Idea

Pairs trading is a statistical arbitrage strategy that aims at exploiting long-run relationships between asset prices. Finding these relationships is a challenging task since there is virtually a universe of stock to be explored. These stocks can be divided by sectors to make research easier, but in this case there may be still many unrelated assets in the same sector, or related assets in different sectors which would not be compared. In this work, I implement solution from the research paper *Enhancing a Pairs Trading strategy with the application of Machine Learning* (2020) by S. M. Sarmento and N. Horta. In their work, they propose to apply ordering points to identify the clustering structure (OPTICS) algorithm to cluster stocks to reduce the search space for possible pairs. They also propose a new approach to select thresholds, on which trading decisions regarding purchase/sale of stocks are made.

Literature review

Pairs selection

There are several approaches to select pairs of possibly related stocks. In the first one should explore all possible combinations of selected stocks[1]. The second one consists in grouping securities (i.e., by sector), and looking for combinations inside these groups[2]. The authors point out the first approach can be used to find more interesting pairs whereas the latter is less likely to result in spurious relations.

There are 3 widespread approaches to select assets for trade among all found pairs. The first one is finding such pairs for which the sum of squared distances between price series of their constituents is minimized[3]. However, for this condition zero spread would be optimal, even though it eliminates trading opportunities. Another way is to choose pairs with high Pearson correlation between prices[4]. However, high correlation does not mean time series have any equilibrium relationships. Finally, a pair can be chosen if its prices are cointegrated[5]. That means that the time series $S_t = Y_t - \beta X_t$, where $Y_t$and X_t are prices, is stationary.

Trading models

One of the most common trading models for pair of assets Y_t and X_t is the following[6]:

1. Calculate spread $S_t = Y_t - X_t$, mean μ and standard deviation σ during formation period;
2. Define the following thresholds: α_L for entering long position, α_S for entering short position and α_{exit} for exiting position;
3. If spread crosses α_L, buy Y and sell X. If it crosses α_S, sell Y and buy X. If α_{exit} is crossed, exit position.

The difficulty lies in determining optimal thresholds. The drawback of this model is that possible future spread values are not accounted for. Therefore, more robust models, which incorporate Machine Learning, have been explored.

[1] Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. European Journal of Operational Research, 259(2), 689–702.
[2] Do, B., & Faff, R. (2010). Does simple pairs trading still work? Financial Analysts Journal, 66(4), 83–95.
[3] Gatev, E., Goetzmann, W. N., & Rouwenhorst, K. G. (2006). Pairs trading: Performance of a relative-value arbitrage rule. The Review of Financial Studies, 19(3), 797–827.
[4] Chen, H., Chen, S., Chen, Z., & Li, F. (2017). Empirical investigation of an equity pairs trading strategy. Management Science.
[5] Vidyamurthy, G. (2004). Pairs Trading: quantitative methods and analysis, (vol. 217).John Wiley & Sons
[6] Gatev, E., Goetzmann, W. N., & Rouwenhorst, K. G. (2006). Pairs trading: Performance of a relative-value arbitrage rule. The Review of Financial Studies, 19(3), 797–827.

Data

I used daily price and return time series for 50 stocks obtained from *Yahoo finance* for the period 2010-01-04 – 2020-09-04. From these stocks, those with more than 5% of the missing values have been dropped, resulting in 44 stocks left, with missing values replaced by zero to avoid computational errors. The price data was split into training (2010-01-04 – 2018-12-31), test (2019-01-01 - 2019-12-31) and final dataset (2020-01-01 - 2020-09-04), which was used for profitability measurement. Training dataset corresponds to the formation period, over which pairs are formed.

Asset class and data frequency that have been used differ from the authors` - they used ETF data at 5-minute frequency. However, I was unable to obtain data at this frequency for such a large number of stocks, and could not find large enough dataset for ETFs. Also, the authors used 208 ETFs, but due to computational constraints I selected only a quarter of this size.

Initial pairs selection

First, I filtered training price dataset and percentage returns for the period 2010-01-05 – 2018-12-31 using AR(1)-TGARCH(1,1) model with Student`s residuals. I split the data into non-overlapping periods of 100 days and fit the model on each period. After that, I obtain standardized residuals of the model (further I refer to them as just 'the residuals'). I do not use sliding window for filtering at this stage due to large amount of computations – I need to filter data for each of 44 stocks separately.

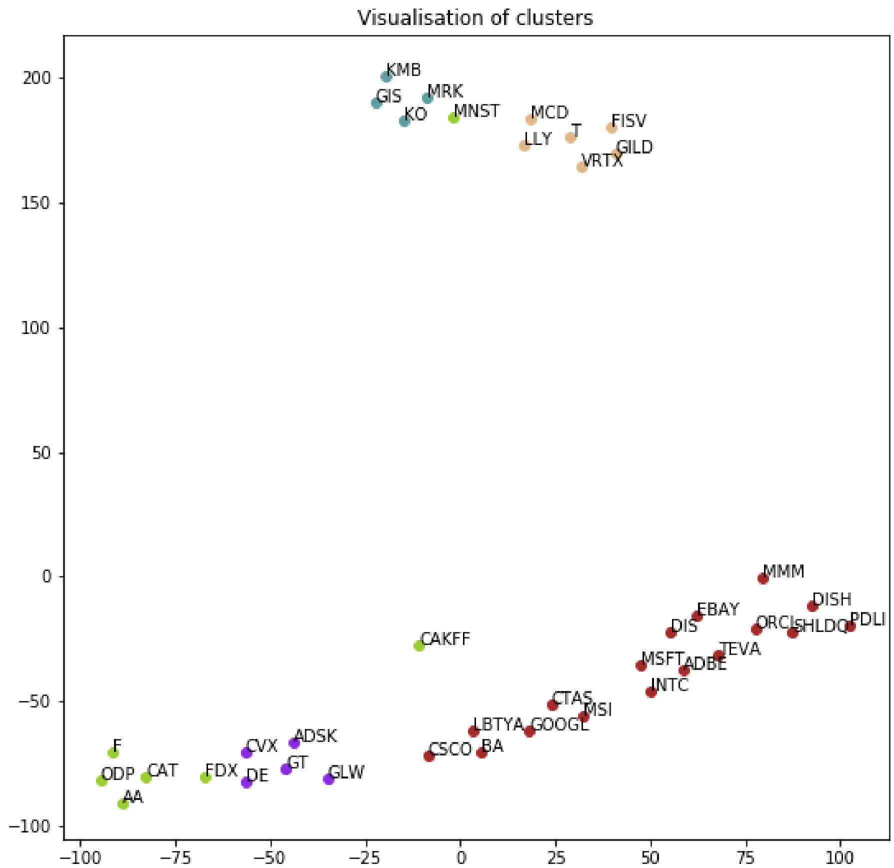
Next, I use filtered returns to cluster stocks. However, currently there are more than 2000 observations for each stock – too large dimensionality of data. Using so many features can result in overfitting and curse of dimensionality. To overcome this, authors suggest using PCA to reduce dimensionality of time series. PCA converts correlated variables into linearly uncorrelated ones. The history of returns for each security is mapped into several features – principal components. The authors refer to Berkhin[7] (2006), who does not recommend using more than 15 components. Our best model chooses small number of principal components – 2.

Ones the time series for each stock are mapped to lower dimensions, clustering is done on the transformed data. When choosing appropriate algorithm, the authors impose the following requirements: no need to choose number of clusters, no need to assign all stocks to clusters, accounting for outliers, no assumptions about shape of clusters, accounting for different densities of clusters and strict assignment of each security to one cluster only. Based on these requirements, density-based OPTICS algorithm is selected. To use it, I need to define minimum number of points to form a cluster.

I use different specifications of PCA and OPTICS to choose the best combination. For each number of principal components from 2 to 15, I fit OPTICS models with minimum number of points varying from 2 to 7. To select the best model, I use silhouette score, which measures the degree of similarity of an observation to its own cluster compared to other clusters. This score ranges from -1 to 1, with better models having higher score. Negative values mean that sample is assigned to incorrect cluster, zero value – that clusters are overlapping. My best model has 5 clusters and minimum 4 points to form a cluster, with silhouette score of 0.43.

---

[7] Berkhin, P. (2006). A survey of clustering data mining techniques. Grouping multidimensional data (pp. 25–71). Sringer.



To produce this picture, I used t-distributed stochastic neighbor embedding (TSNE) to make it more representative. Since it is another clustering algorithm, it may not map points to the PCA clusters exactly, so it can be that some points are of different color than their cluster.

I would like to compare the performance of the algorithm in 3 cases. First one uses unfiltered prices for trading and pairs selection, second one – filtered prices for both cases, third one uses filtered data to select pairs and unfiltered prices for trading. Using filtered and unfiltered training data, I group stocks according to the chosen clusters. In each group, those pairs are selected that satisfy the following conditions in this order:

1. At first stage, those pairs are selected in which prices exhibit cointegrating relationship. I test for this relationship using augmented Engle-Granger two-step cointegration test. This test has null hypothesis of absence of cointegration, which is rejected at 5% confidence level.
2. After that, I investigate if time series are mean-reverting. Hurst exponent can be used for this. For the price to be mean-revering, the exponent should lie in range (0, 0.5).
3. After mean-reverting time series are chosen, I calculate half life of mean-reversion. It shows how long it takes to mean-revert. This value should lie between 1 day and 1 year. This is done to ensure mean-reverting does not take too long.

4. Finally, pair spread should cross its mean at least 12 times per year, resulting on average in 1 trade per month, at least.

After checking these requirements, in total across all clusters there are 9 pairs based on unfiltered prices and 112 pairs based on filtered ones.

Further pairs selection and threshold determination

The following steps take long time to compute, so I randomly choose 14 filtered pairs. Thus, if more computational power could be obtained, the results would be more robust and representative.

I choose limited number of stocks to invest in, so first I need to determine most profitable pairs. For this, I use test datasets for filtered and unfiltered prices, accordingly for each case. On these datasets, I simulate trading with sliding window of size 200. However, to trade, I need to determine thresholds α_L and α_S, described in literature review, for each stock individually. I approximate them as follows: take filtered or unfiltered (depending on case) training dataset and calculate spreads on it. After that, determine percentage change in spread. Finally, calculate 0.2 and 0.8 quantiles and 0.1 and 0.9 deciles of this distribution of spread changes. Then simulate trading the pair on test dataset with given set of thresholds and calculate portfolio returns for all periods. It is calculated with respect to capital invested. Then select set of thresholds that gives highest mean return.

I assume I have 500 000$ of capital per equity, so assuming I choose 5 final pairs, I cannot spend more than $500\ 000\$ * \frac{1}{5} = 100\ 000\$$ per trade per one security. So I trade with all this sum but without leverage. Commission of 0.1% is taken into account. For each period, the trading algorithm with fixed thresholds is as follows:

1. Window of size 200 is taken. This size takes enough points for the model to be estimated more or less correctly. In case filtered prices are to be used for trading, both prices in pair are filtered with AR(1)-TGARCH(1,1). Spread $Y - X$ is calculated and is used to fit ARMA(1,1) model. This model is used to make a 1 step prediction of spread, which is used to calculate expected spread change.
2. If this change is less than the lowest threshold α_S, then I sell all of Y and invest all in X. If this change is larger than highest threshold α_L, I sell all X and invest in Y. So this is the strategy described in literature review. However, I trade with the limit that one trade is not more than 100 000$.
3. The portfolio is revalued on each day prior to any trades.

After mean returns and optimal thresholds are obtained, I choose only profitable pairs and sort them by profitability. There are 7 profitable stocks for unfiltered prices, 9 – for filtered and 11 for mixed dataset.

	return	pair
3	29.7411	MSFT-CSCO
5	29.6805	SHLDQ-EBAY
0	22.9931	GLW-ADSK
6	19.6898	C-GS
1	13.9339	CSCO-BA
4	12.6802	ORCL-GOOGL
2	3.95607	INTC-CSCO

	return	pair
5	4.25236e+16	MMM-CSCO
2	4.48005e+09	TEVA-ADBE
4	4.48002e+09	BA-ADBE
3	4.44627e+09	ADBE-DIS
1	4.07369e+07	MRK-KMB
0	6.40438e+06	CTAS-MSI
7	4.73295e+06	DISH-CTAS
8	650308	BAC-FITB
6	449330	TEVA-CTAS

	return	pair
1	39.7107	CTAS-MSI
8	39.7107	TEVA-CTAS
9	39.7107	DISH-CTAS
0	31.9033	LBTYA-DISH
5	24.1216	TEVA-ADBE
6	24.1216	ADBE-DIS
4	20.2446	DIS-DISH
10	18.2835	BAC-FITB
2	17.5786	MRK-KMB
7	13.8523	BA-ADBE

Left picture corresponds to returns on test set of pairs formed with unfiltered prices, the central one – with filtered ones, and the right – with mixed data. As we can see, there are quite many pairs formed with IT companies – Google (GOOGL), Cisco (CSCO), Microsoft (MSFT), Ebay (EBAY), Intel (INTC) and Adobe (ADBE). Sometimes they form pairs with companies from other sectors – for instance, Boeing (BA) or Sears Holdings (SHLDQ) from retail.

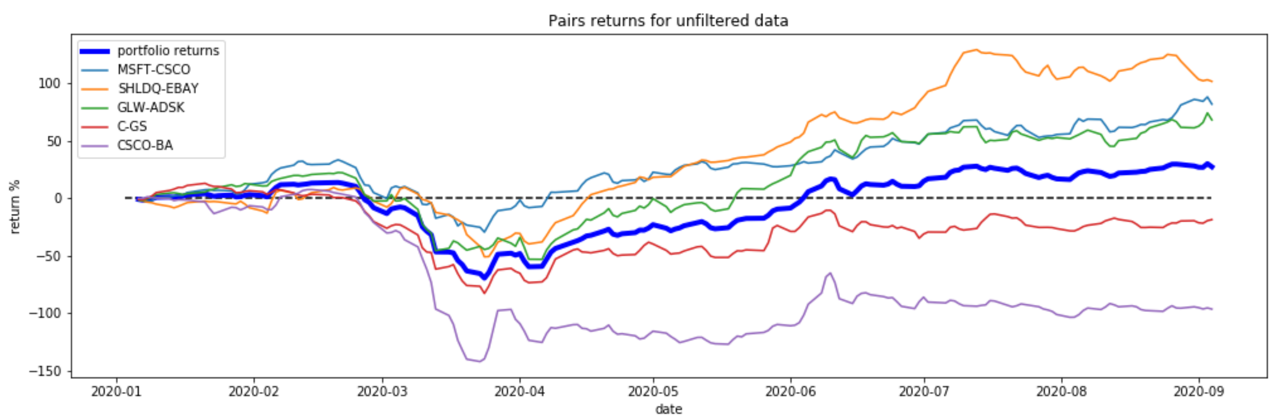
At this stage, we can notice huge mean returns for filtered prices. It turns out that for them, the algorithm works incorrectly, since filtered prices may have negative values as well as positive. This leads to incorrect calculation of spread – for instance, when subtracting negative filtered price, the spread increases. This may trigger incorrect trading decisions based on thresholds. But what also contributes to such unrealistic results is that purchase of stocks can occur at negative prices, thus contributing to overall return. Besides, it could also be that I choose to invest 100 000$ even for filtered portfolio, which is a large sum given that filtered prices have a mean of zero. Probably, more realistic results could be produced if smaller sum than 100 000$ had been taken, but it is the minimum required by the case.

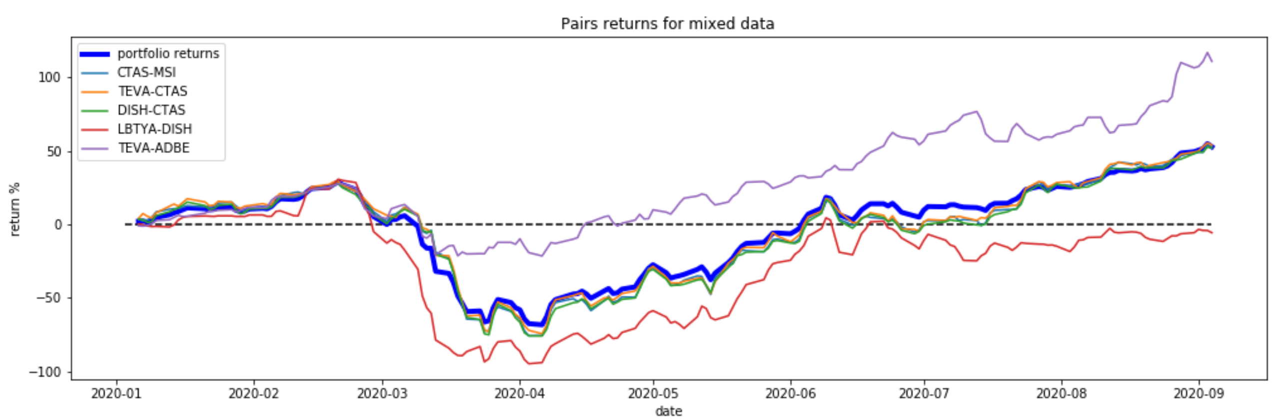
Pairs selected with mixed data show the highest (realistic) return. Notice that based on mixed data, MMM-CSCO pair, which was most profitable on filtered dataset, is found to be unprofitable and is not shown. In fact, the most profitable pairs of filtered dataset turn out to be less profitable on mixed dataset. We cannot compare pairs profitability on unfiltered and mixed datasets, since the pairs are different.

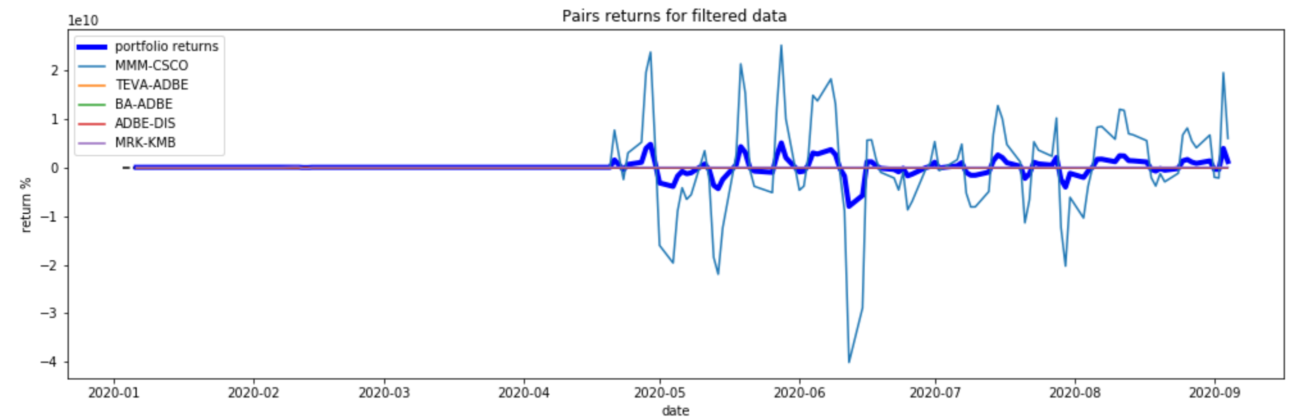
I choose 5 stocks from each group with highest returns for final portfolio.

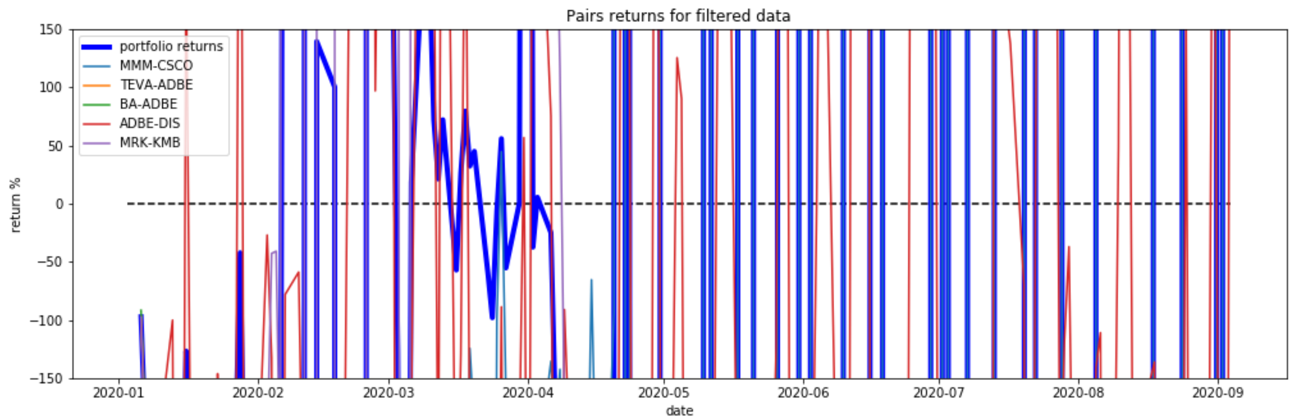
Backtest

I trade chosen pairs based on filtered, unfiltered and mixed data, on final dataset with the trading algorithm described above, and obtain the following results.









As we can see, filtered prices once again produced enormous, although not realistic, returns, which oscillate around zero. For unfiltered and mixed data, we can see a period of decline for all stocks from 2020-03 to 2020-06, linked to global turndown due to coronavirus. This decline is larger for portfolio chosen on mixed data. We can also note that in portfolio based on unfiltered data, two selected pairs C-GS and CSCO-BA actually showed negative return, reducing portfolio profitability. Still, portfolio return is positive at 27%. Portfolio based on mixed data seems to be more profitable at the end, although more volatile. It also has one pair which had negative return during the period.

Below I present statistics and ratios for portfolios, and in some cases, also for each pair.

Total cumulative returns

```
For filtered data
cumulative return for MMM-CSCO is 5947095638.0%
cumulative return for TEVA-ADBE is -193.0%
cumulative return for BA-ADBE is -194.0%
cumulative return for ADBE-DIS is 772.0%
cumulative return for MRK-KMB is -200.0%
cumulative return for portfolio is 1189419165.0%
```

```
For mixed data
cumulative return for CTAS-MSI is 48.0%
cumulative return for TEVA-CTAS is 48.0%
cumulative return for DISH-CTAS is 48.0%
cumulative return for LBTYA-DISH is -6.0%
cumulative return for TEVA-ADBE is 111.0%
cumulative return for portfolio is 50.0%
```

```
For unfiltered data
cumulative return for MSFT-CSCO is 82.0%
cumulative return for SHLDQ-EBAY is 101.0%
cumulative return for GLW-ADSK is 68.0%
cumulative return for C-GS is -19.0%
cumulative return for CSCO-BA is -96.0%
cumulative return for portfolio is 27.0%
```

Sharpe ratio

```
For filtered data
Sharpe ratio for MMM-CSCO is 0.03
Sharpe ratio for TEVA-ADBE is -12.69
Sharpe ratio for BA-ADBE is -12.3
Sharpe ratio for ADBE-DIS is -0.16
Sharpe ratio for MRK-KMB is -0.05
Sharpe ratio for portfolio is 0.03
```

```
For unfiltered data
Sharpe ratio for MSFT-CSCO is 1.05
Sharpe ratio for SHLDQ-EBAY is 0.76
Sharpe ratio for GLW-ADSK is 0.49
Sharpe ratio for C-GS is -1.19
Sharpe ratio for CSCO-BA is -1.74
Sharpe ratio for portfolio is -0.14
```

```
For mixed data
Sharpe ratio for CTAS-MSI is -0.22
Sharpe ratio for TEVA-CTAS is -0.22
Sharpe ratio for DISH-CTAS is -0.22
Sharpe ratio for LBTYA-DISH is -0.77
Sharpe ratio for TEVA-ADBE is 0.87
Sharpe ratio for portfolio is -0.12
```

Beta

```
For filtered data
beta for MMM-CSCO pair is -14001491.65
beta for TEVA-ADBE pair is -0.12
beta for BA-ADBE pair is -0.03
beta for ADBE-DIS pair is 40.8
beta for MRK-KMB pair is -73670.62
portfolio beta is -2815024.33
```

```
For unfiltered data
beta for MSFT-CSCO pair is -0.23
beta for SHLDQ-EBAY pair is 0.07
beta for GLW-ADSK pair is -0.3
beta for C-GS pair is -0.56
beta for CSCO-BA pair is -1.09
portfolio beta is -0.42
```

```
For mixed data
beta for CTAS-MSI pair is -0.85
beta for TEVA-CTAS pair is -0.85
beta for DISH-CTAS pair is -0.85
beta for LBTYA-DISH pair is -0.7
beta for TEVA-ADBE pair is -0.29
portfolio beta is -0.71
```

Maximum drawdown

```
For filtered data
maximum daily drawdown for MMM-CSCO is 1343868129.42%
maximum daily drawdown for TEVA-ADBE is 0.0%
maximum daily drawdown for BA-ADBE is 0.0%
maximum daily drawdown for ADBE-DIS is 127.09%
maximum daily drawdown for MRK-KMB is 9934607.76%
maximum daily drawdown for portfolio is 268773605.38%
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
maximum weekly drawdown for MMM-CSCO is 1816766099.22%
maximum weekly drawdown for TEVA-ADBE is 0.0%
maximum weekly drawdown for BA-ADBE is 0.0%
maximum weekly drawdown for ADBE-DIS is 127.09%
maximum weekly drawdown for MRK-KMB is 13501852.74%
maximum weekly drawdown for portfolio is 363353199.9%
```

```
For unfiltered data
maximum daily drawdown for MSFT-CSCO is 0.49%
maximum daily drawdown for SHLDQ-EBAY is 0.33%
maximum daily drawdown for GLW-ADSK is 0.47%
maximum daily drawdown for C-GS is 0.45%
maximum daily drawdown for CSCO-BA is 0.38%
maximum daily drawdown for portfolio is 0.34%
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
maximum weekly drawdown for MSFT-CSCO is 0.58%
maximum weekly drawdown for SHLDQ-EBAY is 0.54%
maximum weekly drawdown for GLW-ADSK is 0.8%
maximum weekly drawdown for C-GS is 0.72%
maximum weekly drawdown for CSCO-BA is 1.29%
maximum weekly drawdown for portfolio is 0.75%
```

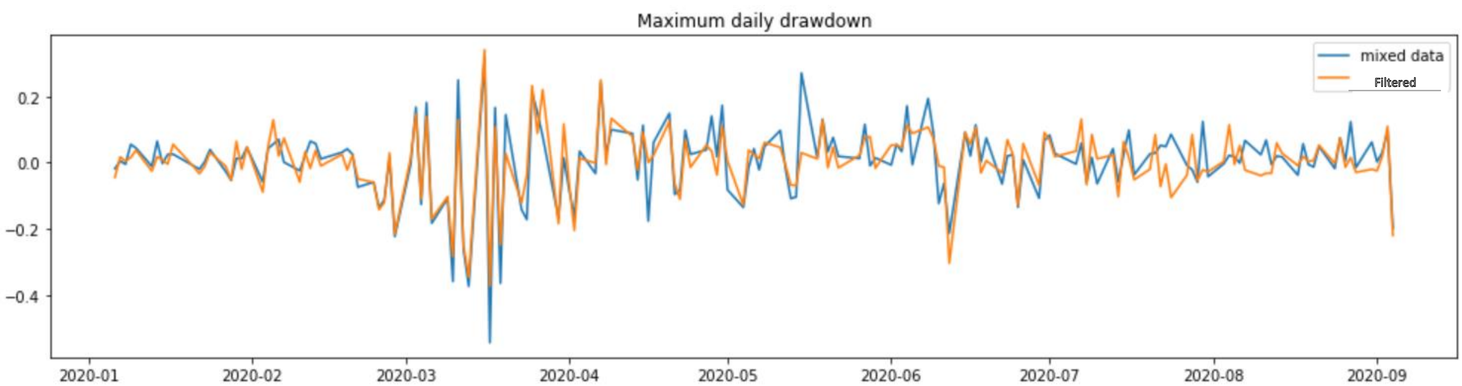
```
For mixed data
maximum daily drawdown for CTAS-MSI is 0.43%
maximum daily drawdown for TEVA-CTAS is 0.43%
maximum daily drawdown for DISH-CTAS is 0.43%
maximum daily drawdown for LBTYA-DISH is 0.35%
maximum daily drawdown for TEVA-ADBE is 0.6%
maximum daily drawdown for portfolio is 0.32%
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
maximum weekly drawdown for CTAS-MSI is 0.97%
maximum weekly drawdown for TEVA-CTAS is 0.97%
maximum weekly drawdown for DISH-CTAS is 0.97%
maximum weekly drawdown for LBTYA-DISH is 1.05%
maximum weekly drawdown for TEVA-ADBE is 0.62%
maximum weekly drawdown for portfolio is 0.74%
```

Accumulated return to maximum drawdown

```
For filtered data
accumulated return to maximum drawdown for MMM-CSCO is 3.05
accumulated return to maximum drawdown for TEVA-ADBE is -48.28
accumulated return to maximum drawdown for BA-ADBE is -48.41
accumulated return to maximum drawdown for ADBE-DIS is 4.35
accumulated return to maximum drawdown for MRK-KMB is -0.0
accumulated return to maximum drawdown for portfolio is 5405213094.37
```

```
For unfiltered data
accumulated return to maximum drawdown for MSFT-CSCO is 226.47
accumulated return to maximum drawdown for SHLDQ-EBAY is 153.26
accumulated return to maximum drawdown for GLW-ADSK is 165.51
accumulated return to maximum drawdown for C-GS is -9.6
accumulated return to maximum drawdown for CSCO-BA is -31.9
accumulated return to maximum drawdown for portfolio is 123.36
```

```
For mixed data
accumulated return to maximum drawdown for CTAS-MSI is 280.83
accumulated return to maximum drawdown for TEVA-CTAS is 280.83
accumulated return to maximum drawdown for DISH-CTAS is 280.83
accumulated return to maximum drawdown for LBTYA-DISH is -2.17
accumulated return to maximum drawdown for TEVA-ADBE is 357.76
accumulated return to maximum drawdown for portfolio is 227.67
```



Rachev ratio

```
For filtered data
Rachev ratio is 1160050.9
```

```
For mixed data
Rachev ratio is 0.19
```

```
For unfiltered data
Rachev ratio is 0.2
```

Value at Risk (VaR) and Expected Shortfall Calculation

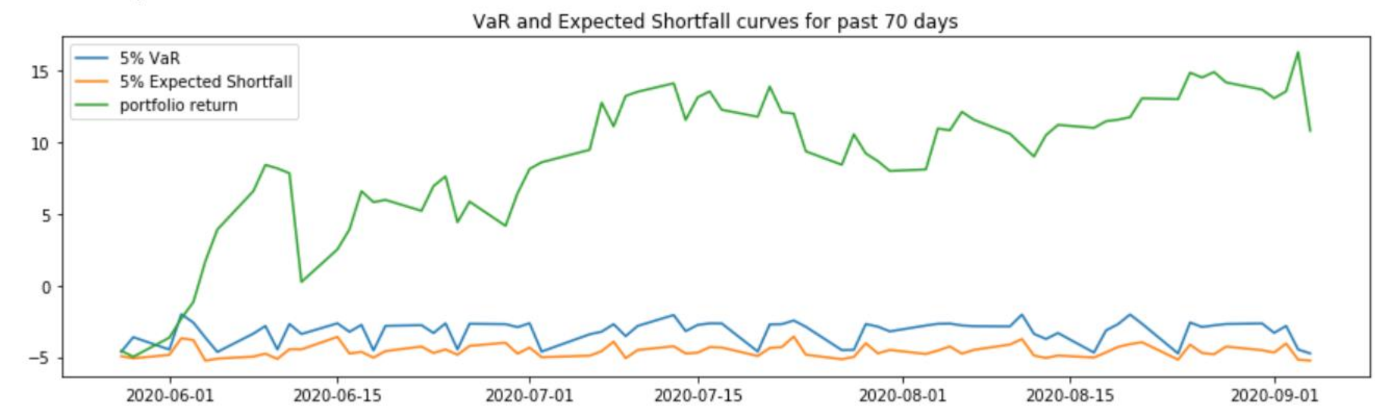
I calculate VaR based on papers *Portfolio Optimization via Pair Copula-GARCH-EVT-CVaR Model* (2011) by L. Deng, C. Ma and W. Yang, and MSc thesis *Value at Risk Estimation. A GARCHEVT-Copula Approach* (2013) by N. K. Bob. VaR is estimated with sliding window of size 200 and is calculated on the basis of portfolio returns. The steps are as follows:

1. For cases when all data used is unfiltered or mixed, I filter pairs returns using AR(1)-TGARCH(1, 1) with Student`s errors and take standardized residuals. Then I make one-step prediction and save its conditional mean and variance.
2. Then I model distribution of standardized residuals (for unfiltered and mixed dataset) or returns (for filtered data). I choose Peak Over Threshold (POT) method for modelling tails, and Student`s

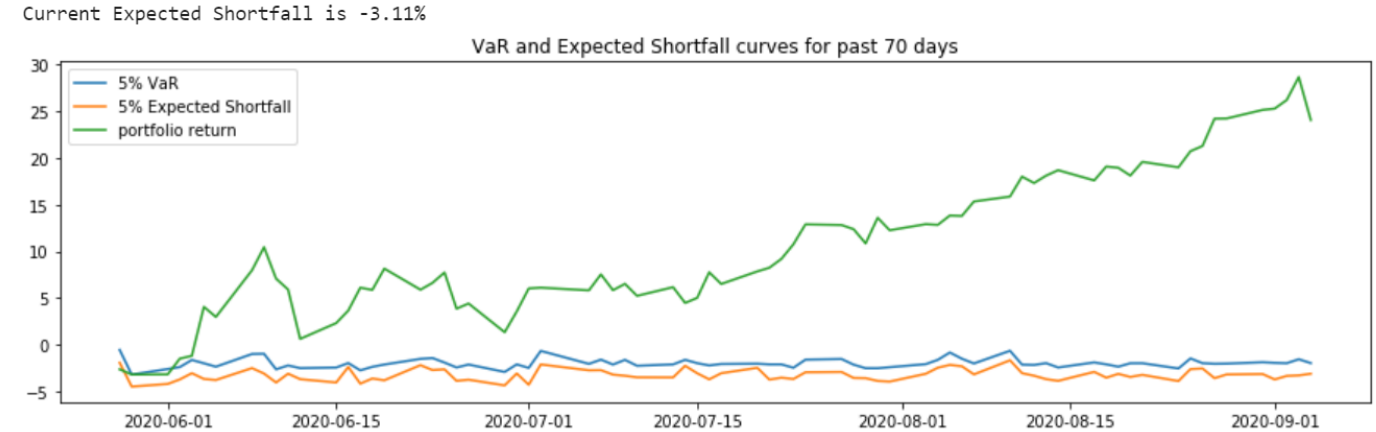
distribution for modelling interior part. To model left tail, first I choose observations that lie below 0.1 decile of the distribution and model them with generalized Pareto distribution. I fit this distribution to observations and save parameters. Then, I apply cumulative density function (CDF) of this distribution to transform observations. I do the same with the right tail, except that first I reverse distribution of returns/residuals by multiplying it by -1.

3. I fit Student`s distribution to the interior, save parameters and transform with CDF.
4. Having down this for returns for all pairs, I fit copula. I have chosen D-Vine copula, as recommended by L. Deng, C. Ma and W. Yang (2011). Samples of size 300 are generated from the fitted coplula for each pair. It would be better to obtain significantly larger generated sample, but in my case I have computational constraints.
5. Then I use inverse distribution functions with parameters fitted in step 2, which correspond to tails and interior, to transform copula samples back to standardized residuals and returns.
6. If standardized residuals are used, they are multiplied by square root of conditional volatility of prediction from step 1, by (degrees of freedom – 2), divided by degrees of freedom and added to conditional mean of prediction to get back to returns.
7. Finally, portfolio return is obtained by averaging all pairs returns, and VaR is obtained as 0.05 quantile of this distribution.

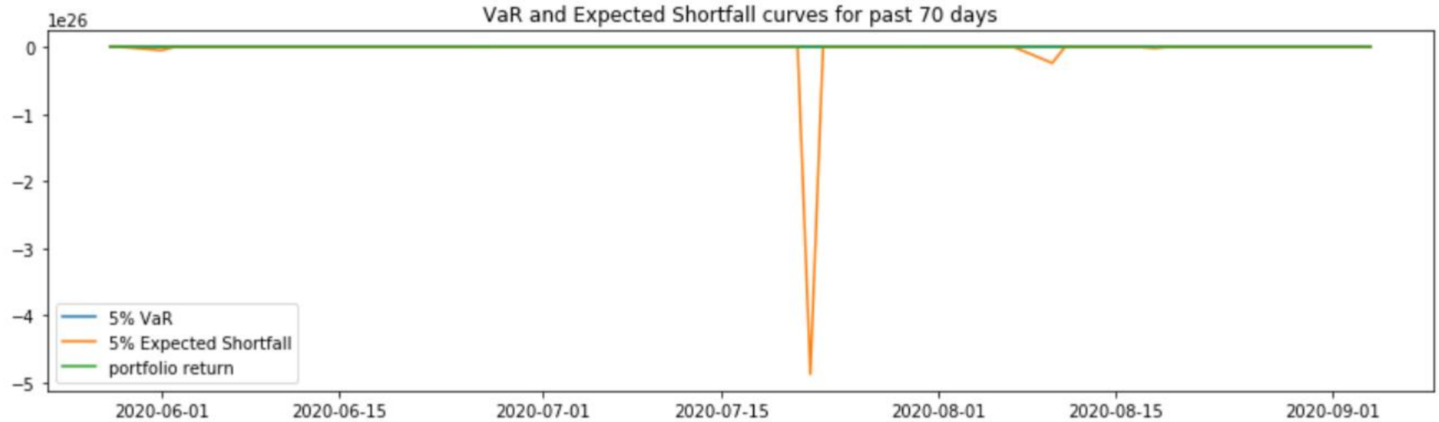
```
For unfiltered data
Current VaR is -4.74%
Current Expected Shortfall is -5.24%
```




```
For mixed data
Current VaR is -1.97%
Current Expected Shortfall is -3.11%
```



```
For filtered data
Current VaR is -163.68%
Current Expected Shortfall is -1.7647500969544796e+23%
```



Conclusion

The model proposed by authors generates significant returns for stocks on daily data frequency. To identify best pairs and make best trading decisions, it is recommended that pairs should be chosen on filtered prices, but trading should be done on the basis of real prices. Resulting portfolio would have highest cumulative return, lowest beta and maximum daily drawdown and VaR compared to the other two portfolios.