625000 664.227500 9.517587e+07 1.046715e+10

max 975.130000 979.400000 954.500000 975.920000 3.633210e+08 1.568277e+10

In [403]:

#Changing the data types for the series in the dataframe so we can use them in our analysis.

bit2016.Volume = bit2016.Volume.astype(float)

bit2016.High = bit2016.High.astype(float)

bit2016.Open = bit2016.Open.astype(float)

bit2016.Close = bit2016.Close.astype(float)

bit2016.Low = bit2016.Low.astype(float)

bit2016.Market\_Cap = bit2016.Market\_Cap.astype(float)

In [410]:

#Assigning a series to our dependent and independent variables x and y.

#Plotting our x and y variables.

x = bit2016.Open

y = bit2016.Close

plt.scatter(x,y, color = 'orange')

plt.xlabel('Bitcoin Open Price per day, 2016')

plt.ylabel('Bitcoin Close price per day, 2016')

plt.show()

In [411]:

#Creating a function that will return the slope and intercept for our x and y variables.

def slope\_intercept(x\_val, y\_val):

x = np.array(x\_val)

y = np.array(y\_val)

m = (((np.mean(x)\*np.mean(y)) - np.mean(x\*y)) /

((np.mean(x)\*np.mean(x)) - np.mean(x\*x)))

m = round(m,2)

b = (np.mean(y) - np.mean(x)\*m)

b = round(b,2)

return m,b

slope\_intercept(x,y)

Out[411]:

(1.0, 1.35)

In [412]:

m,b = slope\_intercept(x,y)

reg\_line16 = [(m\*i)+b for i in x]

plt.scatter(x,y, color = 'orange')

plt.plot(x,reg\_line16)

plt.xlabel('Bitcoin open price per day, 2016')

plt.ylabel('Bitcoin close price per day, 2016')

plt.title("Regression line")

plt.show()

In [413]:

#A function that returns the root mean square error.

#The RMSE measures how spread out the residuals (prediction errors) are.

def rmse(y1,y\_hat):

y\_actual = np.array(y1)

y\_pred = np.array(y\_hat)

error = (y\_actual - y\_pred)\*\*2

error\_mean = round(np.mean(error))

error\_sq = np.sqrt(error\_mean)

return error\_sq

rmse(y,reg\_line16)

Out[413]:

14.66287829861518

In [417]:

#Splitting the data into training and testing.

#We will use one subset to train our model and the second subset to test our model.

x = bit2016data.Open.values.reshape(-1,1)

y = bit2016data.Close.values.reshape(-1,1)

xTrain, xTest, yTrain, yTest = train\_test\_split(x, y, test\_size = 1/4, random\_state = 0)

In [418]:

linearregressor = LinearRegression()

linearregressor.fit(xTrain,yTrain)

Out[418]:

LinearRegression(copy\_X=True, fit\_intercept=True, n\_jobs=1, normalize=False)

In [419]:

#Plotting our training data with the regression line.

plt.scatter(xTrain, yTrain, color = 'orange')

plt.plot(xTrain, linearregressor.predict(xTrain), color = 'black')

plt.title('Open vs. Close (Training set)')

plt.xlabel('Bitcoin Open price per day, 2016')

plt.ylabel('Bitcoin Close price per day, 2016')

plt.show()

In [420]:

#Plotting our test data with the regression line.

plt.scatter(xTest, yTest, color = 'orange')

plt.plot(xTrain, linearregressor.predict(xTrain), color = 'black')

plt.title('Open vs. Close (Test set)')

plt.xlabel('Bitcoin Open price per day, 2016')

plt.ylabel('Bitcoin Close price per day, 2016')

plt.show()

In [373]:

#Returns the mean for each series.

bit2016.mean(axis=0)

Out[373]:

Open 5.671415e+02

High 5.743023e+02

Low 5.604534e+02

Close 5.684922e+02

Volume 8.592451e+07

Market\_Cap 8.929758e+09

dtype: float64

In [374]:

bit2017 = pd.read\_csv('2017bitcointrading.csv')

bit2017.head()

Out[374]:

Date Open High Low Close Volume Market\_Cap

0 31-Dec-17 12,897.70 14,377.40 12,755.60 14,156.40 12,136,299,520 237,465,823,980

1 30-Dec-17 14,681.90 14,681.90 12,350.10 12,952.20 14,452,599,808 217,239,803,085

2 29-Dec-17 14,695.80 15,279.00 14,307.00 14,656.20 13,025,500,160 245,793,634,125

3 28-Dec-17 15,864.10 15,888.40 13,937.30 14,606.50 12,336,499,712 244,929,825,575

4 27-Dec-17 16,163.50 16,930.90 15,114.30 15,838.50 12,487,600,128 265,559,172,050

In [375]:

bit2017['Open'] = bit2017['Open'].astype(str).str.replace(',','')

bit2017['Low'] = bit2017['Low'].astype(str).str.replace(',','')

bit2017['High'] = bit2017['High'].astype(str).str.replace(',','')

bit2017['Close'] = bit2017['Close'].astype(str).str.replace(',','')

bit2017['Volume'] = bit2017['Volume'].astype(str).str.replace(',','')

bit2017['Market\_Cap'] = bit2017['Market\_Cap'].astype(str).str.replace(',','')

In [376]:

bit2017.head()

Out[376]:

Date Open High Low Close Volume Market\_Cap

0 31-Dec-17 12897.70 14377.40 12755.60 14156.40 12136299520 237465823980

1 30-Dec-17 14681.90 14681.90 12350.10 12952.20 14452599808 217239803085

2 29-Dec-17 14695.80 15279.00 14307.00 14656.20 13025500160 245793634125

3 28-Dec-17 15864.10 15888.40 13937.30 14606.50 12336499712 244929825575

4 27-Dec-17 16163.50 16930.90 15114.30 15838.50 12487600128 265559172050

In [377]:

bit2017.Open = bit2017.Open.astype(float)

bit2017.Low = bit2017.Low.astype(float)

bit2017.High = bit2017.High.astype(float)

bit2017.Close = bit2017.Close.astype(float)

bit2017.Volume = bit2017.Volume.astype(float)

bit2017.Market\_Cap = bit2017.Market\_Cap.astype(float)

In [421]:

c = bit2017.Open

d = bit2017.Close

plt.scatter(c,d, color = 'silver')

plt.xlabel('Bitcoin Open Price per day, 2017')

plt.ylabel('Bitcoin Close price per day, 2017')

plt.show()

In [424]:

def slope\_intercept(x\_val, y\_val):

x = np.array(x\_val)

y = np.array(y\_val)

m = (((np.mean(x)\*np.mean(y)) - np.mean(x\*y)) /

((np.mean(x)\*np.mean(x)) - np.mean(x\*x)))

m = round(m,2)

b = (np.mean(y) - np.mean(x)\*m)

b = round(b,2)

return m,b

slope\_intercept(c,d)

Out[424]:

(1.0, 35.39)

In [425]:

m,b = slope\_intercept(c,d)

reg\_line17 = [(m\*i)+b for i in c]

In [426]:

plt.scatter(c,d, color = 'silver')

plt.plot(c,reg\_line17, color='black')

plt.xlabel('Bitcoin open price per day, 2017')

plt.ylabel('Bitcoin Close price per day, 2017')

plt.title("Regression line for 2017")

plt.show()

In [427]:

rmse(d,reg\_line17)

Out[427]:

402.17284841222187

In [428]:

c = bit2017.Open.values.reshape(-1,1)

d = bit2017.Close.values.reshape(-1,1)

cTrain, cTest, dTrain, dTest = train\_test\_split(c, d, test\_size = 1/4, random\_state = 0)

In [429]:

plt.scatter(cTrain, dTrain, color = 'silver')

plt.plot(cTrain, linearregressor.predict(cTrain), color = 'black')

plt.title('Open vs. Close (Training set) 2017')

plt.xlabel('Bitcoin Open price per day, 2017')

plt.ylabel('Bitcoin Close price per day, 2017')

plt.show()

In [430]:

plt.scatter(cTest, dTest, color = 'silver')

plt.plot(cTrain, linearregressor.predict(cTrain), color = 'black')

plt.title('Open vs. Close (Test set) 2017')

plt.xlabel('Bitcoin Open price per day, 2017')

plt.ylabel('Bitcoin Close price per day, 2017')

plt.show()

In [386]:

bit2017.mean(axis = 0)

Out[386]:

Open 3.970645e+03

High 4.144688e+03

Low 3.811708e+03

Close 4.006034e+03

Volume 2.382867e+09

Market\_Cap 6.645752e+10

dtype: float64

In [387]:

bit2018 = pd.read\_csv('2018bitcointrading.csv')

bit2018.head()

Out[387]:

Date Open High Low Close Volume Market\_Cap

0 31-Dec-18 3,866.84 3,868.74 3,725.87 3,742.70 4,661,840,806 65,331,499,158

1 30-Dec-18 3,822.38 3,901.91 3,797.22 3,865.95 4,770,578,575 67,475,512,827

2 29-Dec-18 3,932.49 3,963.76 3,820.41 3,820.41 4,991,655,917 66,672,244,158

3 28-Dec-18 3,653.13 3,956.14 3,642.63 3,923.92 5,631,554,348 68,471,837,969

4 27-Dec-18 3,854.69 3,874.42 3,645.45 3,654.83 5,130,222,366 63,768,757,101

In [388]:

bit2018.Open = bit2018.Open.astype(str).str.replace(',','')

bit2018.Close = bit2018.Close.astype(str).str.replace(',','')

bit2018.High = bit2018.High.astype(str).str.replace(',','')

bit2018.Low = bit2018.Low.astype(str).str.replace(',','')

bit2018.Volume = bit2018.Volume.astype(str).str.replace(',','')

bit2018.Market\_Cap = bit2018.Market\_Cap.astype(str).str.replace(',','')

In [389]:

bit2018.Open = bit2018.Open.astype(float)

bit2018.Low = bit2018.Low.astype(float)

bit2018.High = bit2018.High.astype(float)

bit2018.Close = bit2018.Close.astype(float)

bit2018.Volume = bit2018.Volume.astype(float)

bit2018.Market\_Cap = bit2018.Market\_Cap.astype(float)

In [431]:

p = bit2018.Open

q = bit2018.Close

plt.scatter(p,q, color = 'green')

plt.xlabel('Bitcoin Open Price per day 2018')

plt.ylabel('Bitcoin Close price per day 2018')

plt.show()

In [432]:

slope\_intercept(p,q)

Out[432]:

(0.98, 123.3)

In [433]:

m,b = slope\_intercept(p,q)

reg\_line18 = [(m\*i)+b for i in p]

In [434]:

plt.scatter(p,q, color = 'green')

plt.plot(p,reg\_line18, color='black')

plt.xlabel('Bitcoin open price per day, 2018')

plt.ylabel('Bitcoin low price per day, 2018')

plt.title("Regression line for 2018")

plt.show()

In [435]:

rmse(q,reg\_line18)

Out[435]:

381.20860431002865

In [395]:

p = bit2018.Open.values.reshape(-1,1)

q = bit2018.Close.values.reshape(-1,1)

pTrain, pTest, qTrain, qTest = train\_test\_split(p, q, test\_size = 1/2, random\_state = 0)

In [436]:

plt.scatter(pTrain, qTrain, color = 'green')

plt.plot(pTrain, linearregressor.predict(pTrain), color = 'black')

plt.title('Open vs. Close (Training set) 2018')

plt.xlabel('Bitcoin Open price per day, 2018')

plt.ylabel('Bitcoin Close price per day, 2018')

plt.show()

In [437]:

plt.scatter(pTest, qTest, color = 'green')

plt.plot(pTrain, linearregressor.predict(pTrain), color = 'black')

plt.title('Open vs. Close (Test set) 2018')

plt.xlabel('Bitcoin Open price per day, 2018')

plt.ylabel('Bitcoin Close price per day, 2018')

plt.show()

In [398]:

bit2018.mean(axis = 0)

Out[398]:

Open 7.601019e+03

High 7.787979e+03

Low 7.350748e+03

Close 7.572299e+03

Volume 6.063552e+09

Market\_Cap 1.292673e+11

dtype: float64