CQF Final project report
Pair Trading strategy Design and Backtest
Yang Du
2021 Jan 18

# Introduction:

The history of pair trading can be traced back to mid-1980 when a group of analyst researchers employed in Morgan Stanley found a statistical arbitrage opportunity from a combination of long and short position in a pair of stocks that are highly positively correlated to each other. Ever since then, pair trading has become one of the most classic and well-known trading strategies adopted by a numbers of hedge funds and investing company. Even nowadays, despite the continuous downward trend in its profitability, pair trading can still have a strong performance during the prolonged turbulence e.g., during the financial crisis. In this project, the pair trading performed on Facebook stock and Alibaba stock will be demonstrated.

Facebook, the F of the famous FAANG, is one of the largest global social media company founded by Mark Zuckerberg. It possesses over 130-billion-dollar assets. Facebook has been the subject of numerous controversies, often involving user privacy (as with the Cambridge Analytica data scandal), political manipulation (as with the 2016 U.S. elections), mass surveillance, psychological effects such as addiction and low self-esteem, and content such as fake news, conspiracy theories, copyright infringement, and hate speech. Commentators have accused Facebook of willingly facilitating the spread of such content and also exaggerating its number of users in order to appeal to advertisers. As of November 18, 2020, Alexa Internet ranks Facebook 6 in global internet usage. (from Wikipedia: https://en.wikipedia.org/wiki/Facebook)


Alibaba, the ‘Amazon’ in China. Alibaba is the world's largest retailer and e-commerce company, and on the list of largest Internet companies. In 2020, it was also rated as the fifth largest artificial intelligence company. It is also one of the biggest venture capital firms, and one of the biggest investment corporations in the world. The company hosts the largest B2B (Alibaba.com), C2C (Taobao), and B2C (Tmall) marketplaces in the world. It has been expanding into the media industry, with revenues rising by triple percentage points year after year. It also set the record on the 2018 edition of China's Singles' Day, the world's biggest online and offline shopping day (from Wikipedia: https://en.wikipedia.org/wiki/Alibaba_Group)

# Material, Method and Terminology:

There are many different types of pair trading strategy. The pair trading implemented in this project is a market neutral strategy that focus on statistical arbitrage. The basic rule to trade in equity market is to long the undervalued ones and short the overpriced ones. However, it is a truth universally acknowledged that stock price is stochastic and non-stationery and regression one stock price on another is meaningless. The true price of a stock is really hard to predict in real life and this is where pair trading steps in. Pair trading tends to solve this problem by transform it into relative pricing. Although the price itself may be hard to predict, the difference between two highly correlated stock prices can be stationary. According to the arbitrage pricing theory aka APT, two stocks that share similar characteristics also tends to behave in more or less the same way in prices. Therefore, the difference, or to put it in a more financial term, the spread between prices, is believed to be stationary and exhibit a mean reverting phenomenon on the long run. For a simple demonstration, let's define

$$\text{Spread} = \text{Price\_A} - \beta * \text{Price\_B}$$

(*where Price_A; Price_B are the prices of two correlated equities, and beta $\beta$ is just some coefficient)

Note that the Spread is a linear combination of Price_A and Price_B. If Spread suddenly increases drastically, we have the following two cases:

$$\text{Spread} \Uparrow = \text{Price\_A} \Uparrow - \beta * \text{Price\_B}$$

Or

$$\text{Spread} \Uparrow = \text{Price\_A} - \beta * \text{Price\_B} \Downarrow$$

In the first scenario the Price_A is overvalued while in the second scenario Price_B is undervalued. Since the spread is mean reverting, at some point in the future, the spread will fall back to its equilibrium, equivalently Price_A will fall in scenario 1 or Price_B will rise in scenario 2 in the future. Similarly, if the spread shrinks, then we will face:

$$\text{Spread} \Downarrow = \text{Price\_A} \Downarrow - \beta * \text{Price\_B}$$

Or

$$\text{Spread} \Downarrow = \text{Price\_A} - \beta * \text{Price\_B} \Uparrow$$

The idea behind the arbitrage is the same just in the opposite manner. This is a just simplified illustration of how pair trading can make the arbitrage. More details will be given in the upcoming section.

As the name suggest, a pair trading strategy involves a pair of instruments, it could be a pair of equities, futures, interest rate products etc. As long as high correlation appears between the prices of the instruments. In this project, Facebook stock and Alibaba stock are used to construct my pair. The close prices for these two companies from 2018 to 2020 are plotted below.

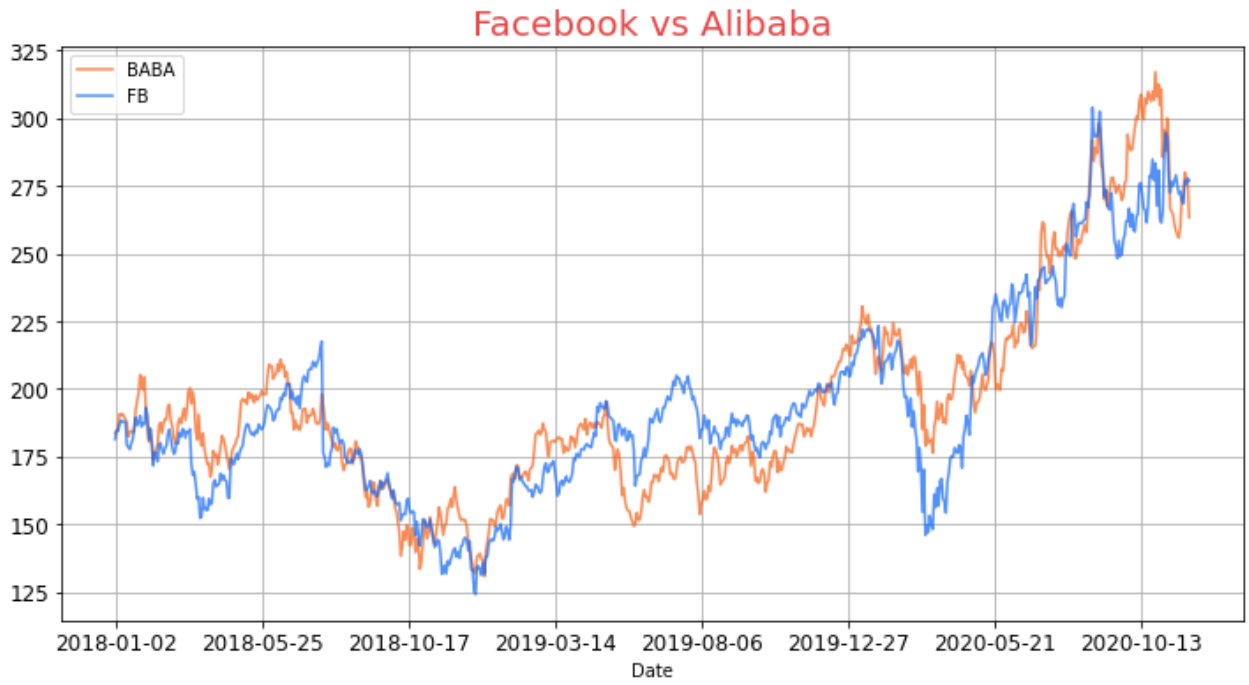



*Figure 1.1, Facebook vs Alibaba close price

The co-movement of two stocks is obvious in the plot. They move in the same direction with almost same patterns at different scales. Visually, they may produce stationary spread. In order to make it more obvious, the plot of their spread is also plotted below


*Figure 1.2, Facebook vs Alibaba close price spread

Figure 1.2 illustrates the spread between Facebook stock and Alibaba stock over time. Although the line looks chaotic at the first sight, it actually oscillates around the horizontal line price = 0. Note that I am not confirming that price = 0 is its equilibrium. The key point to take away here is that the line is exhibiting some mean-reverting quality meaning it doesn't just go up or down all the way through time but instead, it is being up and down repeatedly and always come across the equilibrium line.

**Cointegration**

Next, what the above paragraph describes is the concept of cointegration. If a set of time series that are integrated of order(d) and they linear combination is of integrated order (less than d), then we conclude that these series are cointegrated. In this case, both Facebook and Alibaba have order of I(1), and differencing them will lead us to an I(0) spread which is a necessary but insufficient condition for a series to be stationary. The differencing process will lead us to another problem: the loss of long run information, which partially explains why the cointegration test. The result of cointegration test tells us if the difference/linear combination of non-stationary variables is stationary so that we can move on to regression without getting spurious results. In other words, the existence of cointegration confirms that long run parameters are indeed consistent. For the pair trading strategy in this project, the stocks must be cointegrated in order to produce stationary spread.

In python, we can simply evaluation the cointegration of two equities using the codes:

```
In [16]: from statsmodels.tsa.stattools import coint
         score, p_value, _ = coint(df.BABA, df.FB)

In [17]: p_value
Out[17]: 0.005866356426351822
```

Since p_value is way lower than 0.05, thus we can reject the null hypothesis which claims there is not cointegration and conclude that these 2 stocks are highly likely to be cointegrated.

**Engle Granger Procedure**

There are three main methods for testing cointegration, namely, Engle Granger; Johansen Test; Phillips–Ouliaris cointegration test. This project will focus on Engle Granger procedure only. Engle Granger consist of two steps:

Step1:

Let Xt and Yt be two time series of I(1), first, fit them into a linear regression:

$$\mathbf{Y_t} = \hat{\beta}\ \mathbf{X_t} + \hat{\mu}\ + \varepsilon_t$$

$$\widehat{\varepsilon_t} = \mathbf{Y_t} - \hat{\beta} * \mathbf{X_t} - \hat{\mu}$$

Then use the ADF test examine the stationarity of $\widehat{\varepsilon_t}$.

Step2:

Construct the error correction equation:

$$\Delta \mathbf{Y_t} = \Phi \Delta \mathbf{X_t} - (1 - \alpha)\,\widehat{\varepsilon_{t-1}}$$

$$\Delta \mathbf{Y_t} = \Phi \Delta \mathbf{X_t} - (1 - \alpha)(\mathbf{Y_{t-1}} - \hat{\beta} * \mathbf{X_{t-1}} - \hat{\mu})$$

And confirm the significance for $(1 - \alpha)$.

**ADF test:**

The Augmented Dicky Fuller test tests the existence of unit roots given a time series. This test is implemented because we want to ensure our estimated $\widehat{\varepsilon_t}$ is stationary. In this project, unlike original form of the ADF equation, in order to improve the accuracy of ADF test, the trend term $\varphi * t$ is removed from the following equation.

$$\Delta\varepsilon_t = \varphi\varepsilon_{t-1} + \varphi_{aug1}\Delta\varepsilon_{t-1} + \mu + \varepsilon_t$$

Where:

$\Delta\varepsilon_t$ is the difference of ε between t and t-1

$\Delta\varepsilon_{t-1}$ is the lagged ε

$\Delta\varepsilon_{t-1}$ is difference of ε between t -1 and t-2

$\mu$ is a constant term

And if coefficient $\varphi$ is significantly different from 0, then the null hypothesis that a unit root is present in this time series is rejected. Thus, the time series is stationary.

**Critical values for Dickey–Fuller *t*-distribution.**

| | Without trend | | With trend | |
|---|---|---|---|---|
| Sample size | 1% | 5% | 1% | 5% |
| T = 25 | −3.75 | −3.00 | −4.38 | −3.60 |
| T = 50 | −3.58 | −2.93 | −4.15 | −3.50 |
| T = 100 | −3.51 | −2.89 | −4.04 | −3.45 |
| T = 250 | −3.46 | −2.88 | −3.99 | −3.43 |
| T = 500 | −3.44 | −2.87 | −3.98 | −3.42 |
| T = ∞ | −3.43 | −2.86 | −3.96 | −3.41 |

(*Figure 1.3)

In addition, the t distribution that is being used in ADF test is different from the one we use for t test. See Figure 1.3 above.

**OU process**

The Ornstein Uhlenbeck process can be used to evaluate assess the quality of the mean reversion of the spread. OU process satisfies three conditions: 1. It is a Gaussian process; 2. it is a Markov process; 3. It is temporarily homogeneous.

And usually it takes the form:

$$d\boldsymbol{\varepsilon_t} = -\Theta(\boldsymbol{\varepsilon_t} - \mu)\,dt + \sigma_{ou} d\mathbf{X_t}$$

where $d\mathbf{X_t}$ is a Brownian motion

This process is put in consideration because it generates mean reversion and the solution to this stochastic equation is:

$$\boldsymbol{\varepsilon_{t+\tau}} = (1 - e^{-\Theta\tau})\mu + e^{-\Theta\tau}\boldsymbol{\varepsilon_t} + \varpi$$

Compare to an AR():

$$\boldsymbol{\varepsilon_{t+\tau}} = C + B\boldsymbol{\varepsilon_t} + \varpi$$

We have

$$C = (1 - e^{-\Theta\tau})\mu$$

$$B = e^{-\Theta\tau}$$

Rearrange the variables and we will have:

$$\Theta = -\frac{\ln(B)}{C^{\tau}}$$

$$\mu = \frac{C}{1-B}$$

$$\sigma_{eq} = \sqrt{\frac{\Sigma}{1-e^{-2\Theta\tau}}}$$

**Trading Strategy**

After acquiring $\mu$ and $\sigma_{eq}$, now we can build our pair trading strategy. Strategy is fairly easy to construct. First, we construct two bounds

Upper bound: $\mu + Z * \sigma_{eq}$

Lower bound: $\mu - Z * \sigma_{eq}$

Z is a constant that is used to adjust the spread between the bounds.

If the residual $\varepsilon_t = Y_t - \hat{\beta} * X_t - \hat{\mu}$ rise above upper bound, short 100% $Y_t$ and long $\hat{\beta}$ $X_t$;

If the residual $\varepsilon_t = Y_t - \hat{\beta} * X_t - \hat{\mu}$ rise below lower bound, long 100% $Y_t$ and short $\hat{\beta}$ $X_t$;

Whenever residual $\varepsilon_t$ come across the equilibrium $\mu$, we clear our position.

# Results:

In this section, I will represent the result of my codes. Please view the appendix section at the end of this report for complete codes and derivations for algorithms.

Data inspection

(*Figure

| Date | BABA | FB |
|---|---|---|
| 2018-01-02 | 183.649994 | 181.419998 |
| 2018-01-03 | 184.000000 | 184.669998 |
| 2018-01-04 | 185.710007 | 184.330002 |
| 2018-01-05 | 190.699997 | 186.850006 |
| 2018-01-08 | 190.330002 | 188.279999 |
| 2018-01-09 | 190.800003 | 187.869995 |
| 2018-01-10 | 189.789993 | 187.839996 |
| 2018-01-11 | 188.750000 | 187.770004 |
| 2018-01-12 | 187.789993 | 179.369995 |
| 2018-01-16 | 182.399994 | 178.389999 |

(*Figure

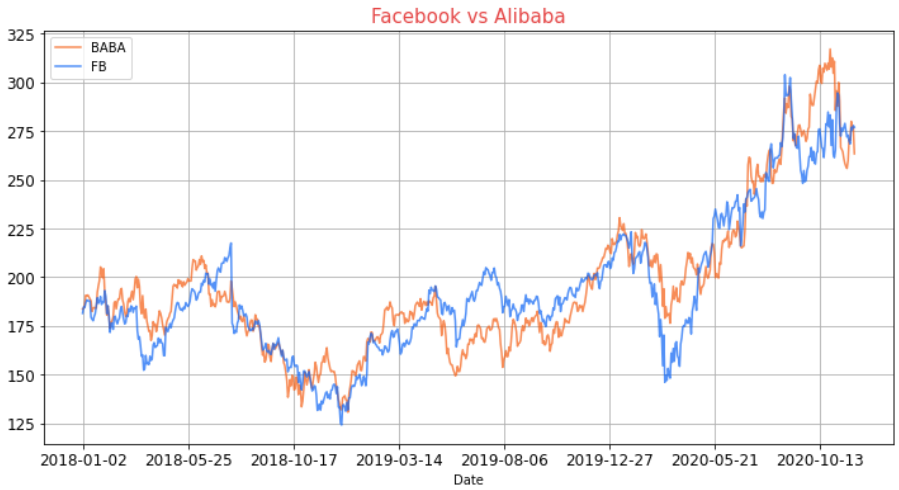



Figure 2.1 is the head columns of the datafile. It is made up of 2 columns:

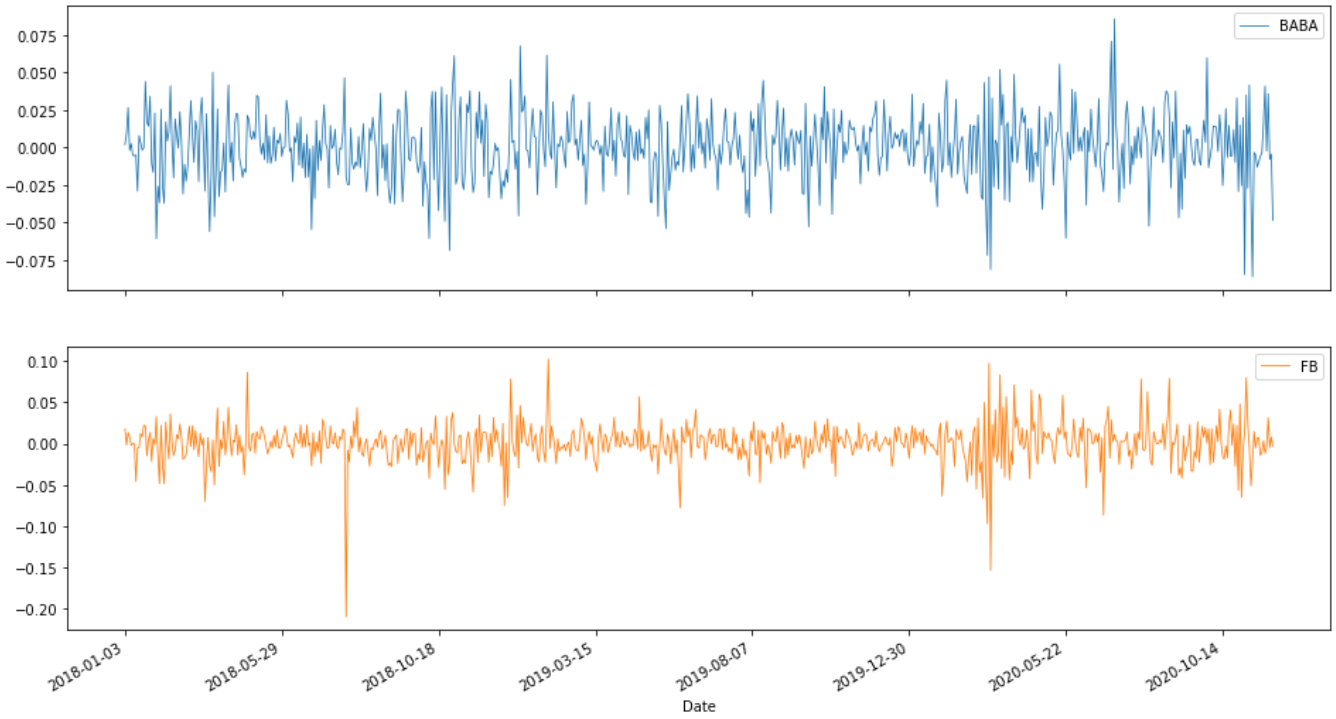
- BABA the daily close price for Alibaba
- FB the daily close price for Facebook

These prices are all every close to each other in terms of their values. They also behave in a very similar way. Not only they follow nearly the same trend, but their prices also actually come across each other like two snakes. Although both Facebook and Alibaba are well known tech companies, it is still surprising to see these 2 stocks has such high level of accordance on their price movements given their different company background.


(*figure 2.3)

As I mentioned previously, we need a stationary residual time series in order to make pair trading work. At this point, it looks like residuals are stationary from the plot, but more tests are needed to ensure the existence of stationarity.

Before we move to Engle Grander, let’s take a minute to look at the returns of both stocks.




(*Figure 2.4, stock returns in graph)

| Date | BABA | FB |
|---|---|---|
| 2018-01-03 | 0.001904 | 0.017756 |
| 2018-01-04 | 0.009251 | -0.001843 |
| 2018-01-05 | 0.026515 | 0.013579 |
| 2018-01-08 | -0.001942 | 0.007624 |
| 2018-01-09 | 0.002466 | -0.002180 |
| ... | ... | ... |
| 2020-11-23 | -0.002330 | -0.004720 |
| 2020-11-24 | 0.035817 | 0.031139 |
| 2020-11-25 | -0.008033 | -0.004814 |
| 2020-11-27 | -0.004475 | 0.008023 |
| 2020-11-30 | -0.048617 | -0.003028 |

(*Figure 2.5, numerical

Here we have the line plots for both stock returns. Both returns are of similar scales approximately from 0.1 to -0.05 if we don't consider outliers. They share a mean reverting property however in contrast, Alibaba's return, the blue line, is generally more volatile than Facebook and this might suggest that investors who are interested in Alibaba are more aggressive than those who are interested in Facebook in terms of trading intensity. Also, it is worth to notice that more people are engaged in trading market after the COVID 19 as both stocks delivered more volatility during the pandemic period. The phenomenon is accordant to economic sense since many businesses are broken, and people lose their jobs due to the lock down and social distancing policies. The market was facing severe liquidity problem. To solve that issue the Federal Reserve System printed a large amount of cash which reduced the interest rate. As a result, people move their money out of bank and invest.

Additionally, in case you are wondering why there is such a great drop on Facebook's return between 2018-05-29 and 2018-10.18. The exact date on which the drop happened was 2018-07-26, and the drop was attributed to the missing key metrics (such as revenue) in the second quarter report that Facebook released the day before.

## VAR

Next, let's run a VAR model on the returns. VAR stands for Vector Auto Regression. VAR is essentially a linear regression on the variable's current value (i.e. value at t) and lagged/past value (i.e. value at t-1; t-2; t-3…) in a vector form. The 'V', vector, of VAR means we can regress multiple variables on their past values simultaneously.

For example, VAR (1) on stock returns:

```
In [11]: model = VAR(returns, lag = 1)
         model.report()
```

Out[11]:

| | Estimate Coefficient | | SD of Estimate | | t-Statistic | |
|---|---|---|---|---|---|---|
| | BABA | FB | BABA | FB | BABA | FB |
| (Lag_1, BABA) | 0.045013 | -0.030688 | 0.043083 | 0.047057 | 1.044792 | -0.652155 |
| (Lag_1, FB) | -0.073990 | -0.099768 | 0.039143 | 0.042752 | -1.890267 | -2.333627 |
| constant | 0.000510 | 0.000652 | 0.000821 | 0.000897 | 0.620490 | 0.727090 |

This result basically says:

$$\mathrm{BABA}_t = 0.045013 * \mathrm{BABA}_{t-1} - 0.07399 * \mathrm{FB}_{t-1} + 0.00051$$
$$\mathrm{FB}_t = -0.030688 * \mathrm{BABA}_{t-1} - 0.099768 * \mathrm{FB}_{t-1} + 0.000652$$

Here what we have are two equations that numerically describe the relationship between current returns and past returns for each stock respectively. The question is are these equations true? In order to know how well this regression goes, we must check the p value of the estimated coefficients. Compare the p value for each coefficient to 0.05 which is 95% of confidence level and we have:

```
In [12]: model.p_value < 0.05
```

Out[12]:

| | BABA | FB |
|---|---|---|
| (Lag_1, BABA) | False | False |
| (Lag_1, FB) | False | True |
| constant | False | False |

<- which is not very promising as we can see 5 of 6 fail to reject the null hypothesis that the coefficient is significantly different from 0. Unfortunately results here are really bad. The above regression is meaningless and cannot be used to predict returns

**Information Criterion (IC)**

Another thing we want to include here is Information Criterion. Note in the previous regression, we only included lag 1 terms, in other words we solely consider the relationship between t and t-1 and as it turned out the regression is spurious. The question is, what if I include more past terms? What if we consider the relation between t, t-1, t-2, t-3.....will we get a better model? IC test tells you how many lagged terms you should include in your model in order to optimize your result. In this project, we only implement AIC and BIC score here. The derivations for these two can be found in the Appendix section. As we can see here, each lag is assigned an IC score and the lag that has the lowest value is the optimal lag. In this case, lag 1 is our best lag since it's got the lowest value in both AIC and BIC. This means having more lagged term in our model does not help model explain our returns any better. Since we have already known lag 1 VAR failed. It is reasonable to conclude that we can't use past returns to predict futures.

```
In [13]: model.IC(lag = 10,)
```

Out[13]:

| Lag | AIC | BIC |
|---|---|---|
| 1 | -15.339415 | -15.301785 |
| 2 | -15.332764 | -15.270047 |
| 3 | -15.331255 | -15.243451 |
| 4 | -15.321393 | -15.208503 |
| 5 | -15.312210 | -15.174233 |
| 6 | -15.308782 | -15.145718 |
| 7 | -15.306070 | -15.117920 |
| 8 | -15.309380 | -15.096142 |
| 9 | -15.303473 | -15.065149 |
| 10 | -15.300814 | -15.037403 |

**Engle Granger (FB - β*BABA)**

Engle Granger suggested the following two steps procedures for testing the existence of cointegration in a time series. First, we run a linear regression on the two stocks we have:

$$FB = \beta * BABA + constant + \varepsilon$$

The coefficient for BABA 0.864299, is our **βcoint** or the hedge ratio we will use in our trading.

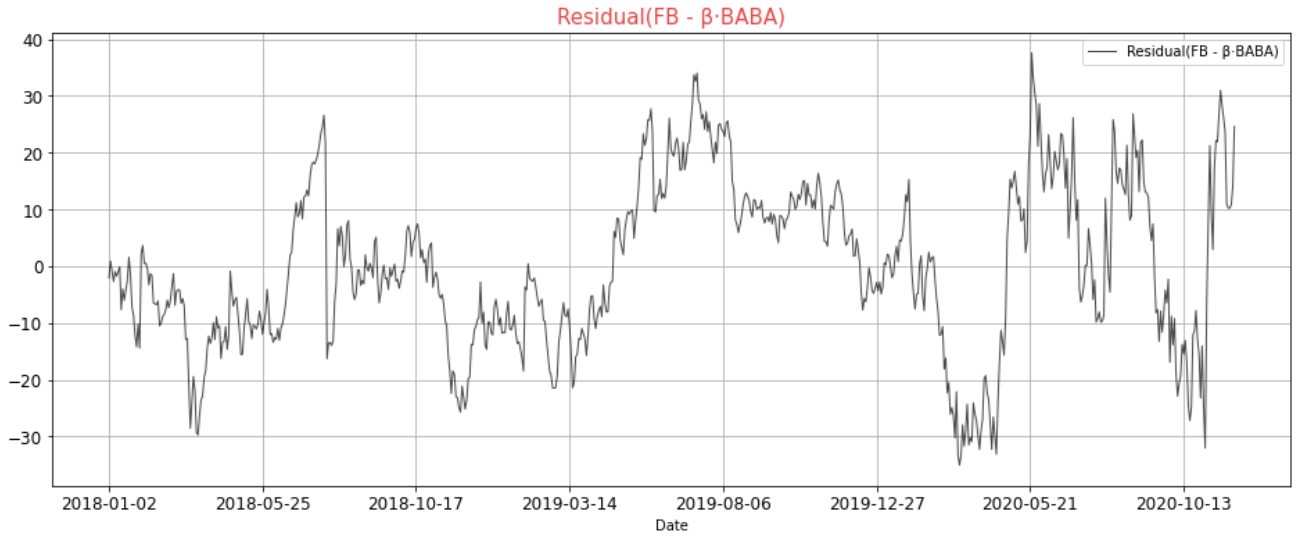
```
In [18]: model = LR(x = BABA, y = FB)
         model.report()
```

Out[18]:

| | Estimate Coefficient | SD of Estimate | t-Statistic |
|---|---|---|---|
| | FB | FB | FB |
| BABA | 0.864299 | 0.013810 | 62.585632 |
| constant | 24.744947 | 2.749676 | 8.999223 |

Next, extract the residuals of this model:

$$\varepsilon = FB - \beta * BABA - constant$$




(*figure 2.6)

run ADF test on the residuals.

Recall that the modified ADF has removed the trend

$$\Delta\varepsilon_t = \varphi\varepsilon_{t-1} + \varphi_{aug1}\Delta\varepsilon_{t-1} + \mu + \varepsilon_t$$

and φ need to be significantly different from 0 in order to claim stationarity.

```
In [22]: from utility import ADF

In [23]: adf_test = ADF(residuals)
         adf_test.report()

         The T-value is [-4.15444074] which is lower than -3.45
         [*]We reject the  H0  hypothesis of unit root. The residuals are stationary.
```

Out[23]:

| | Estimate Coefficient (ΔResidual(FB - β·BABA)) | SD of Estimate (ΔResidual(FB - β·BABA)) | t-Statistic (ΔResidual(FB - β·BABA)) |
|---|---|---|---|
| (Lag 1, Residual(FB - β·BABA)) | -0.047907 | 0.011532 | -4.154441 |
| (Lag 1, ΔResidual(FB - β·BABA)) | 0.030824 | 0.037194 | 0.828745 |
| constant | 0.030233 | 0.164179 | 0.184150 |

The number marked red is the t stats for φ. It is way lower than -3.44 which is within 1% significant level so we can now reject the null hypothesis and say the residuals are stationary.

Next, construct the error correction equation:

$$\Delta\mathbf{FB_t} = \Phi * \Delta\mathbf{BABA_t} - (1-\alpha)\,\widehat{\varepsilon_{t-1}}$$

$$\Delta\mathbf{FB_t} = \Phi * \Delta\mathbf{BABA_t} - (1-\alpha)(\mathbf{FB_{t-1}} - \hat{\beta} * \mathbf{BABA_{t-1}} - \hat{\mu})$$

And confirm the significance for $(1-\alpha)$.

In [24]:
```
Δy = pd.DataFrame(FB).diff().dropna().add_prefix('Δ')
Δx = pd.DataFrame(BABA).diff().dropna().add_prefix('Δ')

X = Δx.join(residuals.shift(1).dropna().add_prefix('(Lag 1, ').add_suffix(')'))

model = LR(X, Δy,add_const = False)
model.report()
```

Out[24]:

| | Estimate Coefficient | SD of Estimate | t-Statistic |
|---|---|---|---|
| | ΔFB | ΔFB | ΔFB |
| ΔBABA | 0.510556 | 0.034257 | 14.903664 |
| (Lag 1, Residual(FB - β·BABA)) | -0.040760 | 0.010647 | -3.828437 |

In [25]:
```
model.p_value < 0.05
```

Out[25]:

| | ΔFB |
|---|---|
| ΔBABA | True |
| (Lag 1, Residual(FB - β·BABA)) | True |

Now since p value for **(1 – α)** is less than 0.05, we have confirmed **(1 – α)** is significant.

**Engle Granger (BABA - β*FB)**

Next, repeat everything we did above expect this time we swap regressor and regressand.

First linear regression:

$$BABA = \beta * FB + constant + \varepsilon$$

The coefficient for FB 0.974831, is our **βcoint** or the hedge ratio we will use in our trading.

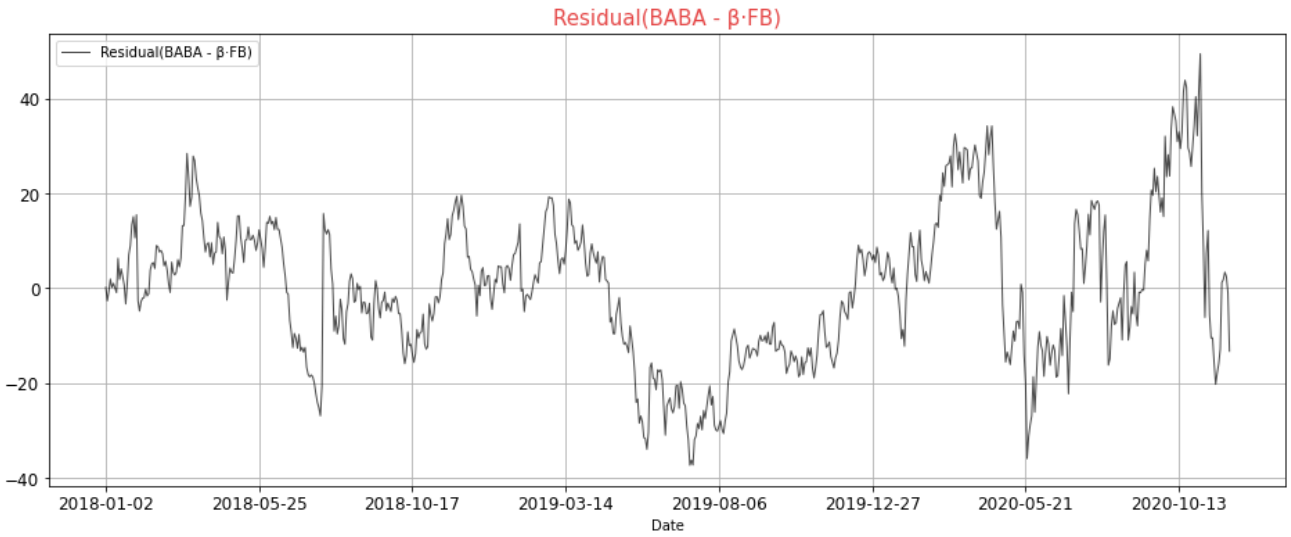
In [26]:
```
model = LR(x = FB, y = BABA,add_const = True)
model.report()
```

Out[26]:

| | Estimate Coefficient | SD of Estimate | t-Statistic |
|---|---|---|---|
| | BABA | BABA | BABA |
| FB | 0.974831 | 0.015576 | 62.585632 |
| constant | 6.633532 | 3.067732 | 2.162357 |

Next, extract the residuals of this model:

$$\varepsilon = BABA - \beta * FB - constant$$




run ADF test on the residuals.
Recall that the modified ADF has removed the trend

$$\Delta\varepsilon_t = \varphi\varepsilon_{t-1} + \varphi_{aug1}\Delta\varepsilon_{t-1} + \mu + \varepsilon_t$$

and φ need to be significantly different from 0 in order to claim stationarity.

```
In [29]: adf_test = ADF(residuals)
         adf_test.report()
```

```
The T-value is [-4.24848594] which is lower than -3.45
[*]We reject the  H0  hypothesis of unit root. The residuals are stationary.
```

Out[29]:

| | Estimate Coefficient (ΔResidual(BABA - β·FB)) | SD of Estimate (ΔResidual(BABA - β·FB)) | t-Statistic (ΔResidual(BABA - β·FB)) |
|---|---|---|---|
| **(Lag 1, Residual(BABA - β·FB))** | -0.047918 | 0.011279 | -4.248486 |
| **(Lag 1, ΔResidual(BABA - β·FB))** | 0.042808 | 0.037176 | 1.151512 |
| **constant** | -0.013579 | 0.170938 | -0.079441 |

The number marked red is the t stats for φ. It is also way lower than -3.44 which is within 1% significant level so we can now reject the null hypothesis and say the residuals are stationary.

Next, construct the error correction equation:

$$\Delta\mathbf{BABA_t} = \Phi * \Delta\mathbf{FB_t} - (1-\alpha)\,\widehat{\varepsilon_{t-1}}$$

$$\Delta\mathbf{BABA_t} = \Phi * \Delta\mathbf{FB_t} - (1-\alpha)(\mathbf{BABA_{t-1}} - \hat{\beta} * \mathbf{FB_{t-1}} - \hat{\mu})$$

And confirm the significance for $(1-\alpha)$.

In [30]:
```
Δy = pd.DataFrame(BABA).diff().dropna().add_prefix('Δ')
Δx = pd.DataFrame(FB).diff().dropna().add_prefix('Δ')

X = Δx.join(residuals.shift(1).dropna().add_prefix('(Lag 1, ').add_suffix(')'))

model = LR(X, Δy,add_const = False)
model.report()
```

Out[30]:

| | Estimate Coefficient | SD of Estimate | t-Statistic |
|---|---|---|---|
| | ΔBABA | ΔBABA | ΔBABA |
| ΔFB | 0.456371 | 0.030571 | 14.928102 |
| (Lag 1, Residual(BABA - β·FB)) | -0.031365 | 0.009481 | -3.308126 |

In [31]:
```
model.p_value < 0.05
```

Out[31]:

| | ΔBABA |
|---|---|
| ΔFB | True |
| (Lag 1, Residual(BABA - β·FB)) | True |

**Trading: Signal Generating**

There is not really a reason to favor one VECM over the other one since residuals in both VECMs are stationery and stats are also really close to each other. So, I will just use **FB - β*BABA** then.

First recall the OU process:

$$\varepsilon_{t+\tau} = C + B\varepsilon_t + \varpi$$

$$C = (1 - e^{-\Theta\tau})\mu$$
$$B = e^{-\Theta\tau}$$

Rearrange the variables and we will have:

$$\Theta = -\frac{\ln(B)}{\tau}$$

$$\mu = \frac{C}{1 - B}$$

$$\sigma_{eq} = \sqrt{\frac{\Sigma}{1 - e^{-2\Theta\tau}}}$$

According to these process, first we need to run a VAR(1) on the residuals.

```
In [61]: model = VAR(residuals, lag = 1)
         model.report()
```

Out[61]:

| | Estimate Coefficient Residual(FB - β·BABA) | SD of Estimate Residual(FB - β·BABA) | t-Statistic Residual(FB - β·BABA) |
|---|---|---|---|
| (Lag_1, Residual(FB - β·BABA)) | 0.95686 | 0.011287 | 84.775016 |
| constant | 0.05061 | 0.172184 | 0.293931 |

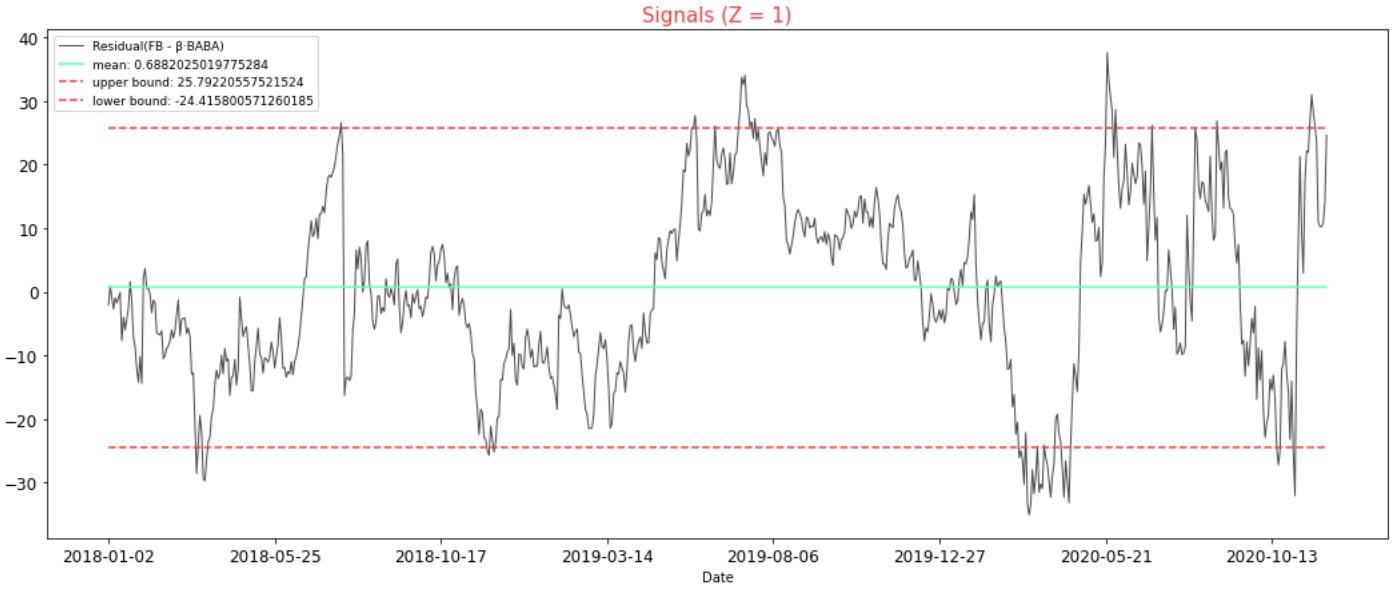
```
In [62]: B = model.coef.values[0][0]
         C = model.coef.values[1][0]
         mu = C/(1-B)
         tau = 1/252
         SSE = np.sum(model.error**2)
         sigma_eq = ((SSE*tau/(1-B**2))**0.5).values[0]
         mu, sigma_eq
```

```
Out[62]: (1.173163682101924, 27.363641646239387)
```

Computing the metrics
B = 0.95686
C = 0.05061
$\mu$ = C/(1-B) = 1.1732
$\tau$ = 1/252
$\Sigma$ = sum of square error * $\tau$
$\sigma_{eq}$ = 27.363642

After acquiring $\mu$ and $\sigma_{eq}$, now we can construct our bounds and generate signals.
Mean = 0.688
Upper Bound = 25.792
Lower Bound = -24.416

First choose Z = 1; here is our signal graph



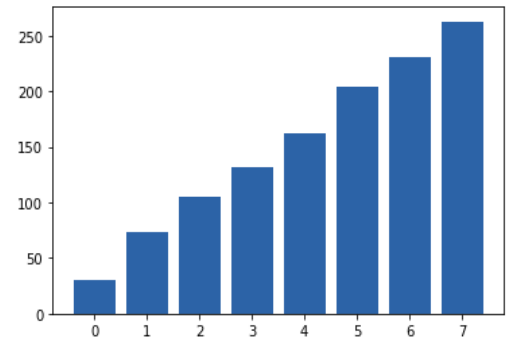

(*Figure

## Trading: Backtesting

*The hedging ratios are for entry 1 is [-1, 0.864299]
*The hedging ratios are for entry 2 is [1, -0.864299]

| Date | BABA | FB | res_past | res | action |
|---|---|---|---|---|---|
| 2018-03-20 | 198.949997 | 168.149994 | -20.317073 | -28.547278 | entry2 |
| 2018-06-21 | 202.210007 | 201.500000 | -0.989369 | 1.985104 | exit |
| 2018-07-24 | 189.000000 | 214.669998 | 24.506539 | 26.572501 | entry1 |
| 2018-07-26 | 194.179993 | 176.259995 | 21.641100 | -16.314566 | exit |
| 2018-11-27 | 156.460007 | 135.000000 | -23.204257 | -24.973207 | entry2 |
| 2019-04-25 | 187.880005 | 193.259995 | -2.639378 | 6.130508 | exit |
| 2019-05-30 | 151.070007 | 183.009995 | 25.656716 | 27.695360 | entry1 |
| 2019-12-11 | 204.639999 | 202.259995 | 2.876275 | 0.644858 | exit |
| 2020-03-09 | 197.660004 | 169.500000 | -20.525140 | -26.082332 | entry2 |
| 2020-04-30 | 202.669998 | 204.710007 | -9.205588 | 4.797541 | exit |
| 2020-05-22 | 199.699997 | 234.910004 | 23.275329 | 37.564507 | entry1 |
| 2020-07-08 | 257.679993 | 243.580002 | 11.699653 | -3.877559 | exit |
| 2020-08-07 | 252.100006 | 268.440002 | 10.908045 | 25.805220 | entry1 |
| 2020-09-16 | 278.140015 | 263.519989 | 7.434466 | -1.621152 | exit |
| 2020-10-16 | 307.309998 | 265.929993 | -16.847979 | -24.422742 | entry2 |
| 2020-11-04 | 295.709991 | 287.380005 | -6.262890 | 7.053146 | exit |
| 2020-11-13 | 260.839996 | 276.950012 | 21.892118 | 26.761263 | entry1 |

```
====================Cumulative Profit====================
0     30.532383
1     73.419449
2    104.523171
3    131.573661
4    162.453535
5    203.895589
6    231.321956
7    262.797841
dtype: float64
```



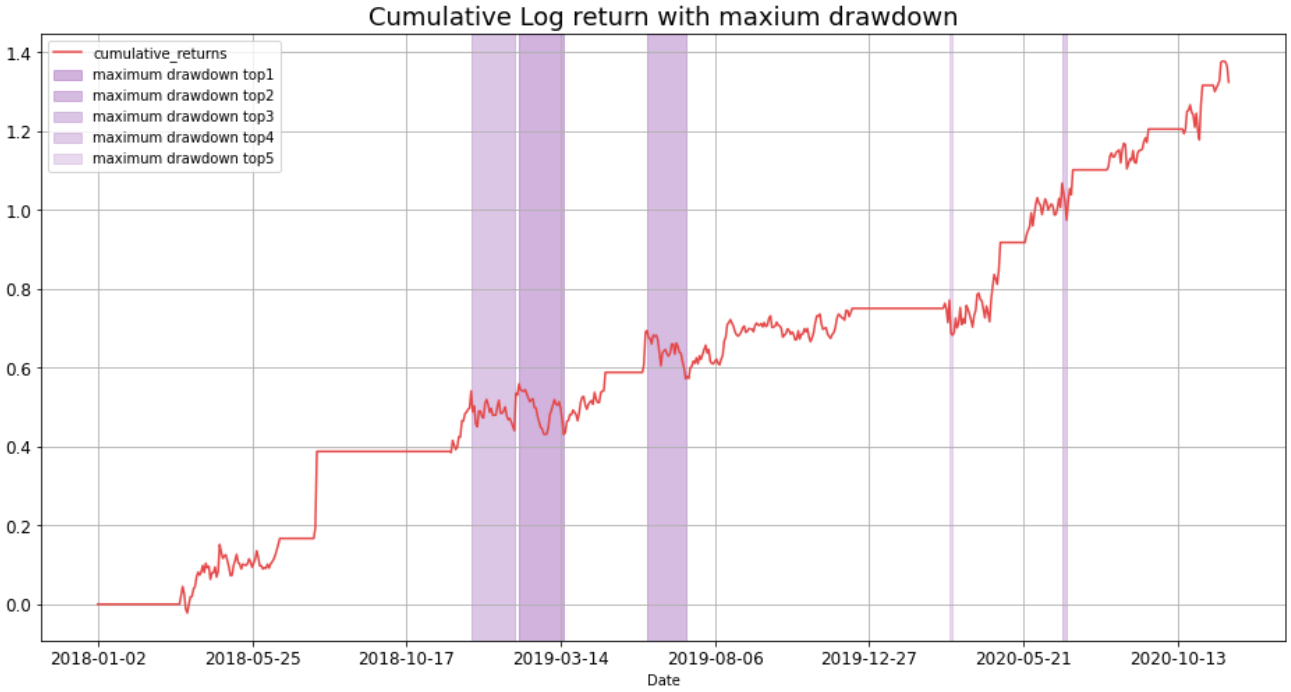

*Figu (*Figure cumulative profits measure in $. Total cumulati (*Figure 3.1.3)

The assumptions here are:

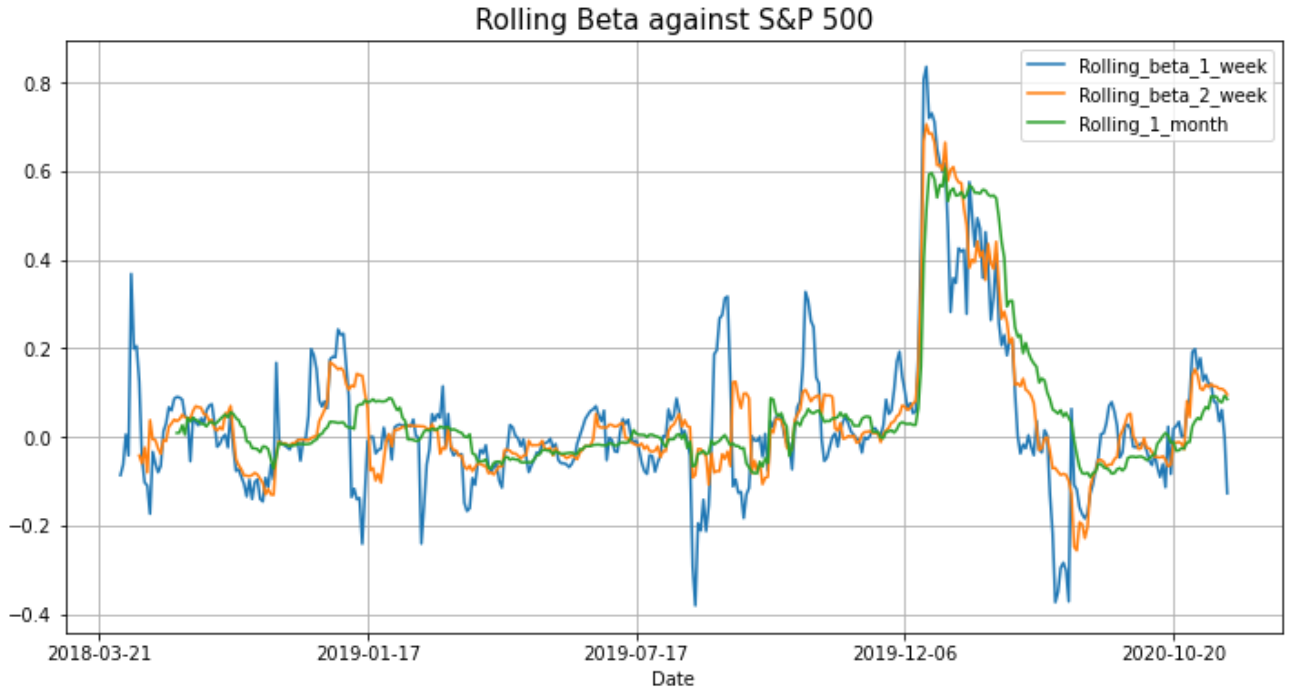
1. The market is frictionless

2. each trade only long 1 stock and short 0.864299 stock or short 1 stock and long 0.864299 stock,

*Figure 3.1.3 is the trading record.
*Figure 3.1.4 is the maximum drawdown graph.
*Figure 3.1.5 is rolling beta against S&P 500.

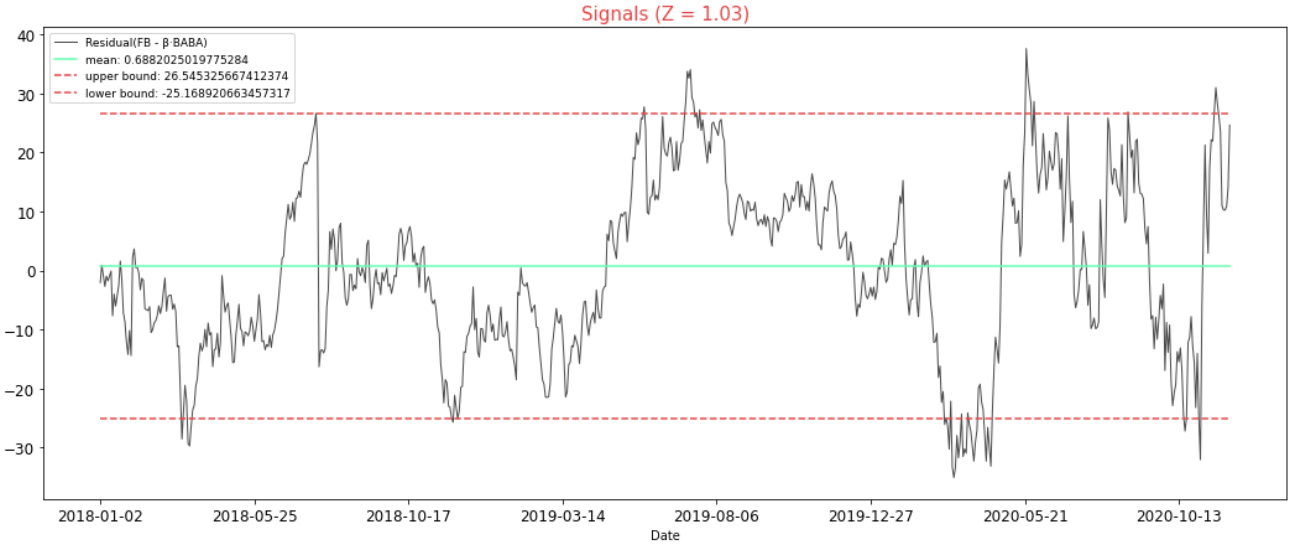



(*Figure

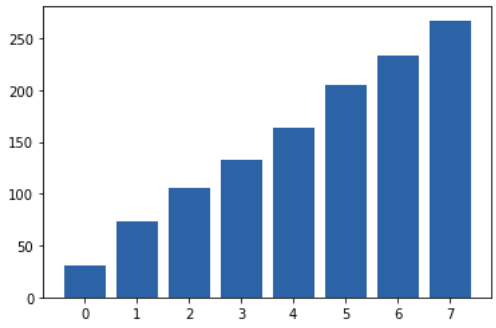



(*Figure

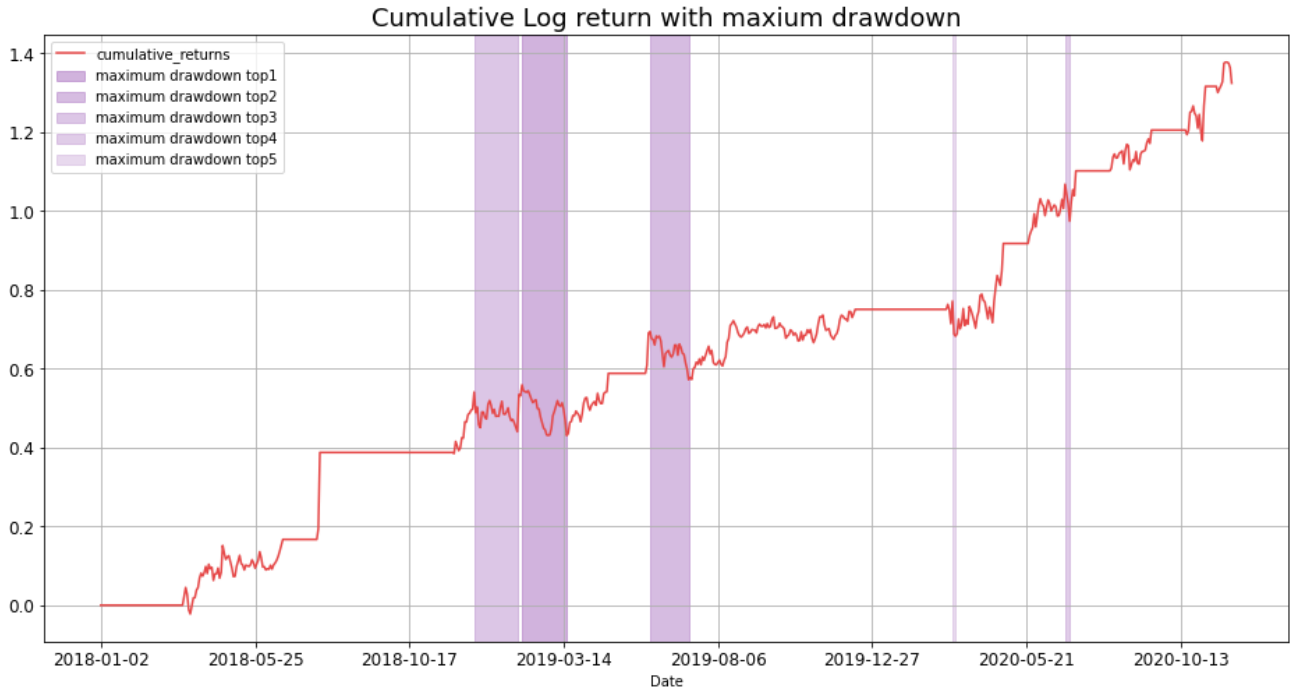
# When Z = 1.03

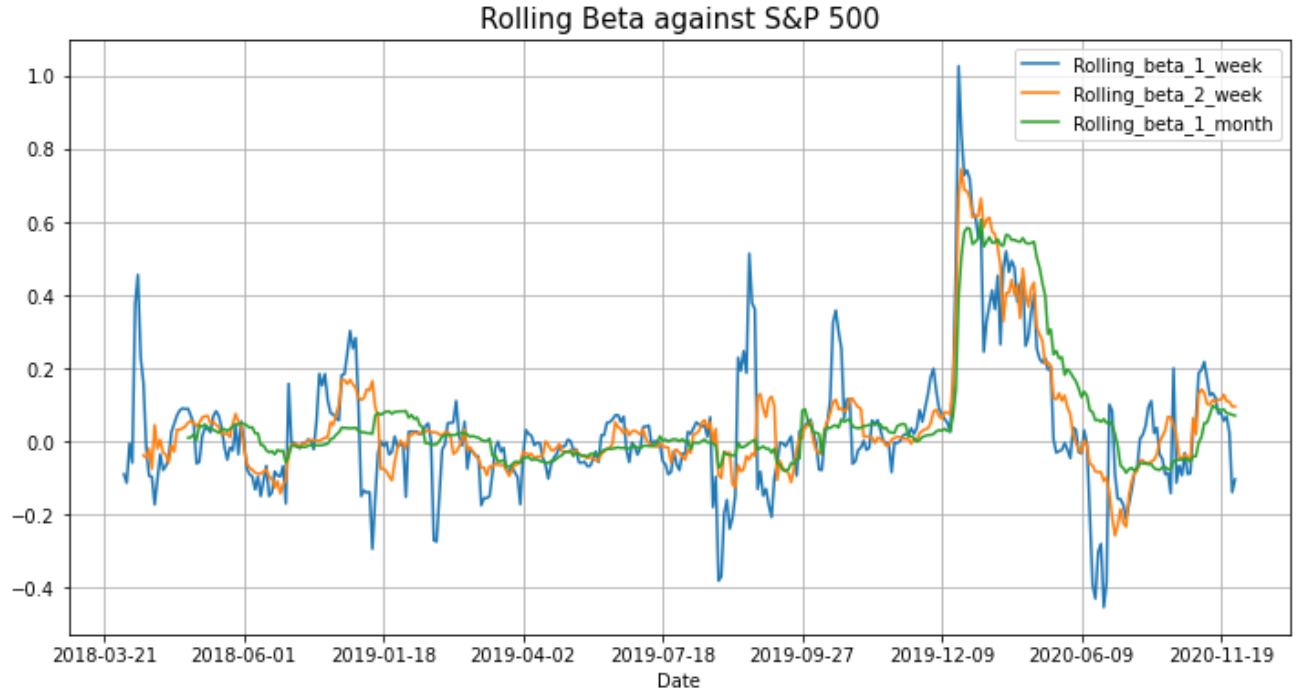



```
====================Cumulative Profit====================
0     30.532383
1     73.419449
2    105.252348
3    132.302839
4    163.182712
5    204.624766
6    233.070189
7    267.330199
dtype: float64
```

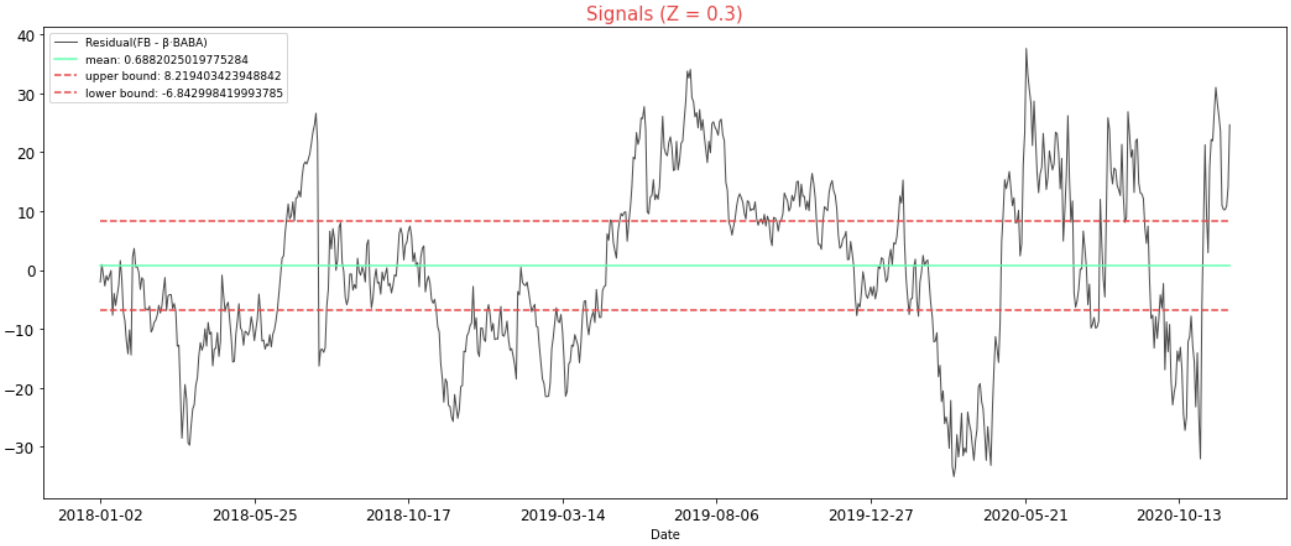



| Date | BABA | FB | res_past | res | action |
|---|---|---|---|---|---|
| 2018-03-20 | 198.949997 | 168.149994 | -20.317073 | -28.547278 | entry2 |
| 2018-06-21 | 202.210007 | 201.500000 | -0.989369 | 1.985104 | exit |
| 2018-07-24 | 189.000000 | 214.669998 | 24.506539 | 26.572501 | entry1 |
| 2018-07-26 | 194.179993 | 176.259995 | 21.641100 | -16.314566 | exit |
| 2018-11-28 | 159.339996 | 136.759995 | -24.973207 | -25.702385 | entry2 |
| 2019-04-25 | 187.880005 | 193.259995 | -2.639378 | 6.130508 | exit |
| 2019-05-30 | 151.070007 | 183.009995 | 25.656716 | 27.695360 | entry1 |
| 2019-12-11 | 204.639999 | 202.259995 | 2.876275 | 0.644858 | exit |
| 2020-03-09 | 197.660004 | 169.500000 | -20.525140 | -26.082332 | entry2 |
| 2020-04-30 | 202.669998 | 204.710007 | -9.205588 | 4.797541 | exit |
| 2020-05-22 | 199.699997 | 234.910004 | 23.275329 | 37.564507 | entry1 |
| 2020-07-08 | 257.679993 | 243.580002 | 11.699653 | -3.877559 | exit |
| 2020-08-26 | 291.959991 | 303.910004 | 8.885487 | 26.824267 | entry1 |
| 2020-09-16 | 278.140015 | 263.519989 | 7.434466 | -1.621152 | exit |
| 2020-10-19 | 305.290009 | 261.399994 | -24.422742 | -27.206866 | entry2 |
| 2020-11-04 | 295.709991 | 287.380005 | -6.262890 | 7.053146 | exit |
| 2020-11-13 | 260.839996 | 276.950012 | 21.892118 | 26.761263 | entry1 |

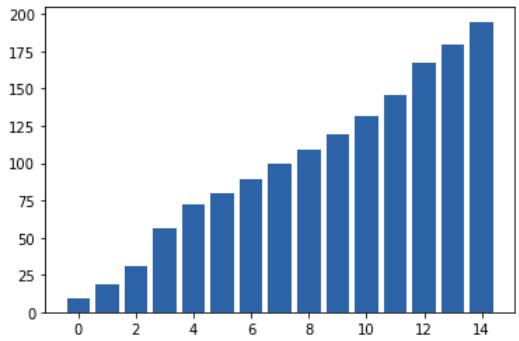


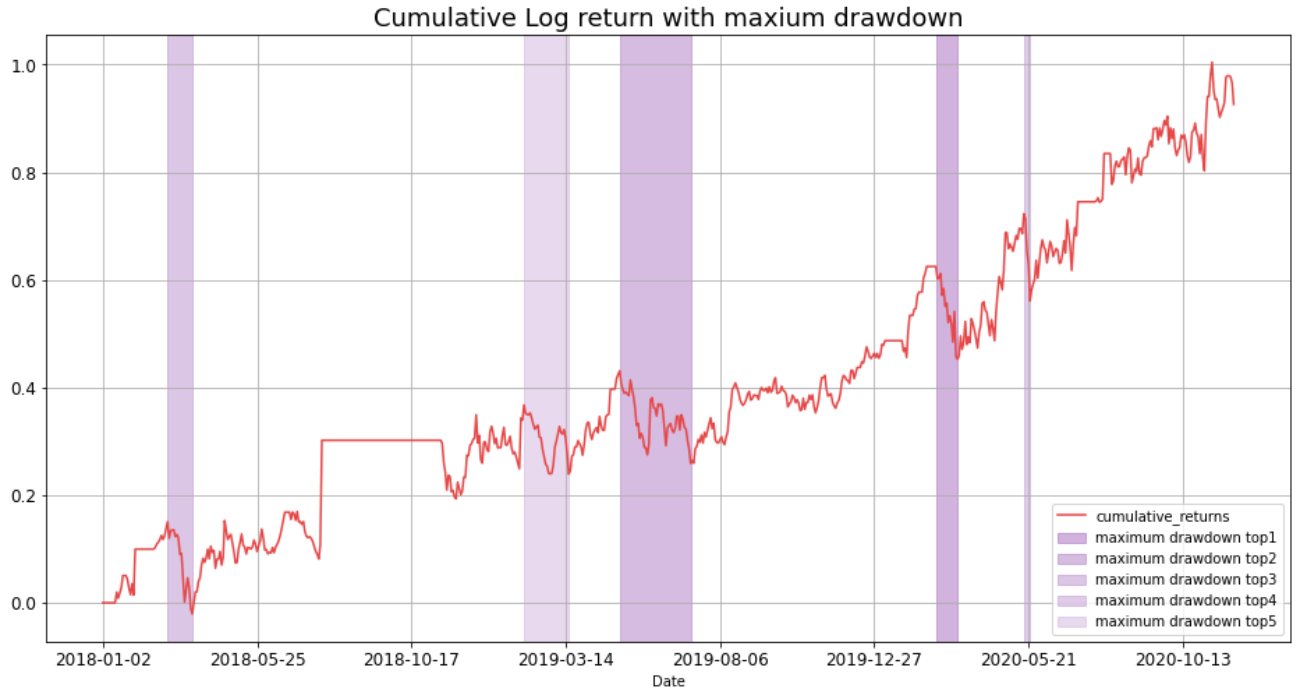

**And finally, Z = 0.3**

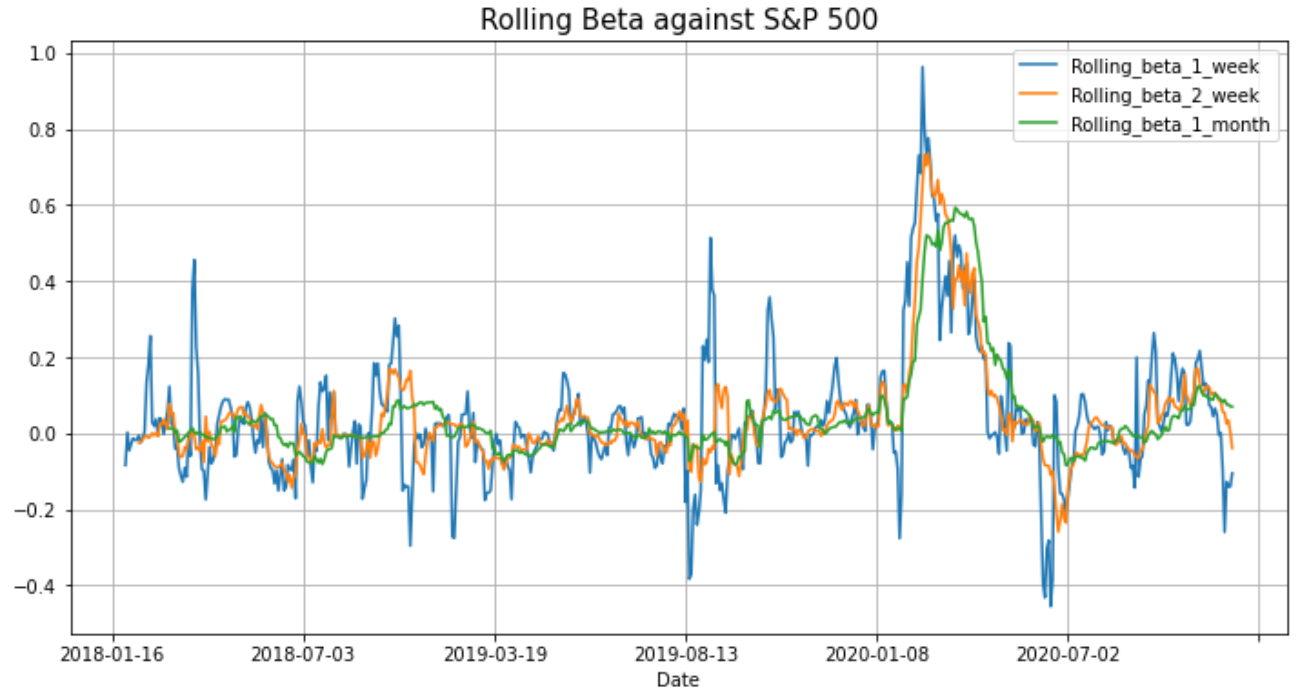



```
====================Cumulative Profit====================
0       9.258398
1      18.659219
2      31.167449
3      56.292915
4      72.043069
5      79.861163
6      89.657493
7      99.499108
8     108.874270
9     119.189966
10    131.946277
11    145.259974
12    167.073263
13    179.602458
14    194.885127
dtype: float64
```




| Date | BABA | FB | res_past | res | action |
|---|---|---|---|---|---|
| **2018-01-12** | 187.789993 | 179.369995 | -0.111418 | -7.681694 | entry2 |
| **2018-01-22** | 184.020004 | 185.369995 | -2.529226 | 1.576704 | exit |
| **2018-01-24** | 195.529999 | 186.550003 | -1.582392 | -7.191367 | entry2 |
| **2018-02-01** | 192.220001 | 193.089996 | -14.422627 | 2.209455 | exit |
| **2018-02-20** | 187.190002 | 176.009995 | -6.139419 | -10.523123 | entry2 |
| **2018-06-21** | 202.210007 | 201.500000 | -0.989369 | 1.985104 | exit |
| **2018-06-26** | 191.419998 | 199.000000 | 6.307836 | 8.810900 | entry1 |
| **2018-07-26** | 194.179993 | 176.259995 | 21.641100 | -16.314566 | exit |
| **2018-11-13** | 146.979996 | 142.160004 | -6.634163 | -9.619637 | entry2 |
| **2019-04-25** | 187.880005 | 193.259995 | -2.639378 | 6.130508 | exit |
| **2019-04-29** | 186.940002 | 194.779999 | 5.043323 | 8.462956 | entry1 |
| **2019-12-11** | 204.639999 | 202.259995 | 2.876275 | 0.644858 | exit |
| **2019-12-13** | 204.910004 | 194.110001 | -4.744135 | -7.738500 | entry2 |
| **2020-01-08** | 218.000000 | 215.220001 | 0.217610 | 2.057827 | exit |
| **2020-01-24** | 213.750000 | 217.940002 | 5.621158 | 8.451100 | entry1 |
| **2020-01-31** | 206.589996 | 201.910004 | 4.509521 | -1.390514 | exit |
| **2020-02-04** | 222.880005 | 209.830002 | -4.737111 | -7.549957 | entry2 |
| **2020-02-10** | 215.770004 | 213.059998 | 0.438348 | 1.825207 | exit |
| **2020-02-12** | 224.309998 | 210.759995 | -5.289381 | -7.855906 | entry2 |
| **2020-02-18** | 220.520004 | 217.800003 | -0.390994 | 2.459791 | exit |
| **2020-02-18** | 220.520004 | 217.800003 | -0.390994 | 2.459791 | exit |
| **2020-02-26** | 208.740005 | 197.199997 | -5.683503 | -7.958771 | entry2 |
| **2020-04-30** | 202.669998 | 204.710007 | -9.205588 | 4.797541 | exit |
| **2020-05-01** | 194.479996 | 202.270004 | 4.797541 | 9.436151 | entry1 |
| **2020-07-08** | 257.679993 | 243.580002 | 11.699653 | -3.877559 | exit |
| **2020-07-23** | 251.880005 | 232.600006 | -2.419060 | -9.844630 | entry2 |
| **2020-07-31** | 251.020004 | 253.669998 | -8.687934 | 11.968660 | exit |
| **2020-08-06** | 265.679993 | 265.279999 | -4.586458 | 10.908045 | entry1 |
| **2020-09-16** | 278.140015 | 263.519989 | 7.434466 | -1.621152 | exit |
| **2020-09-17** | 275.720001 | 254.820007 | -1.621152 | -8.229518 | entry2 |
| **2020-11-04** | 295.709991 | 287.380005 | -6.262890 | 7.053146 | exit |
| **2020-11-05** | 287.750000 | 294.679993 | 7.053146 | 21.232949 | entry1 |




**Summary of results:**

As the results show, the choice of Z will impact the performance of strategy. There is a trade off on Z as in if Z value is too large, we will generate more profit at

sacrifice of number of trades. This is understandable because larger Z means the bounds are further away from equilibrium and so it would be harder to residual to reach the bound in order to generate trade signals. Thus, if Z is too high, we will end up with a strategy that will never ever execute any trade order. On the other hand, if Z is very low, (e.g., compare the trading record for Z = 0.3 and Z = 1) then more trades will be executed but for less profits. That being said, a low Z value can be still risky because in the above model we did not consider cost and fees for each transaction. If the cost for each trade exceeds the profit, then no matter how frequently trades are executed, the strategy will always yield negative return.

For the pair of Facebook and Alibaba:
###########Profit#############
Z = 1.03 Profit = $ 267.33
Z = 1 Profit = $ 262.79
Z = 0.3 Profit = $ 194.88

Profit was maximized at Z = 1.03 with $267.33.

**Cointegration Over Time**
Lastly, I will shift the data 1 month forward and estimate the cointegration. The shifted data can be found in data2.csv. Here I will just attach the results.

| | Estimate Coefficient FB | SD of Estimate FB | t-Statistic FB |
|---|---|---|---|
| **BABA** | 0.894459 | 0.014181 | 63.076613 |
| **constant** | 19.816957 | 2.852997 | 6.946013 |

The βcoint here is 0.894459, close to the βcoint we previously have which is 0.864299.

```
The T-value is [-4.04219767] which is lower than -3.45
[*]We reject the  H0  hypothesis of unit root. The residuals are stationary.
```

| | Estimate Coefficient (ΔResidual(FB - β·BABA)) | SD of Estimate (ΔResidual(FB - β·BABA)) | t-Statistic (ΔResidual(FB - β·BABA)) |
|---|---|---|---|
| **(Lag 1, Residual(FB - β·BABA))** | -0.046338 | 0.011464 | -4.042198 |
| **(Lag 1, ΔResidual(FB - β·BABA))** | 0.059493 | 0.037226 | 1.598149 |
| **constant** | 0.042902 | 0.172465 | 0.248759 |

| | Estimate Coefficient ΔFB | SD of Estimate ΔFB | t-Statistic ΔFB |
|---|---|---|---|
| **ΔBABA** | 0.470399 | 0.033576 | 14.009931 |
| **(Lag 1, Residual(FB - β·BABA))** | -0.036619 | 0.010246 | -3.573904 |

P value < 0.05

| | ΔFB |
|---|---|
| ΔBABA | True |
| (Lag 1, Residual(FB - β·BABA)) | True |

While the β has changed slightly over time, the cointegration still exist stocks between two stocks.

# Discussion and Conclusion:

In this section, the following points will be discussed

- The weakness of model
- How to improve model: Kalman filter

There are several weaknesses of this model. First pair trading is a neutral position strategy that heavily rely on mean-reversion property of residuals. In practice the market is full of noise so the market might not correct within a period of time which is risky to traders. Also, in the back testing procedure, metrics like transaction cost and fees or order flows etc. are not put in consideration. From this perspective, the model is more of a theoretical proof that statistical arbitrage via a pair of cointegrated equities is possible. In the above strategy, we use Engle Granger two step to test cointegration. The pro side of Engle Granger is obvious: it is every easy to understand and implement. However, unlike the other methods it is not efficient and cannot used to determine the number of cointegration relationships. Also, during backtesting, we used a static hedging ratio [1, β = 0.864299] but in practice we cannot trade 0.864299 stock, that does not make any sense in real world. This is something we can improve too. Hence, in general, this model is still too simple compared to industry level.

Kalman filter has a number of applications in various tech industries, especially in technology that involves signal generating such as guidance, navigation, auto vehicle's control. In order to use Kalman Filter, we need to write out our state space model. A state space model describes the relation between observable variables and unobservable variables has the general form:

$$\beta_t = \mathbf{A} * \beta_{t-1} + w$$
$$y_t = \beta_t * x_t + v$$

Where w ~ N(0, Q), v ~ N(0, R), beta βt unobservable variable and y and x are observable. The first equation is called state equation which describes how unobservable variable is vary based on its past. e.g., if A = 1, then the state equation is a random walk model. Xt and Yt are the observable incoming information. In our project, x and y will be the price of Alibaba and Facebook. The Kalman filter involves

two steps: 1. prediction where the Kalman filter produces estimates of current state variables and 2. update once the outcome of next measurement is observed.

Prediction

$$\beta_{t\_est} = \mathbf{A} * \boldsymbol{\beta}_{\mathbf{t-1}}$$

$$P_{t\_est} = \mathbf{A}_{\mathbf{t-1}} * P_{t-1} * \mathbf{A'}_{\mathbf{t-1}} + Q$$

Update

$$V_t = y_t - \boldsymbol{\beta}_{t\_est} * x_t$$

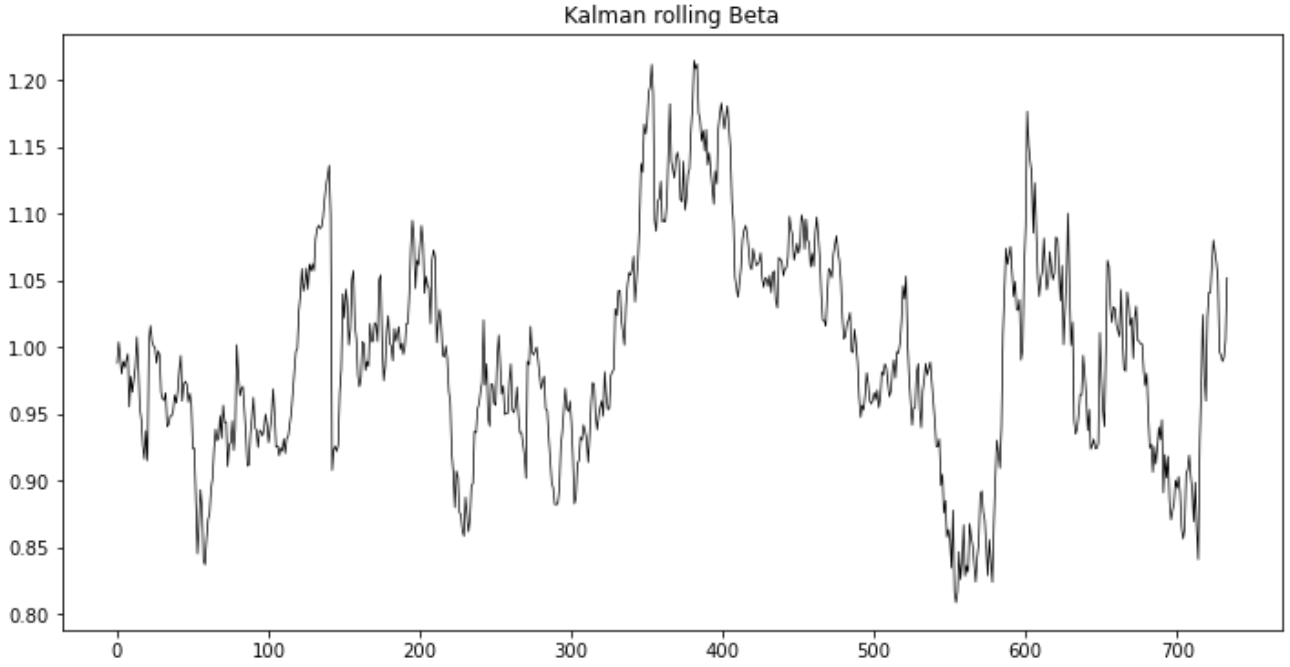
$$S_t = \mathbf{X}_{\mathbf{t}} * P_{t\_est} * \mathbf{X'}_{\mathbf{t}} + R_t$$

$$K_t = P_{t\_est} * \mathbf{X'}_{\mathbf{t}} * S_t^{-1}$$

$$\beta_t = \boldsymbol{\beta}_{t\_est} + V_t * K_t$$

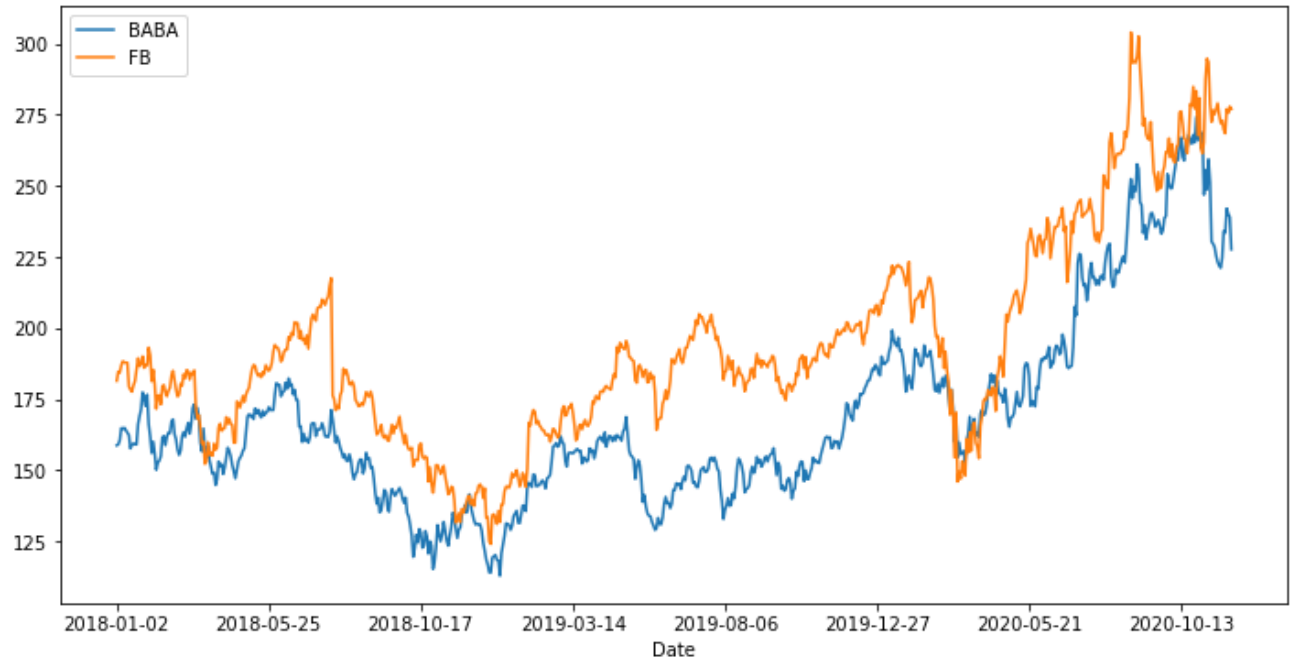
$$P_t = P_{t\_est} - \mathbf{k}_{\mathbf{t}} * S_t * K'_t$$

The implementation can be found in Appendix and code section.
The updated beta:

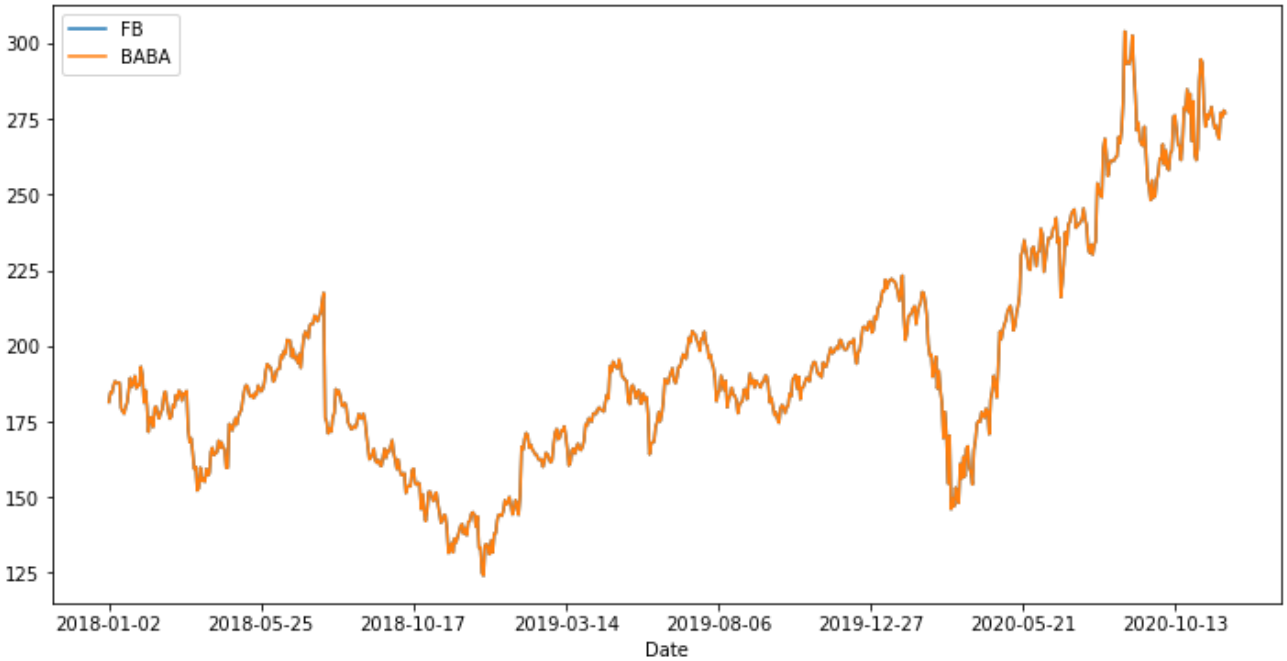



To see how power Kalman filter is, compare the following:

FB and BABA * β_static



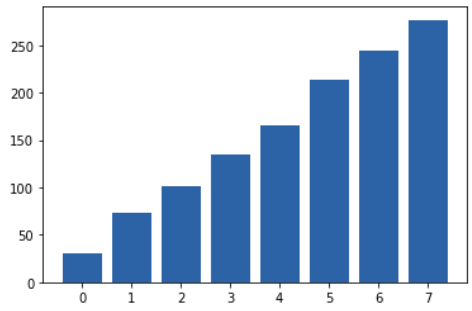

FB and BABA * β_Kalman



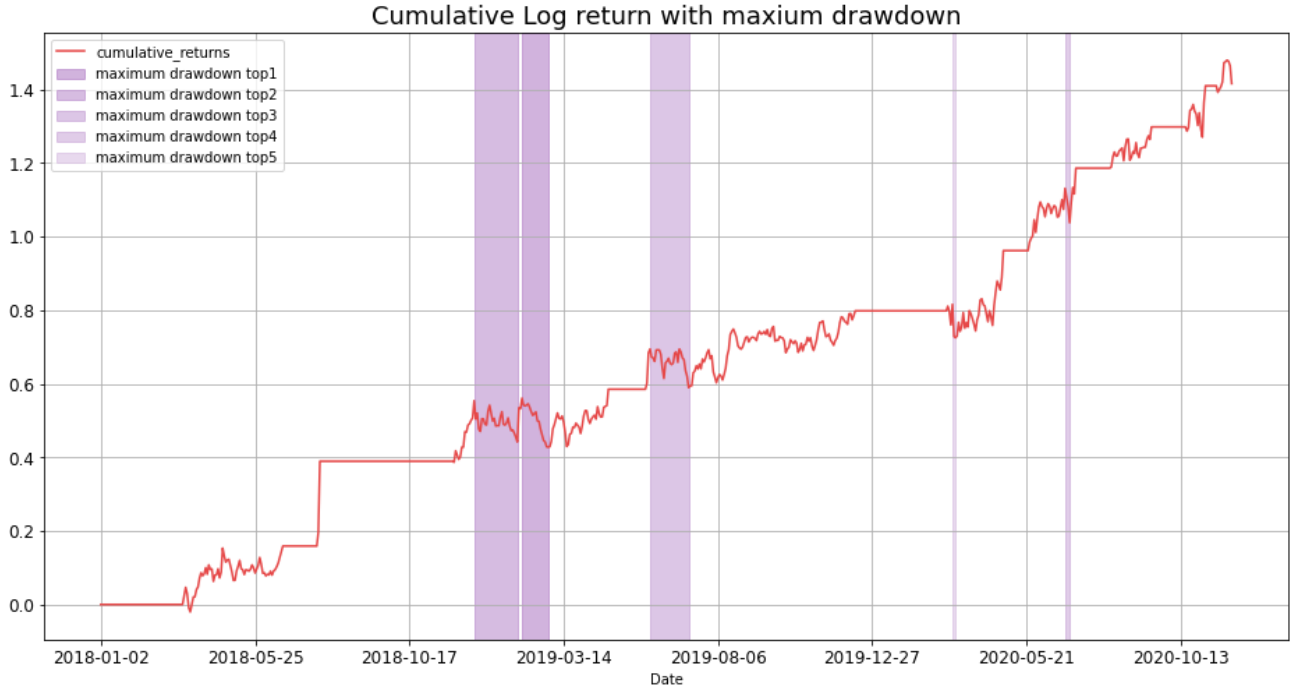

Since Kalman filter update β at each time point, the regression fits nearly perfectly with Kalman filter beta. And the trading results are listed here:

Note that the cumulative profit is now 276.76, even better than 267.33 the optional profit we had previously. Again, it only improves $9.43 because we only traded one stock at each time.

```
====================Cumulative Profit====================
0     30.078133
1     73.563542
2    100.886098
3    134.927843
4    165.238010
5    213.101530
6    244.252524
7    276.769777
dtype: float64
```




| Date | BABA | FB | res_past | res | action |
|---|---|---|---|---|---|
| 2018-03-20 | 198.949997 | 168.149994 | -20.317073 | -28.547278 | entry2 |
| 2018-06-21 | 202.210007 | 201.500000 | -0.989369 | 1.985104 | exit |
| 2018-07-24 | 189.000000 | 214.669998 | 24.506539 | 26.572501 | entry1 |
| 2018-07-26 | 194.179993 | 176.259995 | 21.641100 | -16.314566 | exit |
| 2018-11-27 | 156.460007 | 135.000000 | -23.204257 | -24.973207 | entry2 |
| 2019-04-25 | 187.880005 | 193.259995 | -2.639378 | 6.130508 | exit |
| 2019-05-30 | 151.070007 | 183.009995 | 25.656716 | 27.695360 | entry1 |
| 2019-12-11 | 204.639999 | 202.259995 | 2.876275 | 0.644858 | exit |
| 2020-03-09 | 197.660004 | 169.500000 | -20.525140 | -26.082332 | entry2 |
| 2020-04-30 | 202.669998 | 204.710007 | -9.205588 | 4.797541 | exit |
| 2020-05-22 | 199.699997 | 234.910004 | 23.275329 | 37.564507 | entry1 |
| 2020-07-08 | 257.679993 | 243.580002 | 11.699653 | -3.877559 | exit |
| 2020-08-07 | 252.100006 | 268.440002 | 10.908045 | 25.805220 | entry1 |
| 2020-09-16 | 278.140015 | 263.519989 | 7.434466 | -1.621152 | exit |
| 2020-10-16 | 307.309998 | 265.929993 | -16.847979 | -24.422742 | entry2 |
| 2020-11-04 | 295.709991 | 287.380005 | -6.262890 | 7.053146 | exit |
| 2020-11-13 | 260.839996 | 276.950012 | 21.892118 | 26.761263 | entry1 |



Although, pair trading is becoming less popular nowadays, it was still considered as one of the most brilliant strategies. It allows a trader to stay comfortably in the middle of a trade, no matter which direction the market goes, it always generates profit. The reality is that while skilled traders may be able to take advantage of the arbitrage, the market often doesn't correct itself as fast as a trader might expect. For this reason, pairs trading is recommended to more experienced traders.

## References:


Cointegration learning – Wilmott Nov 2013,
By Richard Diamond

Analysis of Financial Time Series,
By RUEY S. TSAY

Applied Economic Forecasting Using Time Series Method
By Eric Ghysels and Massimiliano Marcellino

Python for Data Analysis
By Wes McKinney


# Codes and Appendix