Algorithmic Trading Strategy

Group Members Name

- David Akanji
- Adewale Aderoju
- Yvan Dinock Dinock
- Daniel Nkameni

Introduction

Algorithmic trading is the way toward utilizing a computer program algorithmic that adheres to a characterized set of guidelines for placing a trade order. The point of the algorithmic trading program is to progressively recognize profitable chances and place the trades in order to make profits at a speed and recurrence that is difficult to coordinate by a human broker. Given the upsides of higher exactness and exceptionally quick execution speed, trading exercises based on computer algorithms have increased enormously and prominence.

This project is about the first style of Pair Trading strategy – Distance Based Pair Trading. Be that as it may, before that, we should initially comprehend what is pair trading.

Description of Distance-based Pair Trading

Pair trading is only a basic trading strategy in which we initially select 2 connected stocks, generally we pick stocks from a similar industry and afterward take a long situation in one stock and a short situation in another. We do this at whatever point we feel that the gap between their prices has suddenly increased. And we do this because we believe that the gap between their prices is mean reverting.

Pair trading using R

The aim of this part is to apply the steps that were described above on a concrete example. In this application we will apply pairs trading to:

- Renault (RNL) and
- PSA Peugeot Citroen (UG) stock prices

This two companies are known to have highly correlated stock prices and that will be shown with calculations. We collected from yahoo finance the close prices for this two companies on a daily basis for a period going from the 25th July 2018 to the 25th July 2020.

We will the proceed with the analysis and the establishment of a trading algorithm. The idea is to base ourselves of the behavior of the two stocks to decide when or not to trade depending on the deviation from their equilibrium trend. It is important to precise that this method applies because of the assumption of a mean-reverting process.

Importation of our data

```
> DATA3 <- read_excel("C:/Users/NKQ/Desktop/WQU FE/ECONOMETRICS/GROUP WORK/
S3/DATA3.xlsx",col_types = c("date", "numeric", "numeric"))
> View(DATA3)
```

The above code permitted the importation of the data into a Data Frame called DATA3 and later its visualization. Now that the data is imported, we can begin our analysis.

Correlation between stock prices

```
> cor(DATA3$RNL,DATA3$UG)
[1] 0.8752831
```

The correlation coefficient between the two stock prices is 0.88 which is very close to 1. This means that the stock prices of Renault and Peugeot are highly correlated and this is the primary condition to apply distance-based pairs trading.

As a statistician, it is important to precise that this correlation is highly significant at 5%. R supplies the following correlation test:

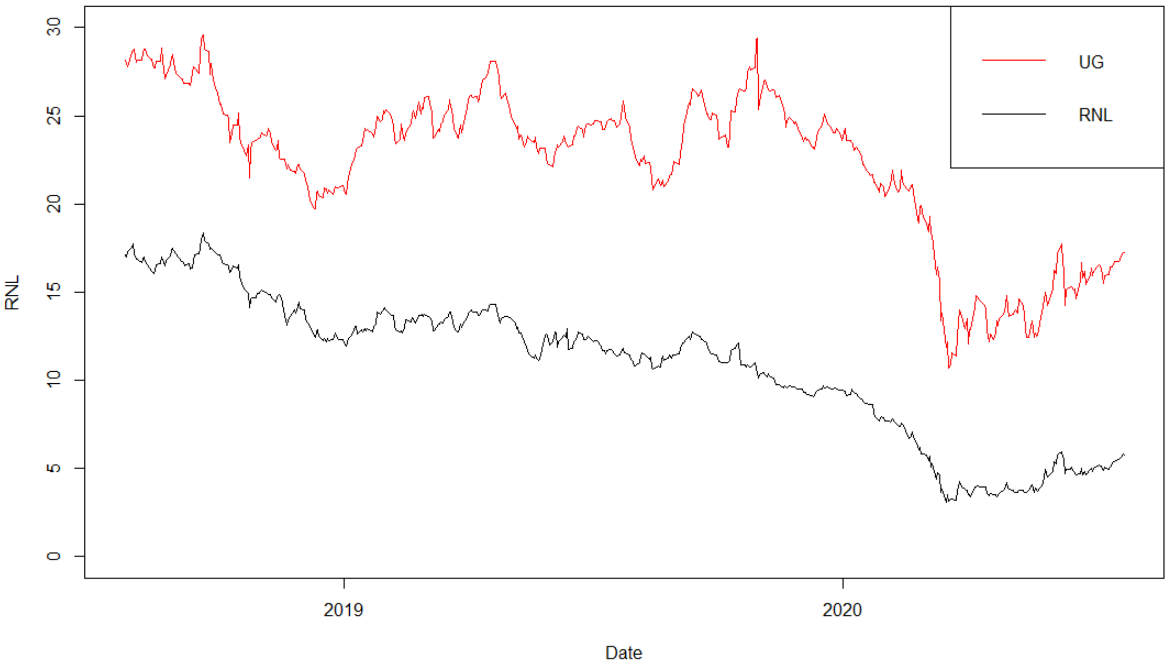
```
> cor.test(DATA3$RNL,DATA3$UG)

        Pearson's product-moment correlation

data:  DATA3$RNL and DATA3$UG
t = 40.551, df = 502, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8531647 0.8942597
sample estimates:
      cor
0.8752831
```

Part A: Pair trading using stock prices

In this part, we will elaborate a trading strategy basing ourselves on stock prices.

Graphical visualization of our data

```
> RNL=DATA3[,c(1,2)]
> UG=DATA3[,c(1,3)]
> plot(RNL,type="l",ylim=c(0,30))
> lines(UG,col="red")
> legend("topright",c("UG","RNL"),col=c("red","black"),lty=1)
```



From our plot, we realize that the two stock prices have a similar evolution, they share the same stochastic trend. This confirms what we observed with the coefficient of correlation between the two stock prices. The chart doesn't provide a clear view about the spread, so the spread has to be calculated.

Calculation of the spread

To calculate the spread, we will use a linear regression and the spread will be the residual of the linear regression. We have the formula:

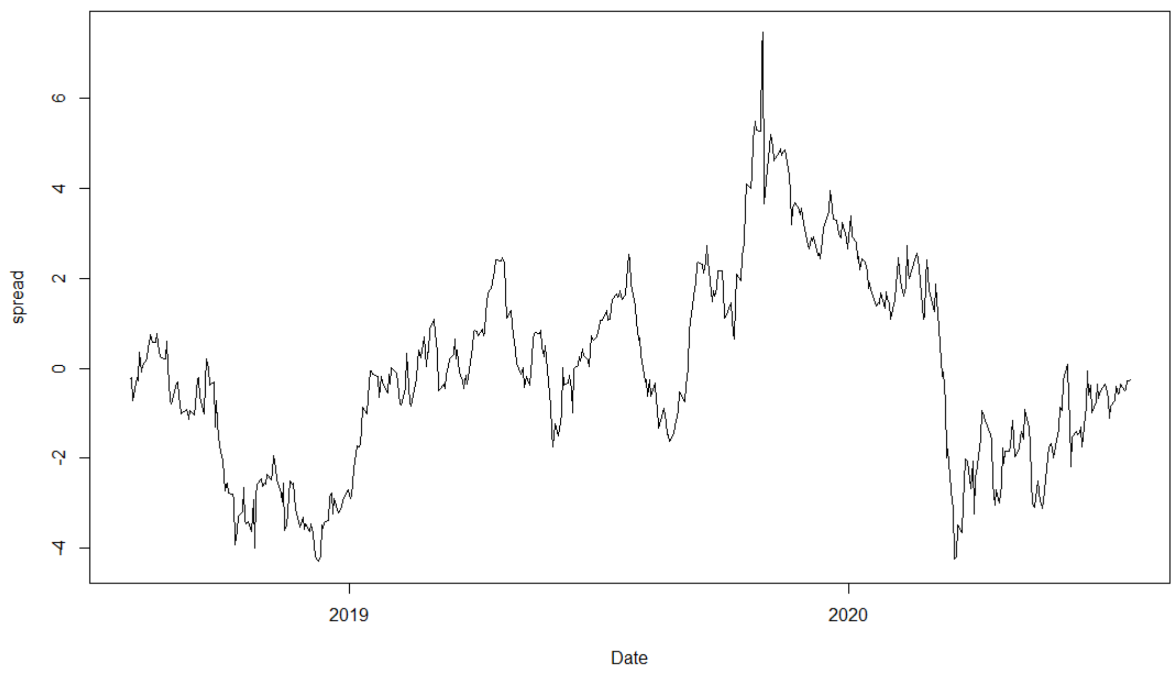
UG = slope*RNL + intercept

Spread = UG – slope*RNL – intercept

```
> UG=zoo(DATA3$UG,DATA3$Date)
> RNL=zoo(DATA3$RNL,DATA3$Date)
> model=lm(UG~RNL)
> spread=resid(model)
```

Plotting the spread gives:

```
> plot(spread,xlab="Date")
```



Calculation of the z-score

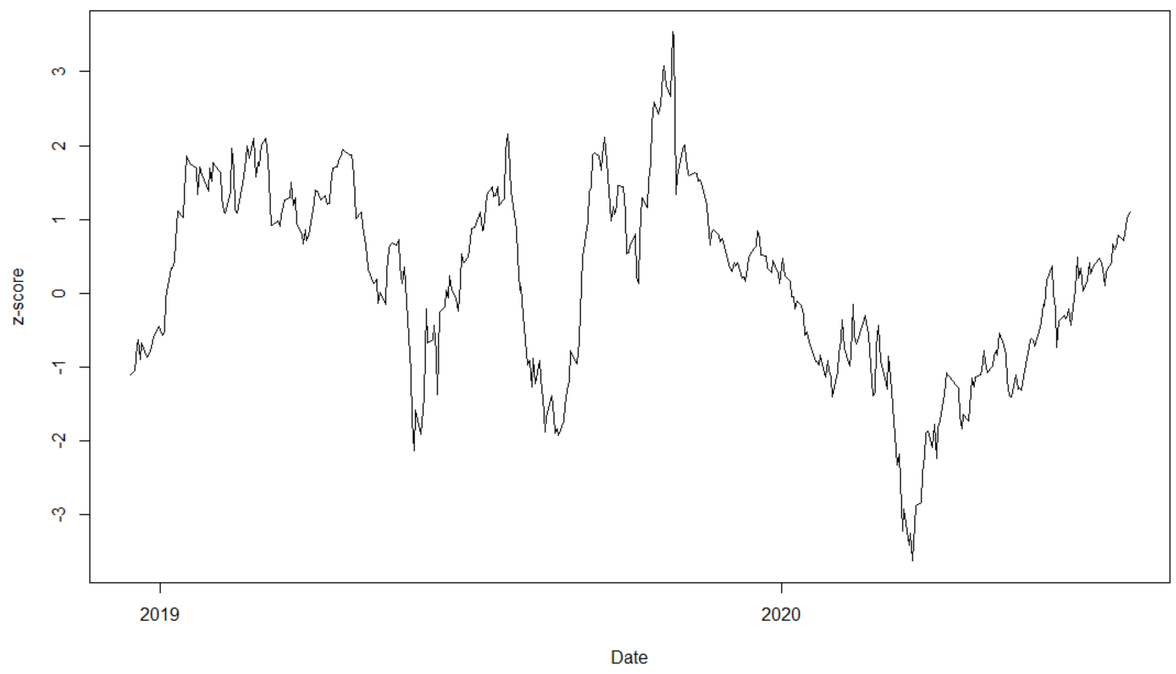
The z-score which gives us the trading decision is calculated as follows:

z-score = (spread – 100-day rolling average of spread)/(100-day rolling standard deviation of spread)

This formula is implemented in R using the following code:

```
> dymean=rollapply(spread,100,mean)
> dymean=zoo(dymean,DATA3$Date[-c(1:99)])
> dystd=rollapply(spread,100,sd)
> dystd=zoo(dystd,DATA3$Date[-c(1:99)])
> z_score=(spread[-c(1:99)]-dymean)/dystd
```

A plot of the z-score gives:



Trading strategy

The trading strategy is based on the z-score. The following rules are used

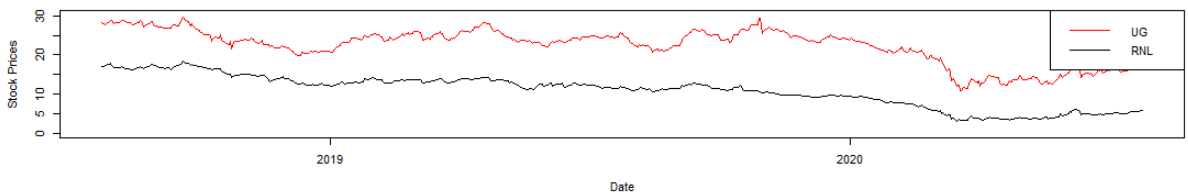
If z-score > 2 buy (long) RNL and sell (short) UG.

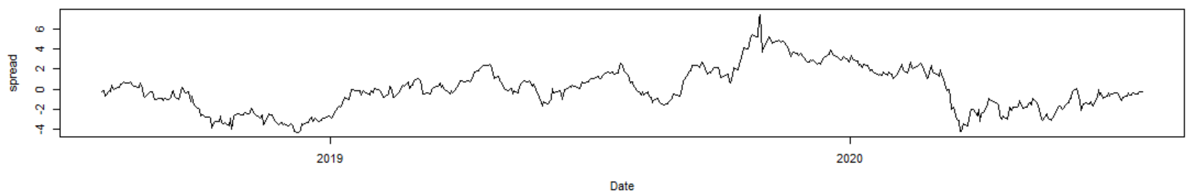
If z-score < -2 buy UG and sell RNL.

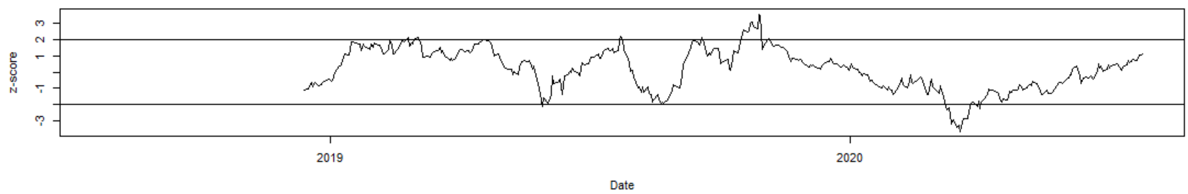
If z-score < 0.5 sell the two stocks (limit our exposure).

The detailed results of this algorithm are found in the attached files DATA3_Treated.xlsx.

The summary of our algorithm is given in the graph below:







We notice that there are trading opportunities around September 2019 and March 2020. Those trading opportunities are opposite. In September 2019, we might have to sell UG and buy RNL while in March 2020 we might have to buy UG and sell RNL. The word "might" here is because the applicability of this method is not always guaranteed.

Part B: Pairs Trading using returns (Backtesting)

In this part we will base our analysis on the returns of the two companies. No need to mention that the data to be used is the same as above.

Calculation of returns

The returns for both companies are calculated using the code:

```
> rRNL=Delt(RNL)
> rUG=Delt(UG)
```

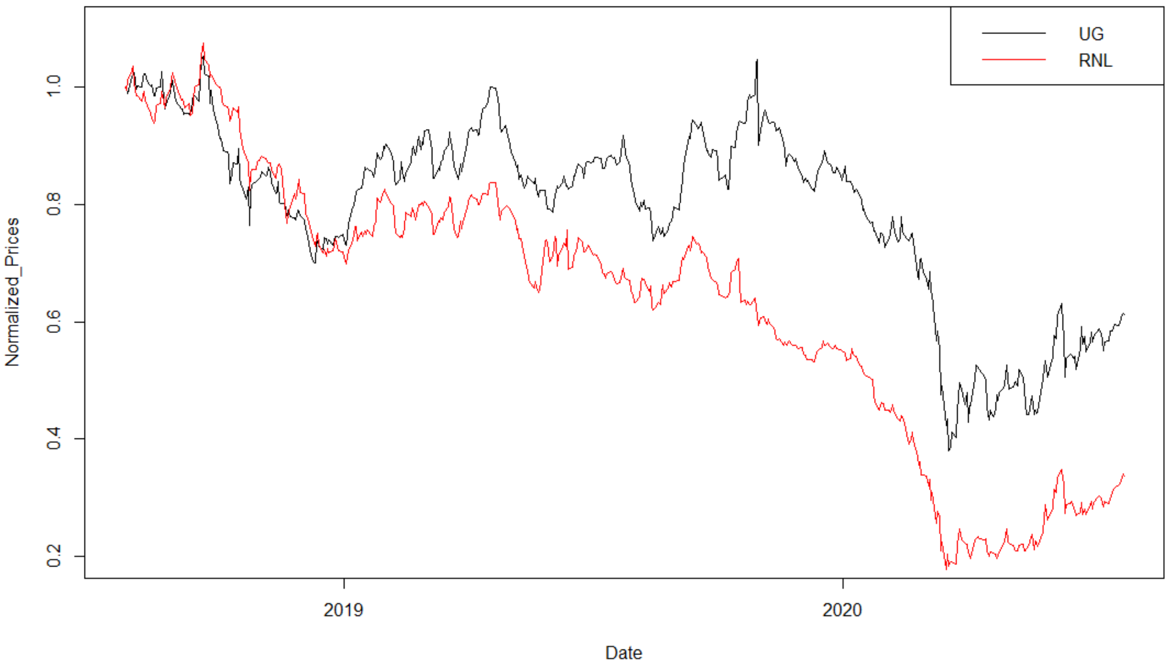
For distance-based trading, we need to normalize the returns and study the distance between them. We will calculate the value of 1 dollar invested in both stocks in the first day of our period of observation. Consider the code below:

```
> rRNL=round(rRNL+1,4)
> rRNL[1]=1
```

```
> norm_RNL=cumprod(rRNL)
> rUG=round(rUG+1,4)
> rUG[1]=1
> norm_UG=cumprod(rUG)
```

We then plot the 2 normalized prices on the same graph for a better visualization of the gap:

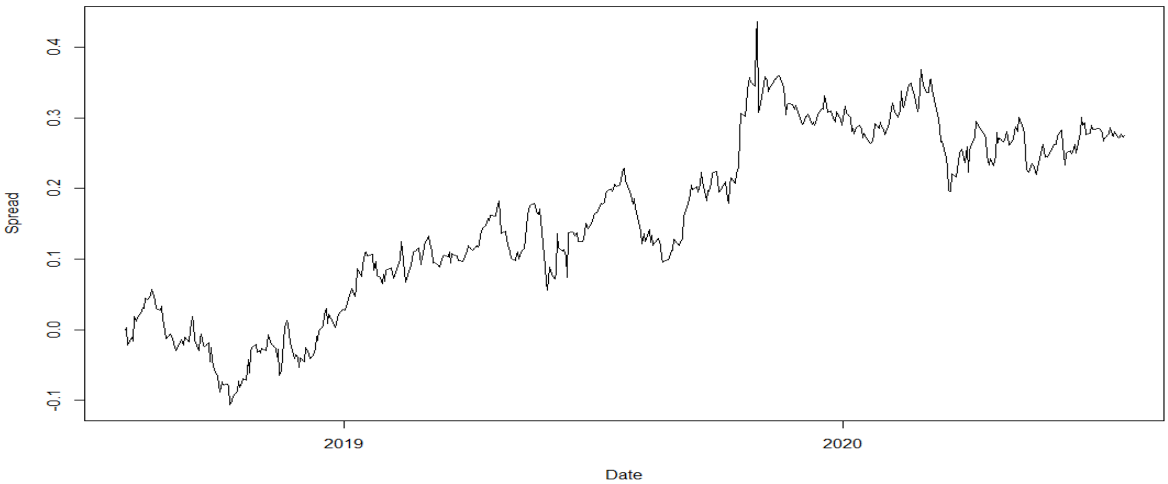
```
> plot(norm_UG,type="l",ylab="Normalized_Prices",xlab="Date",ylim=c(0.2,1.1
))
> lines(norm_RNL,col="red")
> legend("topright",c("UG","RNL"),col=c("black","red"),lty=1)
```



Calculation of the spread

The spread is calculated by subtracting the normalized returns

```
> spread2=norm_UG-norm_RNL
> plot(spread2,type="l",xlab="Date",ylab="Spread")
```



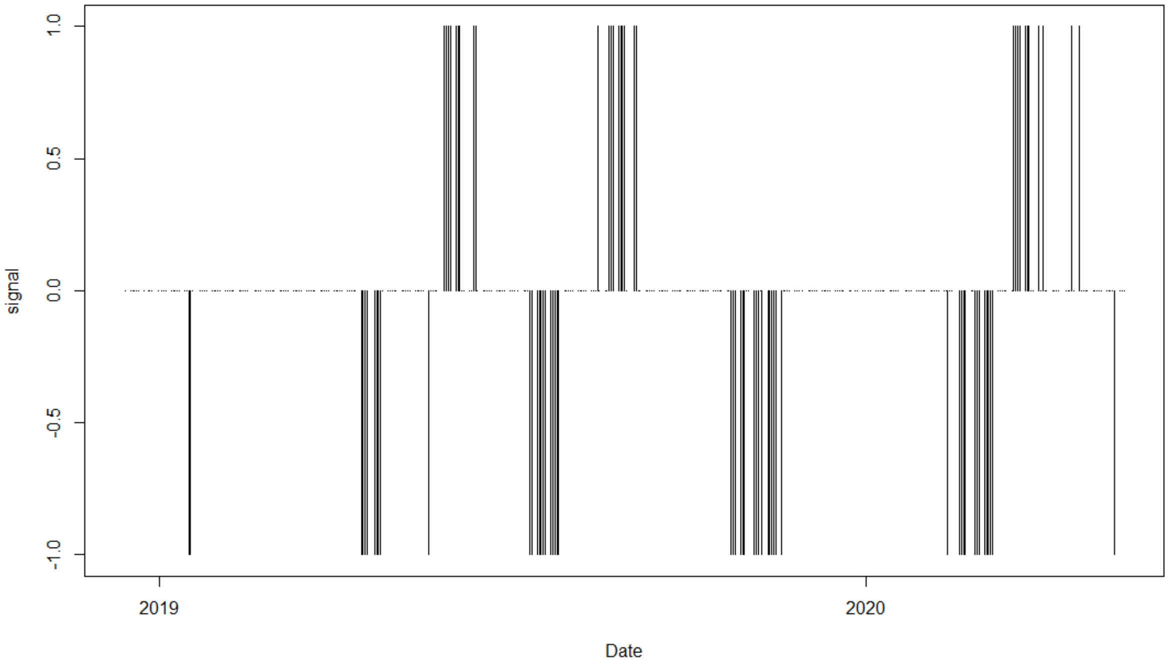
Calculation of the trading signal

The logic here will be to calculate the rolling mean and the rolling standard deviation on 100 periods. Whenever the spread goes above its 100-period rolling mean by one standard deviation, we'll short the spread expecting the mean reversion behavior to hold true. And when even the spread goes below its 100-period rolling mean by one standard deviation, we'll go long on the spread. The signal is generated using the code:

```
> dymean=rollapply(spread2,100,mean)
> dystd=rollapply(spread2,100,sd)
> upb=dymean+dystd
> lob=dymean-dystd
> signal=ifelse(spread2[-c(1:99)]>upb,-1,ifelse(spread2[-c(1:99)]<lob,1,0))
```

A plot of the signal gives:

```
> plot(signal,type="h",xlab="Date")
```



Analyzing the spread returns

Having the spread between returns, our return will depend on the return for the period next to that of the signal. Hence, we will use that lag function to calculate the return of this strategy. Consider the following code:

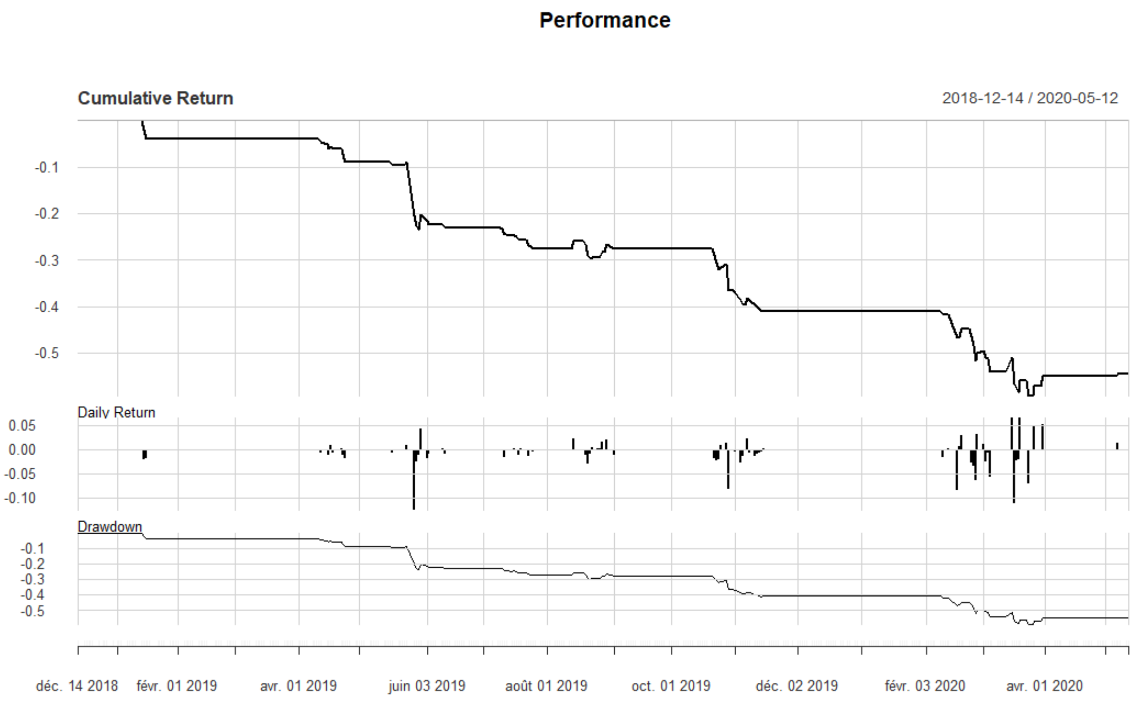
```
> spread_return=rUG-rRNL
> trade_return=spread_return*lag(signal)
```

Let's now analyze the returns beginning by generating a summary for the returns:

```
> summary(trade_return)
     Index                          trade_return
 Min.   :2018-12-14 00:00:00   Min.   :-0.125700
 1st Qu.:2019-04-24 06:00:00   1st Qu.: 0.000000
 Median :2019-08-28 12:00:00   Median : 0.000000
 Mean   :2019-08-29 13:01:01   Mean   :-0.002098
 3rd Qu.:2020-01-05 06:00:00   3rd Qu.: 0.000000
 Max.   :2020-05-12 00:00:00   Max.   : 0.064300
```

Then a graphical visualization of the performance

```
> charts.PerformanceSummary(trade_return)
```



Cumulative return

```
> Return.cumulative(trade_return)
                           x
Cumulative Return -0.5450489
```

Annualized return

```
> Return.annualized(trade_return)
                        [,1]
Annualized Return -0.4291563
```

Sharp ratio

```
> SharpeRatio(trade_return,Rf=0,p=0.95,FUN="StdDev")
                                 [,1]
StdDev Sharpe (Rf=0%, p=95%): -0.1352923
```

Annualized Sharp Ratio

```
> SharpeRatio.annualized(trade_return,Rf=0)
                                  [,1]
Annualized Sharpe Ratio (Rf=0%) -1.743559
```

Trading using Excel

In this part, we implement a trading strategy (pair trading) using Excel spreadsheets. Our data concern Renault (RNL) and PSA Peugeot Citroen (UG) stocks'. We begin by basic statistics followed by econometrics.

1. Pair trading in Excel using Basic statistics

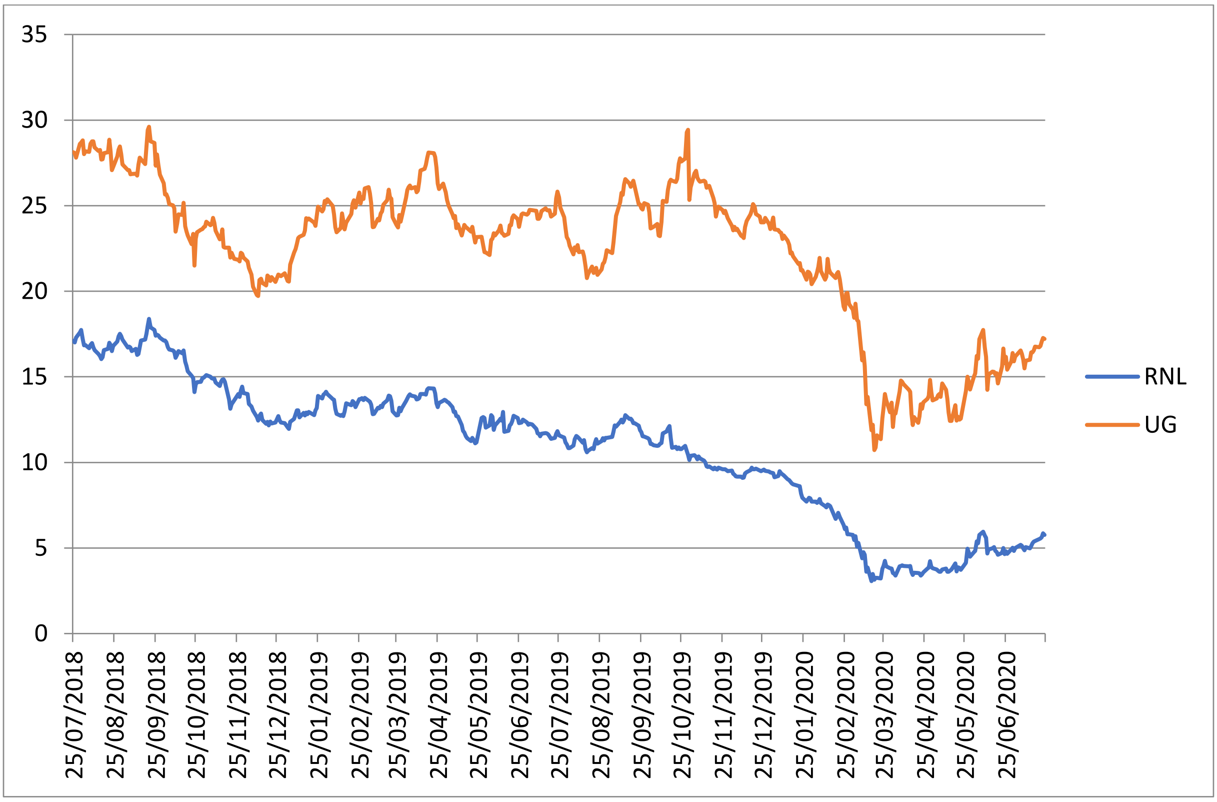
Calculations of correlation coefficient of the two series give us 0.8752. That's really close to 1 and shows that the stocks are high correlated (positively) and then we can apply pair trading.

For our work, we do pair trading fo July (2020). As in class' notes, the algorithm is the following:

we trade if abs(difference at t-1 period – average value of difference) >2*std(difference) else we don't trade. Results are available in attached file. We can simply say that there aren't opportunities for trade.

2. Pair trading using econometrics

The following graph shows evolutions of the two stocks over the period:



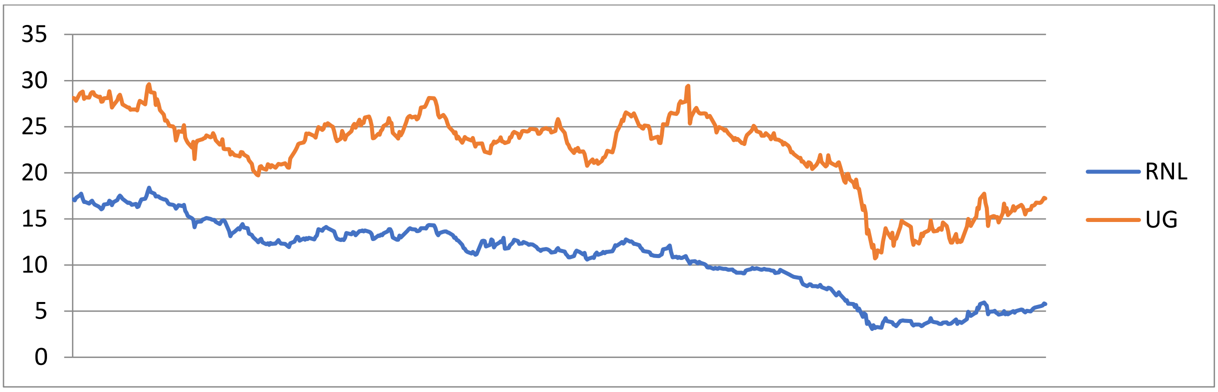
It is clear that the stocks have the same trend: they decrease over the period. The second observation is that the spread between them seems to be constant despite some weak enhancements (over sub periods 12/25/2018-05/25/2019 and 10/25/2019-12/25/2019).

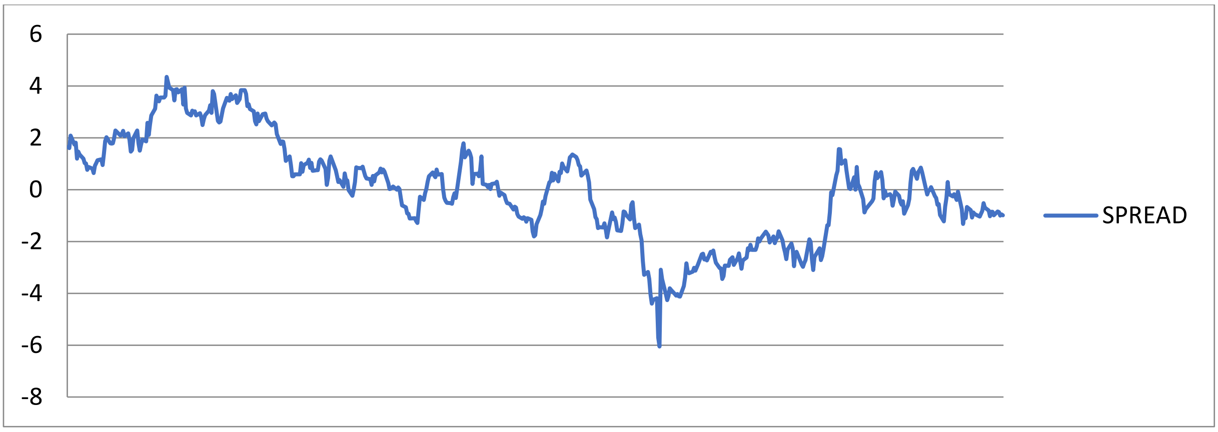
We made a linear regression between RNL and UG. The result is $\widehat{RNL} = -6.941 + 0.797 * UG$. All results are available in the attached file. Then, we calculate the residuals' serie for spreads serie' estimation.

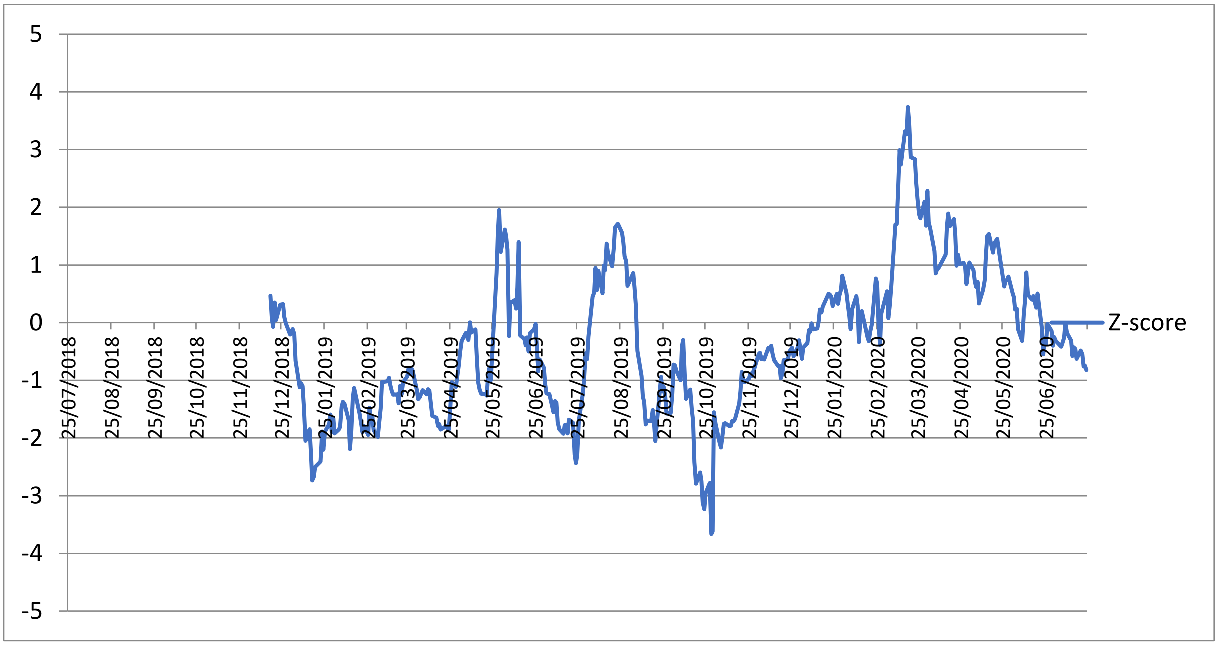
We opt for the following principle: if in the last 100 days the z-score is >2 then we sell RNL and buy UG because there are opportunities (the RNL share is expected to fall and the UG share is expected to rise). On the other hand if z-score <-2 then we buy RNL and sell UG (we have here the opposite of the previous case). In the case where z-score<0.5 then we sell both shares, knowing that there is no profit opportunity.

Data and calculations results lead us to sell both RNL and UG (see attached excel file).

The algorithm provided the following result:







- If the spread increases, trading opportunities appear (portfolio value increased as z-score was above 2) as on the sub period 03/11/2020-03/25/2020.

- If the spread drops, some losses appeared (like most of the time in period) so we have to sell both stocks to limit our exposure;
- If spread decreases and z-score fall under -2, trading opportunities also appear like on the sub period 10/17/2019-10/30/2019.

Below is discussed some of the cons to be improved on why using pair trading strategy;

- **Model Failure**

The most important thing to watch out for when pair trading is that the assumption that a correlation is real, which two stocks will return thereto correlated relationship after any divergence. simply because two stocks are correlated historically doesn't mean that they're going to still be correlated into the longer term . Identifying weak points during a correlation model are often extremely difficult, and therefore the potential failure of the market neutral assumption during a pair trade is an inherent risk of this sort of trading.

- **Commissions**

Another thing to enhance on in pair trading is that one pair trade leads to twice the commissions as a typical trade. For traders operating on relatively narrow margins, that difference in commissions are often the difference between a profit and a loss. Thus, most pair traders are forced to trade relatively high volumes, which needs more capital and may increase risk.

- **Price Filling**

Profiting in pair trading often relies on razor-thin margins and transactions with large share volumes, so there's significant risk that stock orders won't be filled at the specified price when opening positions during a pair trade. Even a difference of a couple of cents within the purchase or sale price of the stocks within the pair trade are often significant due to the high volume of those trades. Furthermore, this risk is amplified by the very fact that four orders, instead of two, got to be placed and filled at the expected price so as for the pair trade to be profitable.

Conclusion

Pair trading may be a powerful trading strategy supported the idea that highly correlated pairs of stocks or other financial instruments will return to their previous correlation after any divergences. The strategy are often employed over both intra-day and long-term timescales, although correlations could also be more or less powerful over different timescales. While pair trading can mitigate risk and permit traders to profit in any market conditions, correlation should be evaluated extremely carefully as any breakdown within the assumption of correlation can cause a pair trading strategy to fail

References

- Heydt, Michael. Mastering pandas for Finance. Packt Publishing Ltd, 2015.
- Elliott, Robert J., John Van Der Hoek*, and William P. Malcolm. "Pairs trading." Quantitative Finance 5.3 (2005): 271-276.
- https://analyticsprofile.com/algo-trading/pair-trading-part-1-code-distance-based-pair-trading-strategy-in-r/
- https://speedtrader.com/pair-trading/