Abstract

This is about my python programming in studying the fluctuations of stocks. This primary purpose of this code is analyze the trends of stock prices (adjusted closed price) across the end of the year and the beginning of the year through the months December and January. It uses MSE or minimum mean error to estimate the trend line of the stock.

Stock Price Analysis - draft

Claire Gong

December 2020

Contents

1 Introduction

In general, predicting the prices of stocks is not reliable. However, there may be some trends for stock price variation across two consecutive months across the year. I have created this project in hopes of finding any common trends across several stocks and years.

2 Formulation

2.1 MSE solution for linear equation

We want to create a program to determine if one should or shouldn't buy or sell a stock. This is determined by being given numerous (N) observations which are points of dates and adjusted close prices:

$$(t_1, \hat{y}_1)(t_2, \hat{y}_2) \ldots (t_N, \hat{y}_N)$$

In order to figure out if the stock has increased or decreased, we want to create a line of best fit of all of the points using the equation MSE (minimum mean square error).

$$y_t = m \cdot t + b \tag{1}$$

We need to find m and b such that the MSE equation is minimized.

$$\text{MSE} = \min_{m,b} \frac{1}{N} \sum_{t=1}^{N} (y_t - \hat{y}_t)^2$$

In order to solve for m and b, we need to derive the MSE equation versus both m and b.

2.1.1 Derivative for m

$$\frac{\partial \frac{1}{N} \sum_{t=1}^{N} (y_t - \hat{y}_t)^2}{\partial m} = \frac{1}{N} \sum_{t=1}^{N} 2(y_t - \hat{y}_t) \frac{dy_t}{dm} = \frac{1}{N} \sum_{t=1}^{N} 2(y_t - \hat{y}_t) \times t \tag{2}$$

2.1.2 Derivative for b

$$\frac{\partial \frac{1}{N} \sum_{t=1}^{N} (y_t - \hat{y}_t)^2}{\partial b} = \frac{1}{N} \sum_{t=1}^{N} 2(y_t - \hat{y}_t) \frac{dy_t}{db} = \frac{1}{N} \sum_{t=1}^{N} 2(y_t - \hat{y}_t) \tag{3}$$

2.1.3 Solving for m and b

Setting EQ-2 and EQ-3 to 0. We first need to rearrange m and isolate b in their respective equations in order to do substitution. Here is m rearranged:

$$m \sum_{t=1}^{N} t^2 - \sum_{t=1}^{N} \hat{y}_t t + b \sum_{t=1}^{N} t = 0 \tag{4}$$

Here is b isolated:

$$b = -\frac{m}{N} \sum_{t=1}^{N} t + \frac{1}{N} \sum_{t=1}^{N} \hat{y}_t \tag{5}$$

Substituting b for m (see appendix):

$$m \left[\sum_{t=1}^{N} t^2 - \frac{1}{N} \left(\sum_{t=1}^{N} t \right)^2 \right] = \sum_{t=1}^{N} \hat{y}_t t - \frac{1}{N} \sum_{t=1}^{N} \hat{y}_t \sum_{t=1}^{N} t \tag{6}$$

Final equation of m isolated:

$$m = \frac{\sum_{t=1}^{N} \hat{y}_t t - \frac{1}{N} \sum_{t=1}^{N} \hat{y}_t \sum_{t=1}^{N} t}{\sum_{t=1}^{N} t^2 - \frac{1}{N} \left(\sum_{t=1}^{N} t \right)^2} \tag{7}$$

Final equation of b isolated. Substituting m EQ-7 into b EQ-5:

$$b = -\frac{\sum_{t=1}^{N} t \left(\sum_{t=1}^{N} \hat{y}_t t - \frac{1}{N} \sum_{t=1}^{N} \hat{y}_t \sum_{t=1}^{N} t \right)}{N \sum_{t=1}^{N} t^2 - \left(\sum_{t=1}^{N} t \right)^2} + \frac{1}{N} \sum_{t=1}^{N} \hat{y}_t \tag{8}$$

2.2 Normalization of MSE solution

While directly comparing the slope across different companies, one may run into intensive cognitive load. To combat this, normalization is required. Let P_B be the price of the beginning time and P_E be the price of the end time.

$$\Delta P = P_E - P_B \tag{9}$$

In order to normalize ΔP, we need to divide it by P_B.

$$\%\Delta P = \frac{P_E - P_B}{P_B} = \frac{P_E}{P_B} - 1 \tag{10}$$

Referring to EQ-1, we can substitute in EQ-1 into EQ-10. Note that P_B is equal to b (the y-intercept), P_E is equal to y, and N is equal to t.

$$\%\Delta P = \frac{P_E}{P_B} - 1 = \left[\frac{y}{b} - 1 \right] = \frac{mt}{b} = \frac{mN}{b} \tag{11}$$

3 Python program

3.1 Python code outline

Exercise: answering the question: should I sell stocks at December 31?

1. For one stock
   - DEC stock gain
   - JAN stock gain
   - the difference
2. Do (1) and average for last 5 years (function call)
3. Do (2) and average for 10 stocks
4. What's the conclusion?

3.2 Library implementation

```
#generation of testing data
m = 0.1
b = 5
n = 100
yhatlist = []
ylist = []
for i in range(0,n):
    noise = random.uniform(-1,1)
    y = m*(i+1)+b
    yhat = y + noise
    yhatlist.append(yhat)
    ylist.append(y)
```

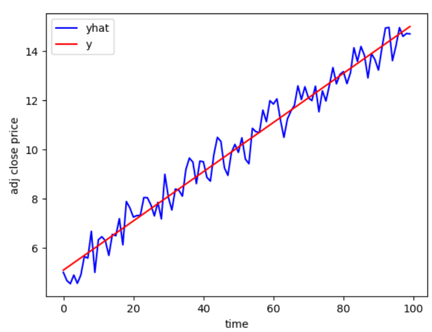

Figure 1: Simulation Result

4 Experimental results

Experimental results will focus on studying the output of the Python code which is a scatter plot depicting the normalized slope values determined by MSE from December and January of the given year. Let the $\%\Delta P(\cdot)$ be the normalized slope of month $\cdot$, where $\cdot$ is either December or January. The following table is a representation of a graph where in each specific quadrant, it shows the condition and hypothetical action.

$\%\ \Delta P(Dec)\downarrow$, $\%\Delta P(Jan)\uparrow$ buy	$\%\ \Delta P(Dec)\uparrow$, $\%\Delta P(Jan)\uparrow$ no change
$\%\ \Delta P(Dec)\downarrow$,$\%\ \Delta P(Jan)\downarrow$ sell	$\%\ \Delta P(Dec)\uparrow$, $\%\Delta P(Jan)\downarrow$ sell

For example, see Fig-4 for Microsoft's stock performance of December and January, the points are mainly in the first and fourth quadrant. This means that the user should hypothetically either choose to buy/sell/keep (no change) or sell their stocks, depending on the user's risk tolerance.

4.1 Plot of two consecutive months

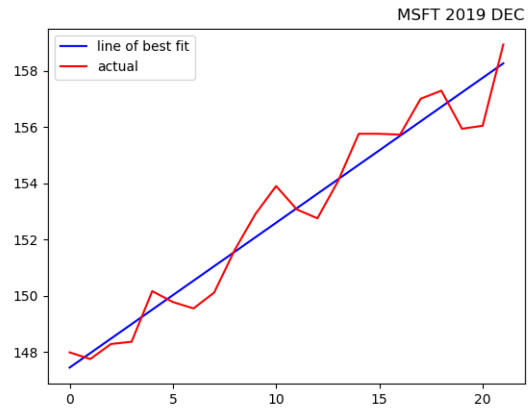

Figure 2: Microsoft December 2019 stock

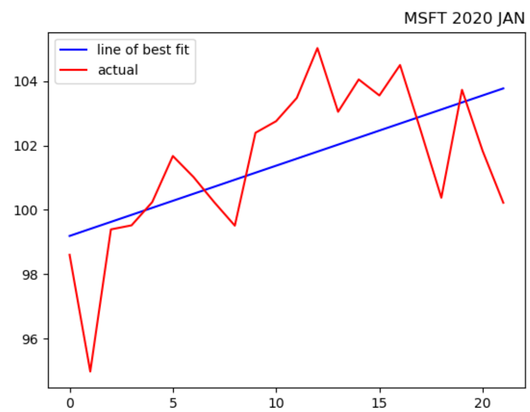

Figure 3: Microsoft January 2020 stock

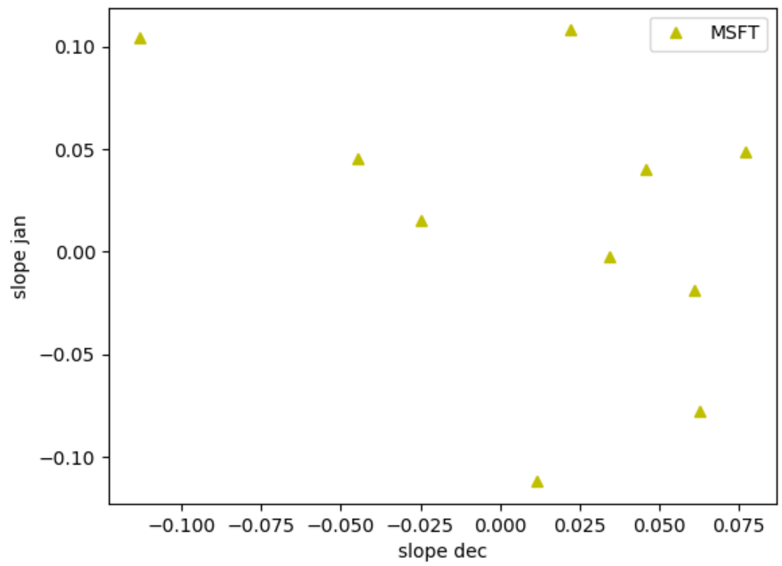

Figure 4: Microsoft scatter plot 2010-2020

4.2 Scatter plot for MSFT

4.3 Other scatter plots

Here are the scatter for some other companies:

5 Conclusion

- We did not observe a distinct pattern in the scatter plot data.
- However, we did notice that most of the time, if it is the December plot is negative, most likely the January plot will be positive, meaning that one may choose to invest in December to receive some gains.
- If December is positive, January generally can be either positive or negative.

6 Appendix

An extra step leading to EQ-6

$$m\sum_{t=1}^{N} t^2 - \sum_{t=1}^{N} \hat{y}_t t + \left(-\frac{m}{N}\sum_{t=1}^{N} t + \frac{1}{N}\sum_{t=1}^{N} \hat{y}_t\right)\sum_{t=1}^{N} t = 0$$

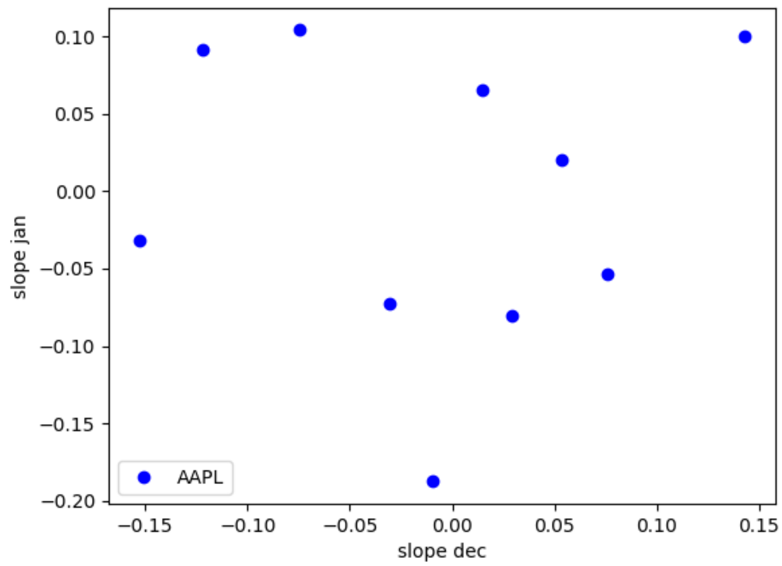

Figure 5: Apple scatter plot 2010-2020

References

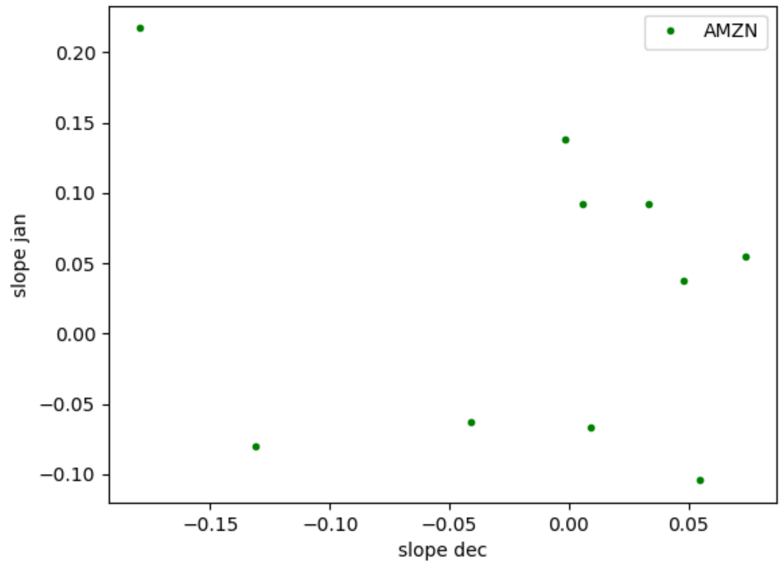

Figure 6: Amazon scatter plot 2010-2020

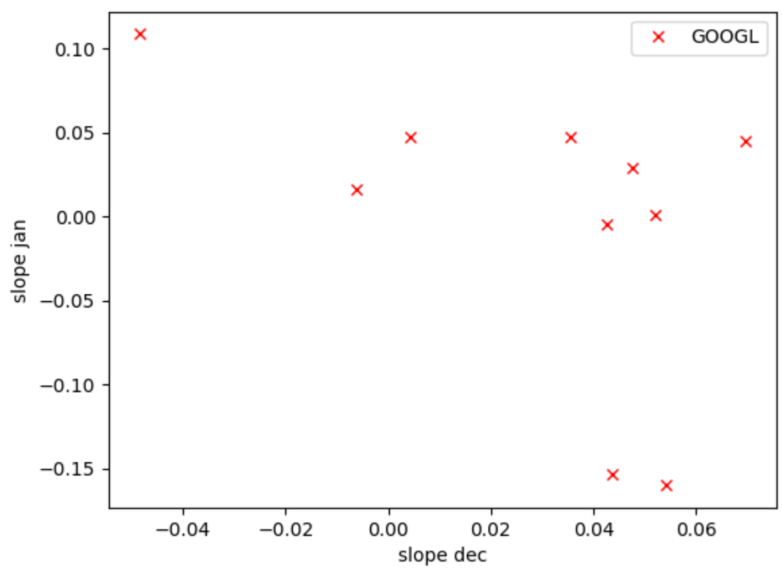

Figure 7: Google scatter plot 2010-2020

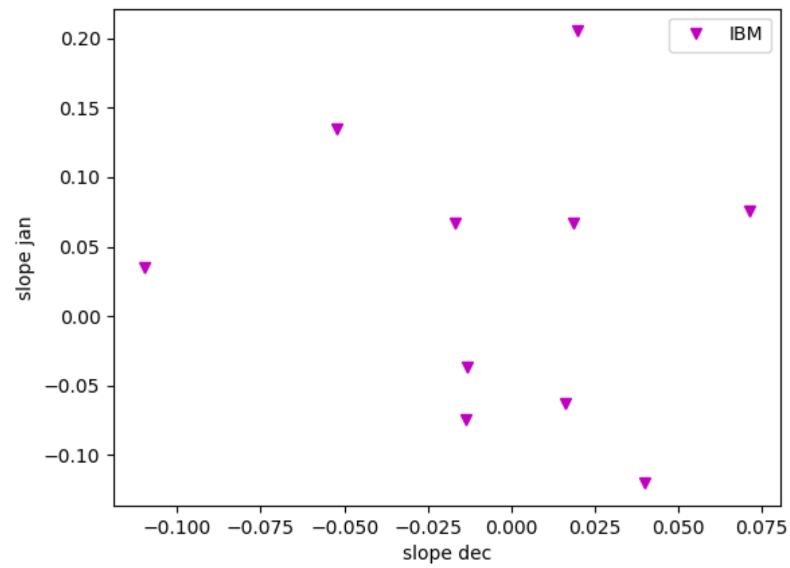

Figure 8: IBM scatter plot 2010-2020

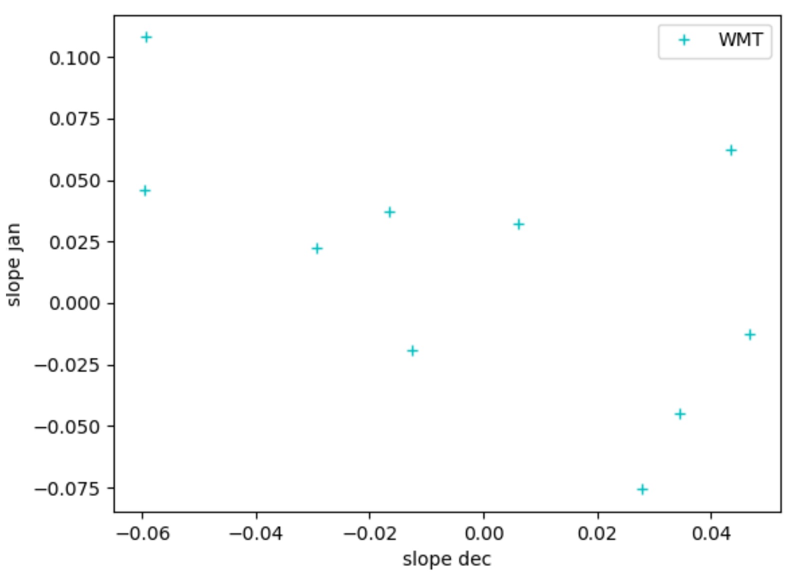

Figure 9: Walmart scatter plot 2010-2020

Figure 10: The Universe