Analysis of pair trading with financial market data

Marcell P. Granát

2021 09 09

Contents

1 Introduction **3**

2 Literature review **3**

2.1 Pairs trading with partial cointegration [Clegg and Krauss, 2018] 3

2.2 Are pairs trading profits robust to trading costs? [Do and Faff, 2012] 4

2.3 The profitability of pairs trading strategies: distance, cointegration and copula methods [Rad et al., 2016] 5

2.4 Selection of a Portfolio of Pairs Based on Cointegration: A Statistical Arbitrage Strategy [Caldeira and Moura, 2013] 5

2.5 On the determinants of pairs trading profitability [Jacobs and Weber, 2015] 6

2.6 On the presistence of cointegration in pairs trading [Clegg, 2014] 6

2.7 On the Power and Size Properties of Cointegration Tests in the Light of High-Frequency Stylized Facts [Krauss and Herrmann, 2017] 7

2.8 Combining Non-Cointegration Tests [Bayer and Hanck, 2013] 8

3 Empirical usage of traditional econometric tools **8**

3.1 Explanatory data analysis 8

3.2 Engle-Granger method 10

3.3 Johansen-test 10

3.4 Engle-Granger method with rolling window 12

3.5 Johansen test with rolling window 12

4 Multivariate simulation **14**

4.1 Data Generating process 14

4.2 Performance of Johansen-test 14

4.3 Comparison of tests performed on 2 or 3 time-series 14

5 Appendix: R codes **17**

Abstract

Pairs trading is a widely known and applied trading strategy. Intuitively, it is about comparing stock-pairs, which seem to move frequently together. If at any time their pairwise relative price differs from its ordinary level, one can expect it is going to revert to that. More specifically a given econometric test is committed to identify the cointegration relation, but in the enormous literature of pairs trading, this process is performed mainly with pairwise tests. The aim of this paper is to argue for an extension of this strategy based on multivariate cointegration tests and contribute to its literature with a Monte Carlo simulation. The presented simulations confirms the relevance of multivariate tests, because if one variable is ommited in the cointegration test, the result van be misspecified.

Keywords— Pairs trading, Cointegration

List of Tables

List of Figures

1 Classification of the core literature. 3
2 Time-series used in this study 9
3 First difference of the time-series 10
4 Results of Engle-Granger method 11
5 Results of Engle-Granger method with rolling window 12
6 Results of Johansen-test with rolling window across time 13

1 Introduction

Pairs trading is a widely known and applied trading strategy. Intuitively, it is about comparing stock-pairs, which seem to move frequently together. If at any time their pairwise relative price differs from its ordinary level, one can expect it is going to revert to that. More specifically a given econometric test is committed to identify the cointegration relation, but in the enormous literature of pairs trading, this process is performed mainly with pairwise tests. The aim of this paper is to argue for an extension of this strategy based on multivariate cointegration tests and contribute to its literature with a Monte Carlo simulation. The paper is structured as follows: section 2 introduces the most relevant literatures to my paper. Section 3 is about the usage of traditional econometric tools, and section 4 describes the chosen research question.

2 Literature review

Many papers describe pairs trading as a profitable strategy, which is based on only the historical data of the time-series. This questiones the hyphothesis of effective markets. Another interesting research quetion why prices are cointegrated in the real life (why do prices frequently move together?).

Identifying cointegration is complex problem. Most of the literature uses only liqued stocks and applies traditional econometric tools (Engel-Granger method, Johansen-test). The most referred paper [Gatev et al., 2006] also applies a simly methodology. However, its profitablity was questioned later by other papers (after calculating with the increasing trading cost) and researchers focus on more complex formulas to identify cointegration relationship more confidently. Thus, the literatures can categories based on their focus: economic interpretation or methodology. However, most papers relies on both. This is visualised on Figure 1.

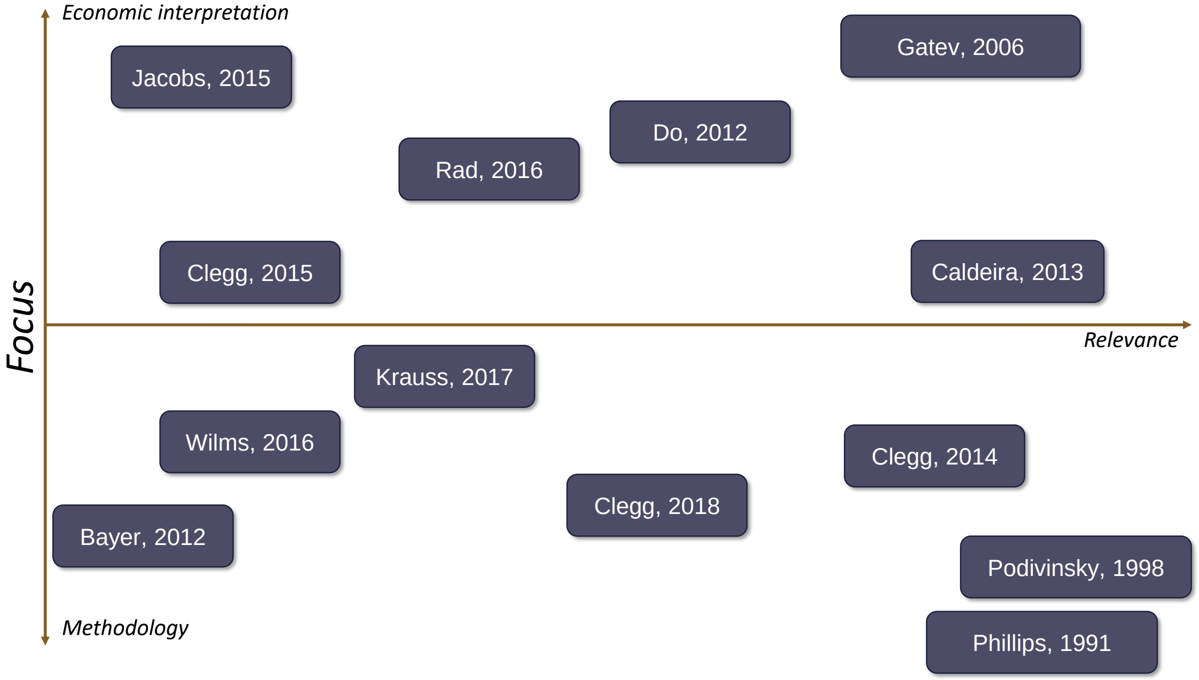


Figure 1: Classification of the core literature.

2.1 Pairs trading with partial cointegration [Clegg and Krauss, 2018]

There is reason to believe that cointegration may not be the most appropriate model for pairs trading. Companies often experience idiosyncratic shocks that are permanent in nature. (The reason can be technological change, development of new markets, legal action etc.) Clegg (2014) shows that poverty of being cointegrated is not persistent. This paper introduces the concept of partial cointegration (PCI).

PCI allows the residuals to have a mean-reverting and random walk components. For this modelling, there is an R package called partialAR. *A key statistic of a PAR model is the proportion of variance attributable to mean-reversion.* The system is identifiable, the estimation is based on *maximum likelihood estimation of the associated Kalman filter.*

The paper contains a simulative and empirical analysis. As a benchmark distance method and two cointegration-based strategies (CI1 is testing with ADF and Johansen, while CI2 is based on Phillips-Perron-test (see Clegg, 2014)) are performed. For CI1, CI2 and CPI the best pairs to trade are chosen based on Sharpe ratio.

The conclusion from the simulative analysis is that PCI outperforms CI, but the advantage disappears if the proportion of variance attributable to mean-reversion or the AR(1)-coefficient is high. Testing for optimal opening and closing threshold also committed.

Evaluating the performance shows that **PCI outperforms the other methods**, (suggested R package: PerformanceAnalytics) its skewness and kurtosis are higher. *Fama-French three-factor model* is also performed. Sub-period analysis shows that the strategy performed significantly better at the time of a crisis, but in **recent years its profitability started to decline (assumed that the reason is its growing popularity)**.

The authors suggest using PCI in other economic contexts as possible further research (other financial markets or macroeconomic).

2.2 Are pairs trading profits robust to trading costs? [Do and Faff, 2012]

Authors examine the impact of trading costs on pairs trading profitability. These costs are **commissions, market impact, short selling fees**. After controlling for these, the approach by *Gatev* (2006) loses its profitability. Pairs trading remains profitable at portfolios formed within industry groups. This advantage is formerly presented in *Do and Faff* (2010). A bit more precise explanation is mentioned from *Hameed, Huang, and Mian* (2010): *"Cancel out common factors and highlight firm-specific shocks that are expected to mean revert."*

Do and Faff examined 29 portfolios, which differ in the way of how pairs are selected (description is in Table 1). One key difference is that matcihng is unconditional, major sector or industry. AN instrsting feature is that beside SSD, the **number of zero crossing (NZC)** is also calculated to chose pairs. The last dimension of portfolio formation is the number of pairs in the protfolio and the sorting order of the SDD and NZC. The dataset is from 1963 to 2009 U.S. stocks, used only liquid stocks. Return from portfolios is the before-cost monthly returns, divided by the number of pairs in the portfolio.

The authors criticise the PTS literature because the papers often do not take care of the trading cost. These papers control for commissions based on *Jones's* (2002) work. Since the costs vary at different investors (intitutional investors face lower commissions.), they can be only an estimation. Doing this, the main conclusion is that PT is not profitable after transaction cost. An interesting finding is that at a deep market[1] - even at strong convergence - the transaction cost is not covered. Next to the commissions, the market impact is also an important feature. Short selling constrains has 3 types: Inability to short at the desired time; the cost of the loan; borrowed stock can be recalled. Hence, the authors calculate with 1% as a cost for shorting.

As a result of the empirical analysis, the 29 portfolios ensure 93 bps excess on average, with 1.05 Sharpe ratio. But after controlling for transaction costs, the profitability is reduced substantially, but the intraindustry portfolio remains profitable. Liquidity decreases the reversal effect and the profitability of PT. The profitability of PT is declining recently, but during bear market - when prices in the market fall by 20% or more - it improves.

As a benchmark, the authors mention the paper from *Hameed, Huang, and Mian* (2010), who investigate the efficiency of buying industry group losers, and selling industry group winners. As a result, PT ensures lower

[1] Traded with high volume and small spread.

risk and lower return than buying extreme losers and selling extreme winners, and so be a preferable strategy for institutional investors.

2.3 The profitability of pairs trading strategies: distance, cointegration and copula methods [Rad et al., 2016]

History of PT starts in the 1980s at Morgan Stanley. The really strong computation capacity requirement is mentioned in the article (possible source to refer). The aim of the paper is to analys the profitability of pairs trading, specially focus on the trading cost. Beside the distance method (DM) - presented formerly by *Gatev (2006)* - the authors examine two more sophisticated methods: cointegration and copula. As the paper writes down, DM is benchmark. The explanation behind the hyphothesis that more complex model is required to keep the profitability is that using an arbitrage strategy becomes popular by the time and then we need a new method to find opportunities.

Trading cost is time-varying in line with Do and Faff (2012). Data are the US equity market from 1962 to 2014 (13 216 days, 23 616 stocks) - Consistent with that of Do and Faff (2012). - , but only the liquid stocks. This is done by removing the bottom decile stocks, in terms of market cap, in each formation period. The trading period is 6 month, formation period is 12 month, start in each month. DM is the same as we saw at *Gatev* (2006), hence the pairs with the lowest SSD are chose to trade. At the C method, ECM is also presented, and the signal is generated with a normalized spread, similarly to the literature ±2 standard deviation is the bound. But for the computation efficiency, in the pairs selection SSD is also used. Copula method is also described widely[2].

Measuring the profitability the authors introduce the return on employed capital (average return from the traded pairs). The performance of the 3 methods are the following (Table 3): DM has higher monthly return (after transaction cost is counted in), but C is almost as good as that, only the copula method underperforms strongly. The weaker performance of the copula method can be attributed to the high proportion of unconverged trades. C has a higher Sharpe-ratio, but as we move closer to recent years (5-year rolling window) the performance of the three strategies becomes more similar. PT is a market-neutral strategy, which means it is an alternative investment strategy for reducing market risk. DM and C perform better during a crisis. The number of trading opportunities with DM and C decline, but it remains stable with the copula method, so it would be reasonable to take care more about that.

To sum up, I would highligth comparison of the three different methods from this article, and the conclusion that the more complex model do not neccesserly perform better, but as PTS is an arbitrage strategy, innovation is required to keep the profit.

2.4 Selection of a Portfolio of Pairs Based on Cointegration: A Statistical Arbitrage Strategy [Caldeira and Moura, 2013]

“Since future observations of a mean-reverting time series can potentially be forecasted using historical data, this literature challenges the stylized fact in financial economics which says that the stock prices shall be described by independent random walk processes; what would automatically imply no predictability in the stock prices.[3]” “Desired characteristics of this class of strategies are market neutrality (low market correlations), and signal generation based on rules rather than fundamentals.” [4] This paper the methodology of cointegration employed, and as the authors profess, this article is a pioneer in this aspect. The papers also contain the story of the first usage (Morgan Stanley).

As cointegration test, the authors choose Johansen test and Engle-Granger. The trading signal is generated by normalized spread, when its absolute value reaches 2, then the positions are opened. Data are daily closing prices of the 50 stocks with the largest weights in the Ibovespa index from Sao Paulo Stock Exchange at the

[2] The methodology is not clear at the moment, but as I know I will learn about it in this semester.

[3] Authors mention the following articles as the core of this topic: Poterba and Summers (1988); Lo and MacKinlay (1990); Gatev et al. (2006), Elliot et al. (2005); Perlin (2009) and Broussard and Vaihekoski (2012)

[4] "According to Fung and Hsieh (1999), a strategy is said to be market neutral if its return is independent of the market’s relative return."

beginning of each trading period. The trading period lasts 4 months, and 1 month trading after that. There is no restriction for industries at pair forming.

A huge advantage of the article is that the sample is split to training and testing. Transaction cost is also included in the calculations (0.1% as the brokerage fee, 0.05% slippage, 0.2% as the rental cost for shorting).

Pairs that passed the cointegration tests (on average 90 per period) are then ranked based on Sharpe-Ratio. The 20 pairs with the highest SR are used for trading in the training period.

The cumulative net profit from the four years of rolling window out-of-sample tests was of 189.29%, with an annual mean of 16.38%. Besides, the pairs trading here implemented showed relatively low levels of volatility and no significant correlation to Ibovespa, confirming its market neutrality. Next to the SR and cumulated profit, Maximum Drawdown is also reported, and all of these show that PTS was less profitable in 2008. The authors also performed a reality check with the bootstrap method, which suggest that this wasn't caused by a structural change in the reversion pattern.

2.5 On the determinants of pairs trading profitability [Jacobs and Weber, 2015]

A few must mention the contribution of this paper is that beyond a US dataset, the authors use an international one, to explorer the effect of the market development on the profitability of pairs trading. From the aspect of the method, the paper is not as complex as the former ones, the simple DM is used.

The international analysis is performed on daily stock prices (2000-2013). The small and illiquid stocks are excluded, but there are not any other regularization in the pairing (resulting in 200 million pairs). For the trading simulation, the 100 pairs with the smallest distance are chosen [5]. The trading signal is twice the historical standard deviation, and for the case of prices do not converge, one month is set as a stop-loss bound. As a measurement of performance, the fraction of converging pairs, monthly return and Fama and French three-factor model is reported. The key conclusion from this analysis is that pairs trading has a robust performance. The returns are about a quarter higher in emerging markets than in developed markets. Profitability is much higher in large markets. This finding is explained with the information overload and limited visibility of individual pairs. This hypothesis is also confirmed the following: The stock market size relative to the GDP is positively correlated with the profitability, while the average industry market share is negative.

The other used dataset contains only US stocks, but a much longer interval (1960-2008). Here I would highlight the fact, that 40% of the trading signal is generated within a day. As the Fama and French model fails to explain the profit, focus on **event-time** test is considered. This one is the second huge contribution of this study. And similarly to the literature, the authors also report the negative trend of the profitability of pairs trading. Exploring the effect of news, the article differentiates idiosyncratic (firm-specific) and common (affect both pair constituents, like macroeconomic shocks) news. The idiosyncratic news is negatively correlated with the profitability, while the common is positive.

At last, the study focuses on time-varying investor attention. The hypothesis was that when higher returns are available through limited attention. Like on Fridays or before holidays, when PT yields a higher return, based on the results. Similarly, profitability is higher, when more trading signal is generated, during a recession, when google searches for the stock-market are more frequent or when the volatility is higher.

2.6 On the presistence of cointegration in pairs trading [Clegg, 2014]

The paper starts with a thorough description of the Engle-Granger cointegration test. The only difference with the one in the book, that here log form is used, probably because logdiff is the most widely used function to transform stock-prices into stationary variables. The following part is a critic about the pairs trading literature. Many times the authors suppose that if two stocks are cointegrated in a year, then this relationship will be held in the next period. Clegg's empirical result shows that it is wrong, and cointegration does not seem to be persistent. For empirical work, the paper reaches the S&P 500 from 2001 until 2013, but only the integrated prices.

[5] Consistent with Gatev (2006).

The following part is about comparing different unit root test on the simulated dataset, to find the most powerful one. A side note is that at least 250 observation is recommended to perform a test, but 500 is ideal. Tests are performed with different rho and variance of the residuals. As a conclusion, we can say that the Phillips-Perron test is the most suggested one.

To formulate a critique at this point, I have to mention the lack of descibtion about the trend and constant term in the used unit root test. Clegg simply skip this important detail, but using a wrong specified test can lead to false results. Even when the used R funtions are listed, tseries::adf.test is written. But this function uses a trend-filter and chose the lag parameter automaticly based on the length of the time-series.

This comparison of the unit root tests is also performed on the constituents of the S&P 500. Only 5% of the tested pairs are cointegrated, and in the next year, 5% of these relations held. χ^2-test is used for the hypothesis that being cointegrated in a year is independent or not from being cointegrated in the previous year. In some years this null-hypothesis can be rejected, so little evidence is found. This test is performed with different unit root tests, with two-year intervals and on unlogged prices. However, the conclusion was not different. A side note in this analysis is that during high price volatility more cointegrated pairs are found (like in 2008)[6].

In the remaining part, Clegg tests if there is a short-term persistence of cointegration. The conclusion is here is that the hypothesis that cointegration is a short-term phenomenon is rejected[7].

To sum up, I would highligth the suggestion of Phillips-Peron test, the idea of testing the persistance of the relationship and the negative results about the profitability.

2.7 On the Power and Size Properties of Cointegration Tests in the Light of High-Frequency Stylized Facts [Krauss and Herrmann, 2017]

Krauss and Herrmann focused on the performance of several cointegration tests. In contrast with the literature of PT the authors used a high-frequency (one-minute) trade data. The used dataset is the DAX 30 (2014), containing 3.8 million observations.

Analysing the cointegration residuals of this dataset non-normality, ARCH effects, jumps and evidence of further nonlinearities are identified. The used tests are: (non-normality) Jarque–Bera test, (ARCH effects) Box test, Engle's ARCH test, (Nonlinearity) BDS test, Tsay test, Luukkonen test, Teräsvirta test, (Jumps) BNS test. I would highlight the advantage that at this point the used R packages are listed.

The way how the authors simulate stock prices (*artificial high-frequency price series* based on stationary bootstrap) having the same stylized features as the dataset is another consideriable contribution of this paper.

In addition, I would highlight that the authors published all the *R packages*, which contain the used unit root tests (*tseries, egcm* (looks extremly useful, since it provides p-values of several tests), *urca, vars, fArma*).

The followng DGPs as cointegration residuals are presented in this study, to measure the effect of the previously mentioned anomalies on size and power: AR(1), AR(1)-GARCH(1,1)[8], MR(3)-STAR(1)-GARCH(1,1)[9], MR(3)-STAR(1)-GARCH(1,1) with reversible jumps and with non-reversible jumps.

An important conclusion here is that non-normality and ARCH effect do not have a significant reduction impact on the size and power properties of the cointegration tests. Hence in the presence of these anomalies, cointegration tests still remain useful.

Turning to the case of jumps (*jump is identified as innovation outside of the one percent or outside of the 99 percent quantile*) the authors claim that the reversible jumps (*persists for only one period before the process bounces back*) do not have an adverse effect on the power and the size. In contrast with the previous, non-reversible jumps significantly reduce them with an increasing rate of the occurence and jump size.

[6] Consistrent with the findings in the literature.

[7] The meaning of "cointegration data set" is not clear for me at this point.

[8] To reflect the conditional heteroscedasticity property.

[9] To reflect the nonlinear property.

The most important conclusion of the paper is that **PGFF and the PP tests exhibit the most favorable** power properties in all settings.

2.8 Combining Non-Cointegration Tests [Bayer and Hanck, 2013]

The aim of the study is to control the size of cointegration test by combine information from individual tests.

An advantage of the paper is that the code (STATA and Matlab) of the suggested method is published online.

In the intro the authors present the most famous cointegration tests (*Engle-Granger, Johansen and error-correction based*), and claim the issue that the decision based on them are not consistent. At this point *Gregory et al.* (2004) is refered, who states that the p-values of the mentioned tests are not correlated. Although this is a relevant point, I would change the used tool to Spearman-correlation. The writers continue with real life examples where some test reject and some tests fail to reject of the null hyphothesis of non-cointegration.

To reach a more robust decision strategy combining the individual tests are based on different aggregation methods. First is the *Fisher*'s (1932) χ^2 test. [If I read well, then a crucial assumption of the aggregation is that the methodology of the indiviual tests must differ, to ensure the independence of the decisions.] Second is the **Union-of-Rejections tests**: reject if one of the individuals rejects.

The power of the both of the above discussed tests are not avaiable in closed form. Hence the study rely on simulations.

Figure 1 shows the power to a given R^2 between the two series in case large sample. Since the location of the highest power value differ at the different tests, pretest to choose an optimal test is suggested by authors [Personally, I think this idea could be implemented in my research]. The problem here is that correct estimation of R^2 is not performable and combinated decision rule outperforms the individuals in almost every case, but never significantly lower.

In case of small samples, the paper suggests a bootstramp methodology, refer to many other studies, which formerly claimed that this tool is relevant to increase the power of the test.

In the next chapter the writers perform a Monte Carlo simulation and the power of the tests are attractive (Here, four different and reasonable DGPs are involved).

To sum it up, the article relies on complex mathematical proofs and the conclusions are ineffective when focusing on the PTS. I would highlight the methodology of estimating the power of individual tests and the evidence of the utility test combination.

3 Empirical usage of traditional econometric tools

In this section I introduce the traditional econometric tools used for pairs trading, and describe their empirical performance on real life dataset. The components of the presented methodology are referring to the techniques from the literature.

3.1 Explanatory data analysis

Engle-Granger method is a simple way to test cointegration in the bivariate case. Cointegration is diagnosed if the two tested series are integrated in the same order and a linear combination of them exist, which has an integration order of the original non-stationer series minus one [Kirchgässner and Wolters, 2007]. The most common is when the tested stock prices are I(1) and their linear combination is stationer.

The used stock prices are presented in figure 2. For a first glance, there is a high chance that some cointegrated pairs can be found in this set of series. To commit the tests the first step is to check the time-series integration order. For this purpose, I use ADF-test with a significance level of 5%. As a result, it is concluded that all the series are I(1) if any of their bivariate linear combinations is stationer, then cointegration is diagnosed. The first difference in the stock prices is shown in figure 3.

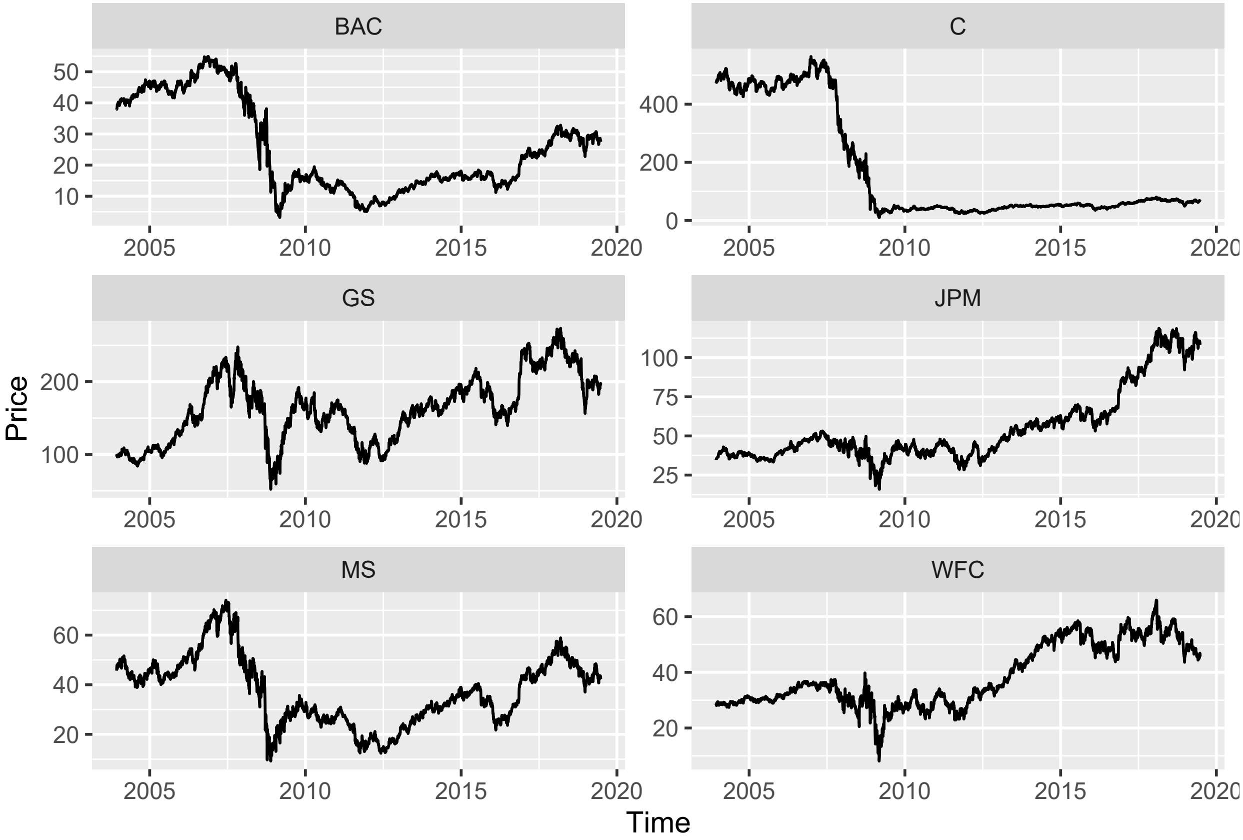


Figure 2: Time-series used in this study

3.2 Engle-Granger method

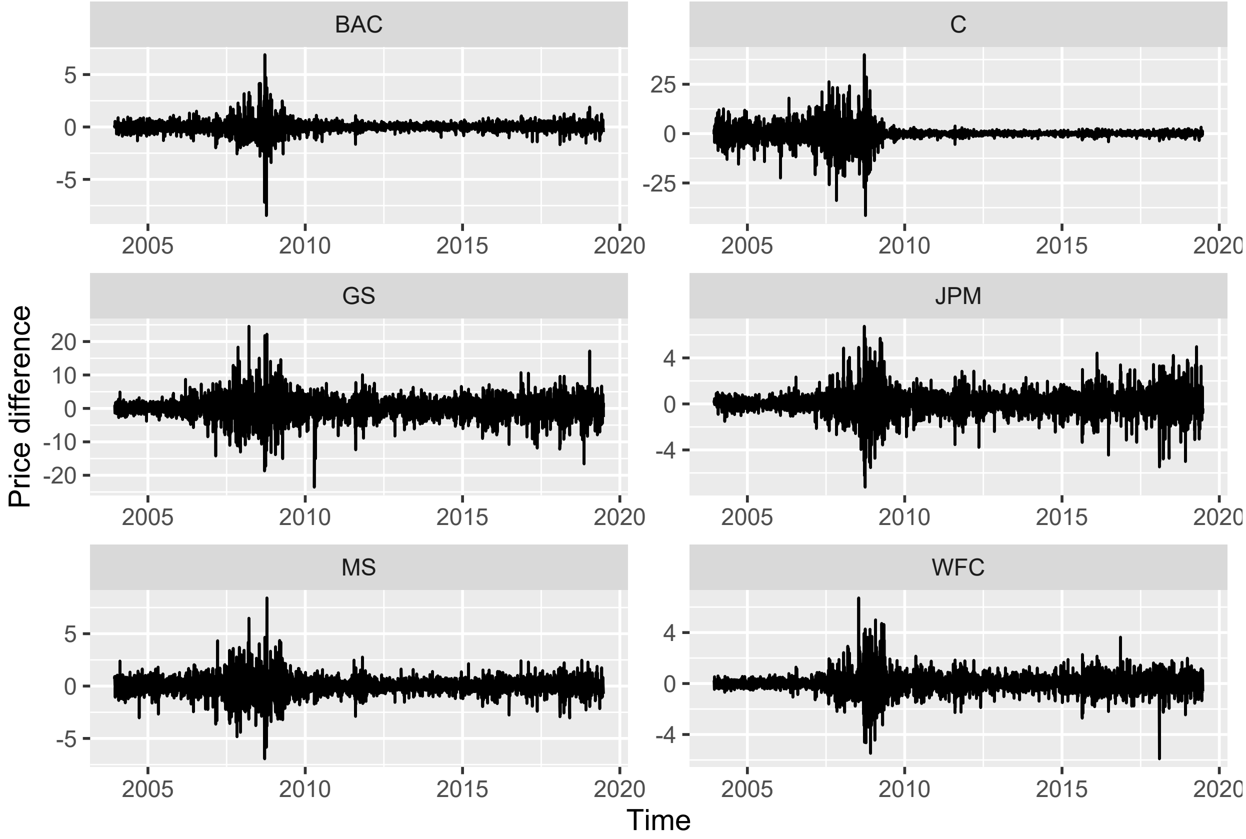


Figure 3: First difference of the time-series

The second step is to run OLS with all the possible pairs and check if there is a series of residuals stationer. Just as at the previous step the stationary test is augmented Dickey-Fuller test without constant or trend component in the auxiliary regression and $\alpha = 5\%$.

With the described parameters[10] the tests confirm only one cointegrated pair (see Figure 4), and that result holds only if the stock price of Bank of America is in regressor role, but it does not, when that is used as dependent variable[11].

3.3 Johansen-test

Johansen test is adequate cointegration test when there are more than two tested series at the same time. This test is performed to estimate the number of cointegrated vectors (r) in the system. If there is any cointegration in the model then $0 < r < k$, where k is the number of tested time-series. The system decomposition is not unique, so we can only estimate the cointegration rank r [Kirchgässner and Wolters, 2007]. The method can be performed with several tests, in this paper I chose the Lmax test. It gives a vector of the test statistics as a result and that may be compared to critical values. The null hypothesis is that $r \leq x$, where $x = 0, 1, 2, ..., k - 1$. The number of cointegrated vectors is the smallest x, under which the null hypothesis is

[10] In my previously mentioned GitHub repository, you may find that I wrote an R function to commit the whole Engle-Granger method with specified parameters. It would be reasonable to see the results with a different stationary test or with a different significance level (especially if calculating its profitability is also in focus). With the written function, it is possible to modify the test parameters and see how the results change.

[11] The matrix of the results is not a symmetrical.

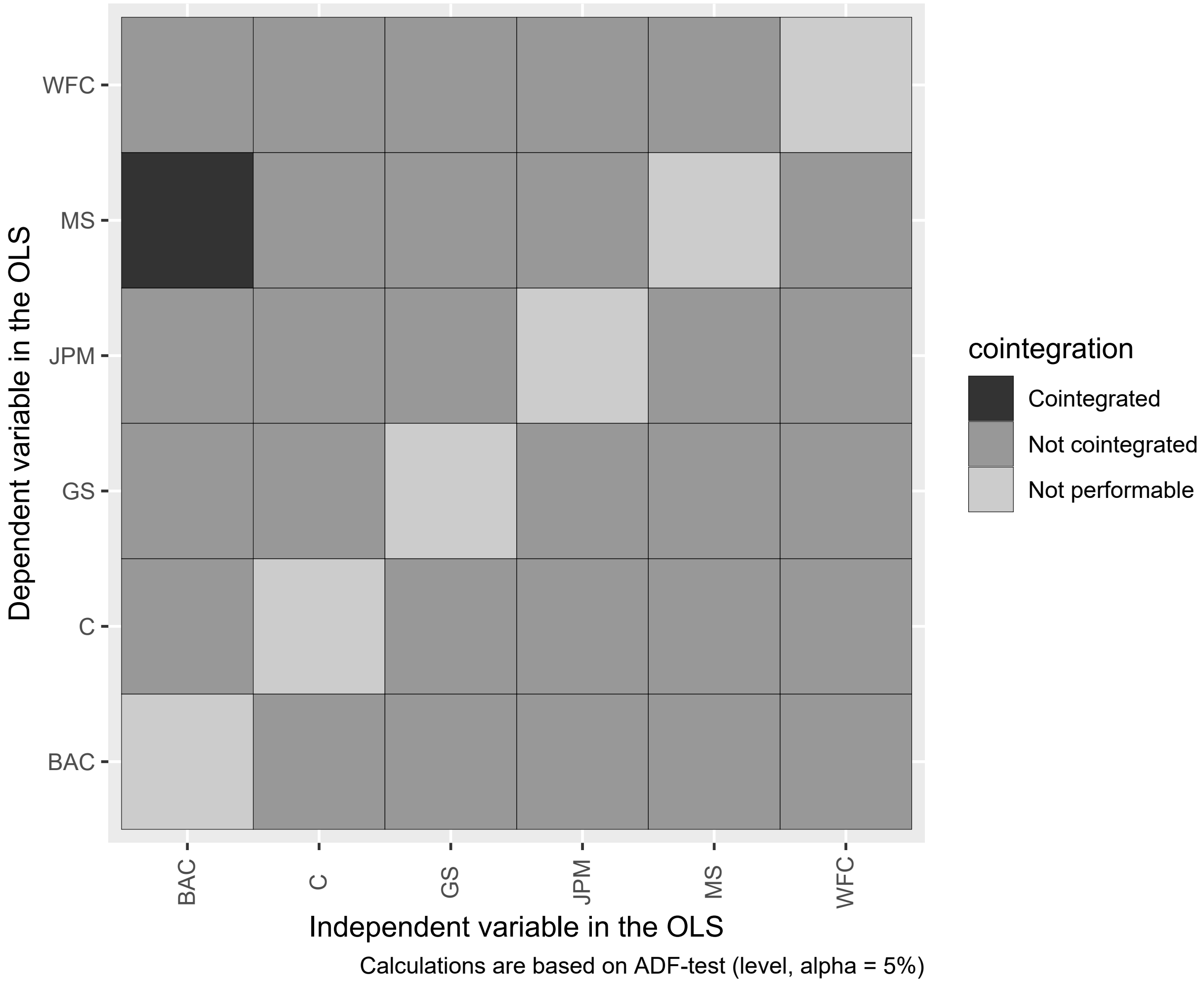


Figure 4: Results of Engle-Granger method

not rejected. The empirical analysis in this study shows that the r in this system is 1 on the full time-interval[12], which confirms the identical result like the one found with the Engel-Granger method.

3.4 Engle-Granger method with rolling window

In this section, I expound the results of the previously presented Engle-Granger method performed with a rolling window. The size of the windows is 250 days. Important to note, it is not sure that a stock price has the same integration order in each window. It can happen that a cointegration test is not performable, because in that period the integration orders do not match. Since this calculation is heavily time-consuming, only three of the six stock will be tested in this paper. This means that the maximum number of cointegrated pairings is 6 ($3 \times 3 - 3$). The test parameters are the same as described before, results are shown in figure 5.

In figure 5 it can be seen that the number of cointegrated pairings reaches the maximum number at the end of 2008, 2012 and in the middle of 2008, 2016. In 2008 there is also a long period when there are 4 cointegrated pairings. This result suggests a pattern that in recession cointegration may be more frequent.

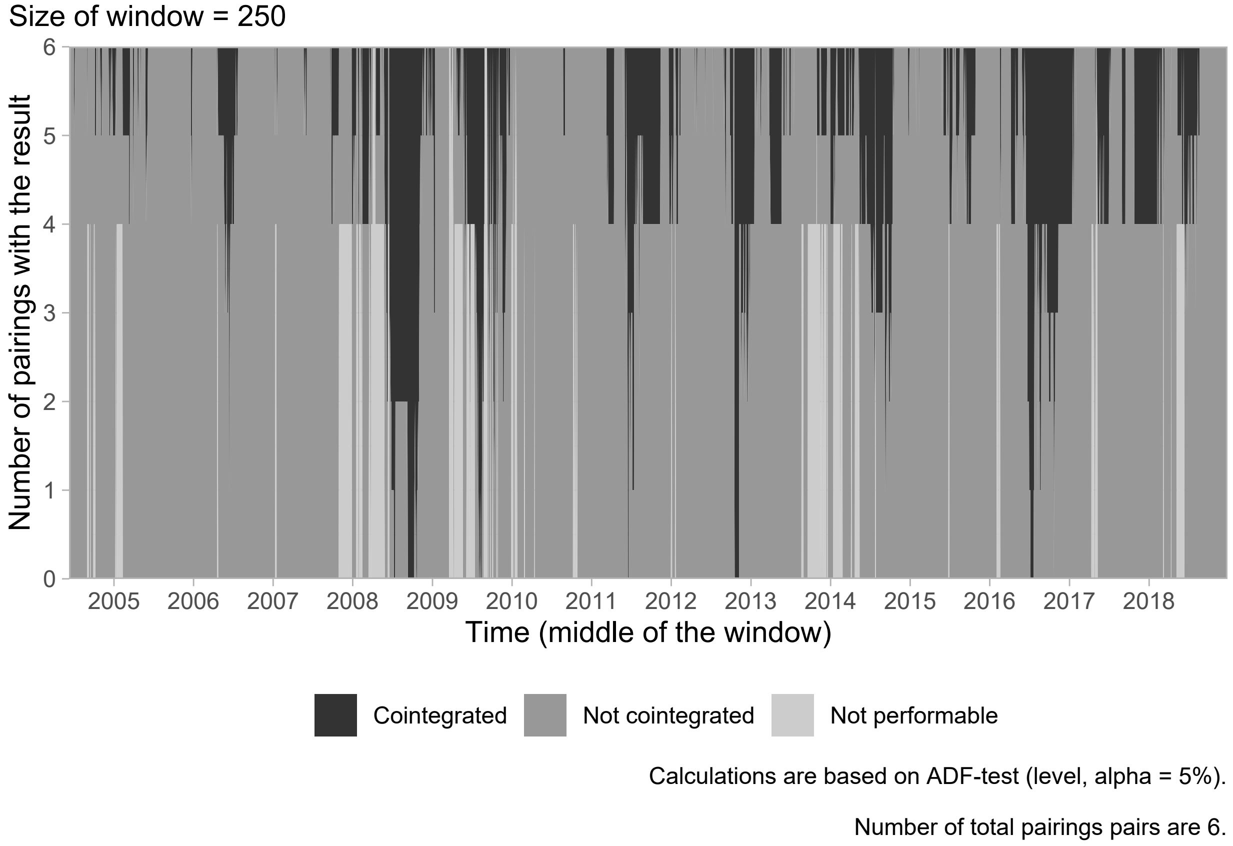


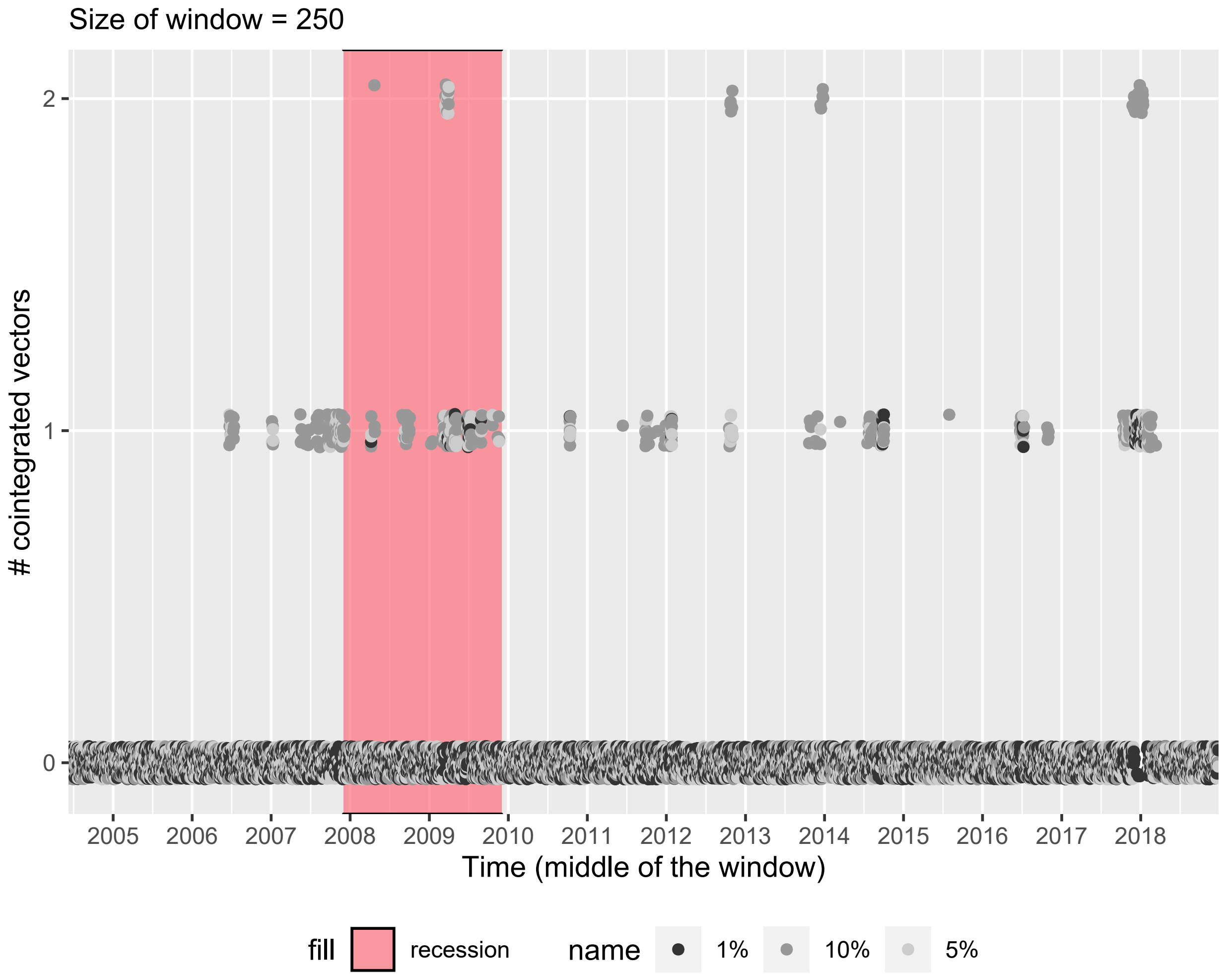
Figure 5: Results of Engle-Granger method with rolling window

3.5 Johansen test with rolling window

Performing the Johansen test with a rolling window is a similar extension as the one presented in the previous chapter. The calculations were performed with the same 250 window size and r is examined at the significance level of 1%, 5% and 10%. The result can be seen in figure 6.

In figure 6 the period of recession is also visualized. It looks like the $r = 1$ result at that time is more frequent

[12] Same result is stated on 1%, 5% and 10% significance level.



Points are jittered around their true y value for better visualisation (the number of cointegrated vectors is interger). Date of recession is from the National Bureau of Economic Research (https://www.nber.org/cycles.html).

Figure 6: Results of Johansen-test with rolling window across time

than most of the case when there is no recession, similarly the $r = 2$ result. One deviation from this pattern is at 2018, where $r = 1$ result is extremely frequent.

Looking at the distribution of the results controlling for the period of recession also confirms this hypothesis. During a recession, the proportion of $r = 2$ result (2.19%) is twice as much as the proportion when there is not recession (1.08%) with 10% significance level. Similarly $r = 1$ is the result of 15.31% of the total tests performed with $\alpha = 10\%$ in periods of recession, while 7.34% is when there is expansion. With different significance level, identical results can be concluded.

4 Multivariate simulation

In the following section I would like to introduce the concept of multivariate cointegration, and the case if the cointegration vector builds on more than one relation, but the test is performed only on two series. The simulation is written in R (just like the simulation in the previous section), and the visualisation is supported by a shiny application. The app can be accessed at the following link: https://marcell-grant.shinyapps.io/PairsTrading/. But this version is uploaded to a shiny server, where the usage time is billed by myself, so if you would like to walk through its features or use for longer time, then please run the application locally with the following R code:

```r
shiny::runGitHub("PairsTrading", "MarcellGranat", ref = "main")
```

4.1 Data Generating process

To simulate the case of omitted variable test, the first step is to choose an appropriate data generating process (DGP). To analyse different cases I defined 5 different DGPs: bivariate case with one or zero rank, and trivariate case with 2, 1 and zero rank. You can find their formulas in the appendix at line 404-444. You can see that 3 parameters can be modified in the DGPs: series length, standard deviation of the innovation (how big the noise is), and the AR parameter (p) of ARIMA part of the series which follows another one. Its function can be easily understand: if p goes to 0, the AR part will be white noise and the y will follow x in the long-run (at the case of bivariate simulation with 1 cointegrational vector). If p goes to 1, the AR will be a pure random walk, and the simulated time-series are not cointegrated.

4.2 Performance of Johansen-test

The first part of the simulations were the simple case to test how well Johansen procedure performs with different parameters of the DGP (how often it can identify the true data generating process). One thing must be sure: If the true DGP is a non-cointegrated bivariate case, then the rate of rejecting the $H_0 : r = 0$ must be 5%. The result was different with the widely used `urca` R package, which is a worrying conclusion, since `urca::ca.jo` function is frequently used in academic papers (like in the papers I read in this field). This technical note questions the results of many papers about Pairs Trading. For my research, I looked for another R package to use Johansen-test. An appropriate alternative is the `tsDyn`. In the application it can be checked that the mentioned result appears.

Underestimation of the cointegration rank is also possible (power of the test). It is frequently mentioned in the literature that cointegration tests require high number of observations (Clegg reports a minimum 250 to perform bivariate test). But if $p < .5$ this does not seem to be the case with Johansen-test. At the field of Simulation 1. in the application, one can whether requierd sample size to confidently manage the test.

4.3 Comparison of tests performed on 2 or 3 time-series

Following the idea of [Podivinsky, 1998] a trivariate DGP can be misspecified if one ommites a variable from the system. This may be a key issue to extend pairs trading strategy. It may be easily imagine that the price of a product is moving frequently together with the price of many different products, but the economic reason is also different. For example a price of x product takes similary shape as y product, because they are in the same industry, and peoples opinion about the future of that industry is a key component. But price of x also depends on the situation of its geographical area, and price of z also depends on that. Thus, observing y and z we could make a confident prediction about x. But if we only watch y and x we would see a random component, and the trivariate cointegration system cannot be identified. Moreover, the seen “random component” could dominate strong enough that we would reject the hyphothesis that even one relation exists in this system.

To analyse this issue I generated trivariate series with one cointegrational rank and performed the test on the three series and pairwisely as well. To compare the performance of the trivariate and pairwise testing I summed the found cointegrational rank of $x - y$ and $x - y$ tests. You can check the result in the application with different parameter setup (different series length and autocorrelation term).

With $length = 450$ and $p = .3$ the result clearly confirms the relevance of multivariate strategies. The pairwise testing found 2 cointegrational relation in less than 30% of the trajectories (this is the correct identification) and found only one relation at about the half of the simulations. In contrast, trivariate tests identified the DGP almost at each trajectory (5% should be the mistaken nominally). This leads to the conclusion that extending pairs trading strategy with multivariate tests is a relevant issue.

References

[Bayer and Hanck, 2013] Bayer, C. and Hanck, C. (2013). Combining non-cointegration tests. *Journal of Time Series Analysis*, 34(1):83–95.

[Caldeira and Moura, 2013] Caldeira, J. and Moura, G. V. (2013). Selection of a portfolio of pairs based on cointegration: A statistical arbitrage strategy.

[Clegg, 2014] Clegg, M. (2014). On the persistence of cointegration in pairs trading. *SSRN Electronic Journal*.

[Clegg and Krauss, 2018] Clegg, M. and Krauss, C. (2018). Pairs trading with partial cointegration. *Quantitative Finance*, 18(1):121–138.

[Do and Faff, 2012] Do, B. and Faff, R. (2012). Are pairs trading profits robust to trading costs? *Journal of Financial Research*, 35(2):261–287.

[Gatev et al., 2006] Gatev, E., Goetzmann, W. N., and Rouwenhorst, K. G. (2006). Pairs trading: Performance of a relative-value arbitrage rule. *The Review of Financial Studies*, 19(3):797–827.

[Gonzalo and Lee, 1998] Gonzalo, J. and Lee, T.-H. (1998). Pitfalls in testing for long run relationships. *Journal of Econometrics*, 86(1):129–154.

[Huck, 2015] Huck, N. (2015). Pairs trading: does volatility timing matter? *Applied Economics*, 47(57):6239–6256.

[Huck and Afawubo, 2014] Huck, N. and Afawubo, K. (2014). Pairs trading and selection methods: is cointegration superior? *Applied Economics*, 47(6):599–613.

[Jacobs and Weber, 2015] Jacobs, H. and Weber, M. (2015). On the determinants of pairs trading profitability. *Journal of Financial Markets*, 23:75–97.

[Kirchgässner and Wolters, 2007] Kirchgässner, G. and Wolters, J. (2007). *Introduction to modern time series analysis*. Springer, Berlin and New York.

[Krauss and Herrmann, 2017] Krauss, C. and Herrmann, K. (2017). On the power and size properties of cointegration tests in the light of high-frequency stylized facts. *Journal of Risk and Financial Management*, 10(1):7.

[Neely et al., 2014] Neely, C. J., Rapach, D. E., Tu, J., and Zhou, G. (2014). Forecasting the equity risk premium: The role of technical indicators. *Management Science*, 60(7):1772–1791.

[Podivinsky, 1998] Podivinsky, J. M. (1998). Testing misspecified cointegrating relationships. *Economics Letters*, 60(1):1–9.

[Rad et al., 2016] Rad, H., Low, R. K. Y., and Faff, R. (2016). The profitability of pairs trading strategies: distance, cointegration and copula methods. *Quantitative Finance*, 16(10):1541–1558.

5 Appendix: R codes

```r
# Setup ----------------------------------------------------------------------------------

library(tidyverse)
library(urca)

WD <- getwd() %>% # root directory
  gsub(pattern = "PairsTrading.*", replacement = "PairsTrading")

load(str_c(WD, "/data.RData")) # financial assets data

theme_set(theme_light() + theme(
  legend.title = element_blank(),
  plot.title.position = "plot",
  plot.tag.position = "topright",
  plot.caption.position = "plot"
))

# EDA ------------------------------------------------------------------------------------

Bankdata %>%
  pivot_longer(-1) %>%
  ggplot(aes(x = Date, y = value)) +
  geom_line() +
  facet_wrap(vars(name), nrow = 3, scales = "free") +
  labs(
    x = "Time", y = "Price"
  )

Bankdata %>% select(-1) %>% cor() %>% data.frame() %>% rownames_to_column() %>%
  pivot_longer(-1) %>% mutate(
  value = ifelse(rowname == name, NA, value)
) %>%
  ggplot(aes(rowname, name, fill = value)) + geom_tile(color = "black") +
      scale_fill_gradient2(
      low = "#00A3AB", high = "#FF5B6B", space = "Lab", na.value = "grey50",
      guide = "legend", midpoint = 0, aesthetics = "fill", limits = c(-1,1)
    ) + labs(
      x = "", y = "", title = "Correlation-matrix", tag = "Not included"
    ) + theme(
      panel.border = element_blank()
)

# Engle-Granger method -------------------------------------------------------------------

Bankdata %>%
  select(-1) %>%
  apply(2, function(x) {
    # number of differences required for stationarity to each series
    forecast::ndiffs(x, test = "adf", alpha = 0.05, type = "level")
  })

Bankdata %>%
  select(-1) %>%
  apply(2, function(x) {
    diff(x)
  }) %>%
  data.frame() %>%
  mutate(
    Date = tail(Bankdata$Date, -1)
  ) %>%
  pivot_longer(-Date) %>%
  ggplot(aes(x = Date, y = value)) +
  geom_line() +
  facet_wrap(vars(name), nrow = 3, scales = "free") +
  labs(
    x = "Time", y = "Price difference"
  )

cointegration_tests <- function(df, test, type, alpha) {
  # test cointegrity for all combination in a df
  ndiff_df <- df %>%
    select(-1) %>%
    apply(2, function(x) { # # of differences required for stationarity to each series
      forecast::ndiffs(x, test = test, alpha = alpha, type = type)
    })

  v <- df %>% select(-1) %>% # remove year ---> IT MUST BE IN THE INPUT DF !
    names(.)
  df2 <- expand.grid(v, v) %>%
    rename_all(funs(c("y", "x"))) %>%
    mutate(
      y = as.character(y),
      x = as.character(x),
      ndiff = ifelse(ndiff_df[y] == ndiff_df[x], ndiff_df[y], 0),
      ndiff = ifelse(y == x, 0, ndiff) # if series are the same, put 0
    )

  v <- vector()
  for (i in seq(nrow(df2))) {
    if (df2[i, 3] != 0) {
      if (lm(y ~ x, data = rename_all(data.frame(y = df[df2[i, 1]], x = df[df2[i, 2]]),
                                   funs(c("y", "x")))) %>%
        broom::augment() %>% .$.resid %>%
        forecast::ndiffs(test = test, alpha = alpha, type = type) == df2[i, 3] - 1) {
        v[i] <- 2 # 2 ---> series are cointegrated
      } else {
        v[i] <- 1 # 1 ---> not cointegrated, but test is commitable
      }
    } else {
      v[i] <- 0 # 0 ---> test is not performable [I(0) OR not the same I() order OR
      # series are the same]
    }
  }
  df2 %>%
    mutate(
      cointegration = v
    ) %>%
    select(y, x, cointegration)
}

cointegration_tests_results <- cointegration_tests(df = Bankdata, test = "adf",
                                                   type = "level", alpha = 0.05)

cointegration_tests_results %>%
  mutate(
    cointegration = case_when(
      cointegration == 0 ~ "Not performable",
      cointegration == 1 ~ "Not cointegrated",
      cointegration == 2 ~ "Cointegrated"
    ),
    cointegration = factor(cointegration, levels = c("Cointegrated", "Not cointegrated",
                                                     "Not performable"))
  ) %>%
  ggplot() +
  geom_tile(aes(x = x, y = y, fill = cointegration), color = "black") +
  scale_fill_grey() +
  theme(
    axis.text.x = element_text(angle = 90, vjust = 0.45),
  ) +
  labs(
    y = "Dependent variable in the OLS",
    x = "Independent variable in the OLS",
    caption = "Calculations are based on ADF-test (level, alpha = 5%)"
  ) + theme(
  panel.border = element_blank()
)

# Johansen-test -------------------------------------------------------------------------

Bankdata %>%
  select(-1) %>%
  ca.jo(type = "eigen", K = 5, ecdet = "none", spec = "longrun") %>%
  summary() # Number of cointegrated vectors = 1

## Engle-Granger method with rolling window ---------------------------------------------

for (i in 1:(nrow(Bankdata) - 249)) {
  if (i == 1) {
    cointegration_tests_rw <- mutate(
      cointegration_tests(df = Bankdata[i:(i + 249), 1:4], test = "adf", type = "level",
                          alpha = 0.05),
      t = i
    )
  } else {
    cointegration_tests_rw <- rbind(cointegration_tests_rw, mutate(
      cointegration_tests(df = Bankdata[i:(i + 249), 1:4], test = "adf", type = "level",
                          alpha = 0.05),
      t = i
    ))
  }
}

cointegration_tests_rw %>%
  filter(y != x) %>%
  ggplot(aes(x = t, y = cointegration)) +
  geom_point() +
  facet_grid(cols = vars(x), rows = vars(y)) +
  scale_y_continuous(breaks = c(0, 1, 2),
                     labels = c("Not performable", "Not cointegrated", "Cointegrated")) +
  labs(
    subtitle = "Size of window = 250",
    y = "Result of the test",
    x = "# window",
    caption = "Calculations are based on ADF-test (level, alpha = 5%)\n
    Dependent variables (in the OLS) are placed horizontal, independents are vertical."
  )

cointegration_tests_rw %>%
  filter(cointegration == 2) %>%
  mutate(cointegration = factor(cointegration)) %>%
  group_by(y, x) %>%
  tally() %>%
  arrange(x) %>%
  mutate(
    n = n / max(cointegration_tests_rw$t),
    n = scales::percent(n, accuracy = .01)
  ) %>%
  pivot_wider(id_cols = y, values_from = n, names_from = x, names_prefix = "x = ") %>%
  arrange(y)

merge(expand.grid(1:(nrow(Bankdata) - 249), c(0, 1, 2)) %>%
        rename_all(funs(c("t", "cointegration"))),
  cointegration_tests_rw %>% filter(y != x) %>%
    group_by(t, cointegration) %>%
    summarise(n = n()),
  all.x = T
) %>%
  mutate(
    n = ifelse(is.na(n), 0, n),
    cointegration = case_when(
      cointegration == 0 ~ "Not performable",
      cointegration == 1 ~ "Not cointegrated",
      cointegration == 2 ~ "Cointegrated"
    ),
    cointegration = factor(cointegration, levels = c("Cointegrated", "Not cointegrated",
                                                     "Not performable")),
    t = as.Date(Bankdata$Date)[t + 125]
  ) %>%
  ggplot() +
  geom_area(aes(x = t, y = n, fill = cointegration)) +
  scale_y_continuous(expand = c(0, 0)) +
  scale_x_date(expand = c(0, 0), date_breaks = "1 year", date_labels = "%Y") +
  theme(
    legend.position = "bottom"
  ) +
  labs(
    subtitle = "Size of window = 250",
    y = "Number of pairings with the result",
    x = "Time (middle of the window)",
    caption = "Calculations are based on ADF-test (level, alpha = 5%).\n
    Number of total pairings pairs are 6."
  ) +
  scale_fill_grey()

# Johansen test with rolling window -----------------------------------------------------

johansen_tests_rw <- data.frame(t = 1:(nrow(Bankdata) - 249)) %>% mutate(
  pct10 = NA, pct5 = NA, pct1 = NA
)

for (i in 1:(nrow(Bankdata) - 249)) {
  if (i == 1) {
    johansen_critical_values <- ca.jo(
      x = Bankdata[i:(i + 249), 2:4], type = "eigen",
      K = 5, ecdet = "none", spec = "longrun"
    )@cval
  }
  johansen_tests_rw[i, 2] <- which.max(rev(ca.jo(
    x = Bankdata[i:(i + 249), 2:4], type = "eigen",
    K = 5, ecdet = "none", spec = "longrun"
  )@teststat) < rev(johansen_critical_values[, 1])) - 1
  johansen_tests_rw[i, 3] <- which.max(rev(ca.jo(
    x = Bankdata[i:(i + 249), 2:4], type = "eigen",
    K = 5, ecdet = "none", spec = "longrun"
  )@teststat) < rev(johansen_critical_values[, 2])) - 1
  johansen_tests_rw[i, 4] <- which.max(rev(ca.jo(
    x = Bankdata[i:(i + 249), 2:4], type = "eigen",
    K = 5, ecdet = "none", spec = "longrun"
  )@teststat) < rev(johansen_critical_values[, 3])) - 1
}

ggplot() +
  geom_ribbon(aes(
    x = c(as.Date("2007-12-01"), as.Date("2009-12-01")),
    ymin = -Inf,
    ymax = Inf,
    fill = "recession"), color = "black", alpha = .6) +
  geom_jitter(data = johansen_tests_rw %>%
                pivot_longer(-1) %>%
                mutate(
                  name = case_when(
                    name == "pct1" ~ "1%",
                    name == "pct5" ~ "5%",
                    name == "pct10" ~ "10%"
```

```r
                  ),
                  t = as.Date(Bankdata$Date)[t + 125]
                ),
              aes(x = t, y = value, color = name),width = 0, height = 0.05) +
  scale_color_grey() +
  theme(
    legend.position = "bottom"
  ) +
  scale_y_continuous(breaks = c(0, 1, 2)) +
  scale_x_date(expand = c(0, 0), date_breaks = "1 year", date_labels = "%Y") +
  labs(
    subtitle = "Size of window = 250",
    y = "# cointegrated vectors",
    x = "Time (middle of the window)",
    caption = str_wrap(str_c(
      "Points are jittered around their true y value for better ",
       "visualisation (the number of cointegrated vectors is interger). ",
        "Date of recession is from the National Bureau of Economic Research ",
        "(https://www.nber.org/cycles.html)."), 50)
  ) +
  theme(
    panel.grid.minor.y = element_blank()
  ) +
  scale_fill_manual(values = c("recession" = "#FF5B6B"))

johansen_tests_rw %>%
  select(-1) %>%
  gather() %>%
  mutate(
    key = case_when(
      key == "pct1" ~ "1%",
      key == "pct5" ~ "5%",
      key == "pct10" ~ "10%"
    ),
    key = factor(key, levels = c("10%", "5%", "1%"))
  ) %>%
  group_by(key, value) %>%
  tally() %>%
  ggplot() +
  geom_bar(aes(x = key, y = n, fill = factor(value, levels = 2:0)), position = "fill",
           stat = "identity", color = "black") +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1), expand = c(0, 0),
                     breaks = seq(from = 0, to = 1, by = .1)) +
  scale_fill_grey() +
  labs(
    title = "Distribution of the Johansen-test results with rolling window",
    x = "Alpha",
    y = "Proportion",
    fill = "Number cointegrated vectors (r)",
    subtitle = "Size of window = 250"
  ) +
  theme(
    legend.title = element_text(),
    legend.position = "bottom"
  )

johansen_tests_rw %>%
  pivot_longer(-1) %>%
  mutate(
    name = factor(name, levels = c("pct1", "pct5", "pct10")),
    t = as.Date(Bankdata$Date)[t + 125],
    t = ifelse(t > as.Date("2007-12-01") & t < as.Date("2009-12-01"), "recession",
               "expansion")
  ) %>% filter(t == "expansion") %>% group_by(name) %>% count(value) %>% pivot_wider(
    id_cols = value, values_from = n, names_from = name
  )  %>% mutate(
    pct1 = scales::percent(pct1/sum(pct1, na.rm = T), accuracy = .01),
    pct5 = scales::percent(pct5/sum(pct5, na.rm = T), accuracy = .01),
    pct10 = scales::percent(pct10/sum(pct10, na.rm = T), accuracy = .01)
  ) %>% rename_all(funs(c("# cointegrated vectors", "1%", "5%", "10%")))

johansen_tests_rw %>%
  pivot_longer(-1) %>%
  mutate(
    name = factor(name, levels = c("pct1", "pct5", "pct10")),
    t = as.Date(Bankdata$Date)[t + 125],
    t = ifelse(t > as.Date("2007-12-01") & t < as.Date("2009-12-01"), "recession",
               "expansion")
  ) %>% filter(t == "recession") %>% group_by(name) %>% count(value) %>%
  pivot_wider(
    id_cols = value, values_from = n, names_from = name
  )  %>% mutate(
    pct1 = scales::percent(pct1/sum(pct1, na.rm = T), accuracy = .01),
    pct5 = scales::percent(pct5/sum(pct5, na.rm = T), accuracy = .01),
    pct10 = scales::percent(pct10/sum(pct10, na.rm = T), accuracy = .01)
  ) %>%
  rename_all(funs(c("# cointegrated vectors", "1%", "5%", "10%")))

library(tsDyn)
library(tseries)

# DGP functions -------------------------------------------------------------------------

simulate_bivariate_0ci <- function(t = 250, innov_sd = .5) {
  # Simulated bivariate non-cointegrated system
  tibble(
    x = cumsum(rnorm(t, 0, innov_sd)),
    y = cumsum(rnorm(t, 0, innov_sd))
  )
}

simulate_bivariate_1ci <- function(t = 250, p = .75, innov_sd = .5) {
  # Simulated bivariate cointegrated system
  y2 <- cumsum(rnorm(t, 0, innov_sd))
  y1 <- y2 + arima.sim(list(ar= p), innov = rnorm(t, 0, innov_sd), n = t)
  tibble(x = y1, y = y2)
```

```r
}

simulate_trivariate_0ci <- function(t = 250, innov_sd = .5) {
  # Simulated bivariate non-cointegrated system
  tibble(
    x = cumsum(rnorm(t, 0, innov_sd)),
    y = cumsum(rnorm(t, 0, innov_sd)),
    z = cumsum(rnorm(t, 0, innov_sd))
  )
}

simulate_trivariate_1ci <- function(t = 250, p = .75, innov_sd = .5) {
  # Simulated trivariate cointegrated system with 1 cointegrating vector
  y2 <- cumsum(rnorm(t, 0, innov_sd))
  y3 <- cumsum(rnorm(t, 0, innov_sd))
  y1 <- .5*y2 + .5*y3 + arima.sim(list(ar= p), innov = rnorm(t, 0, innov_sd), n = t)
  tibble(x = y1, y = y2, z = y3)
}

simulate_trivariate_2ci <- function(t = 250, p = .75, innov_sd = .5) {
  # Simulated trivariate cointegrated system with 2 cointegrating vectors
  y3 <- cumsum(rnorm(t, 0, innov_sd))
  y1 <- y3 + arima.sim(list(ar= p), innov = rnorm(t, 0, innov_sd), n = t)
  y2 <- y3 + arima.sim(list(ar= p), innov = rnorm(t, 0, innov_sd), n = t)
  tibble(x = y1, y = y2, z = y3)
}

# Setup ----------------------------------------------------------------------------

library(tidyverse)
library(tsDyn)
set.seed(2021)

# DGP functions ----------------------------------------------------------------------------

simulate_bivariate_0ci <- function(t = 250, innov_sd = .5) {
  # Simulated bivariate non-cointegrated system
  tibble(
    x = cumsum(rnorm(t, 0, innov_sd)),
    y = cumsum(rnorm(t, 0, innov_sd))
  )
}

simulate_bivariate_1ci <- function(t = 250, p = .75, innov_sd = .5) {
  # Simulated bivariate cointegrated system
  y2 <- cumsum(rnorm(t, 0, innov_sd))
  y1 <- y2 + arima.sim(list(ar= p), innov = rnorm(t, 0, innov_sd), n = t)
  tibble(x = y1, y = y2)
}

simulate_trivariate_0ci <- function(t = 250, innov_sd = .5) {
  # Simulated bivariate non-cointegrated system
  tibble(
    x = cumsum(rnorm(t, 0, innov_sd)),
    y = cumsum(rnorm(t, 0, innov_sd)),
    z = cumsum(rnorm(t, 0, innov_sd))
  )
}

simulate_trivariate_1ci <- function(t = 250, p = .75, innov_sd = .5) {
  # Simulated trivariate cointegrated system with 1 cointegrating vector
  y2 <- cumsum(rnorm(t, 0, innov_sd))
  y3 <- cumsum(rnorm(t, 0, innov_sd))
  y1 <- .5*y2 + .5*y3 + arima.sim(list(ar= p), innov = rnorm(t, 0, innov_sd), n = t)
  tibble(x = y1, y = y2, z = y3)
}

simulate_trivariate_2ci <- function(t = 250, p = .75, innov_sd = .5) {
  # Simulated trivariate cointegrated system with 2 cointegrating vectors
  y3 <- cumsum(rnorm(t, 0, innov_sd))
  y1 <- y3 + arima.sim(list(ar= p), innov = rnorm(t, 0, innov_sd), n = t)
  y2 <- y3 + arima.sim(list(ar= p), innov = rnorm(t, 0, innov_sd), n = t)
  tibble(x = y1, y = y2, z = y3)
}

# Multivariate simulation ---------------------------------------------------------------

df <- expand.grid(
  n = 2:3,
  ci = 0:2,
  control_1 = c("t", "p", "innov_sd"),
  control_2 = c("t", "p", "innov_sd"), stringsAsFactors = FALSE
) %>%
  filter(control_1 < control_2 &
           ci < n &
           (ci != 0 | (control_1 != "p" & control_2 != "p"))
  ) %>%
  arrange(n, ci) %>%
  tibble()

control_values <- list(
  t = 1:20 * 50,
  p = c(1:9 / 10, .92, .94, .96, .98),
  innov_sd = c(.2, .5, 1, 1.5, 2)
)

df <- df %>%
  mutate(
    fixed = map2_chr(control_1, control_2, ~ setdiff(c("t", "p", "innov_sd"),
                                                     c(.x, .y))),
    control_value_1 = map(control_1, ~ control_values[[.]]),
    control_value_2 = map(control_2, ~ control_values[[.]]),
  ) %>%
  unnest(control_value_1) %>%
  unnest(control_value_2)
```

```r
df <- df %>%
  mutate(
    f = str_c(ifelse(n == 2, "simulate_bivariate_", "simulate_trivariate_"), ci, "ci"),
    innov_sd = ifelse(control_1 == "innov_sd", control_value_1, .5),
    t = ifelse(control_2 == "t", control_value_2, 250),
    p = ifelse(control_1 == "p", control_value_1, .75),
    p = ifelse(control_2 == "p", control_value_2, p)
  ) %>%
  select(n, ci, innov_sd, t, p, f) %>%
  slice(rep(1:n(), each = 10000))

simulate_and_estimate_rank <- function(f, innov_sd, t, p) {
  if (f != "simulate_bivariate_0ci" & f != "simulate_trivariate_0ci") {
    invoke(f, list(innov_sd = innov_sd, t = t, p = p)) %>%
      tsDyn::VECM(lag = 0, estim = "ML", include = "none") %>%
      tsDyn::rank.test(type = 'trace', cval = 0.05) %>%
      .$r
  } else {
    invoke(f, list(innov_sd = innov_sd, t = t)) %>%
      tsDyn::VECM(lag = 0, estim = "ML", include = "none") %>%
      tsDyn::rank.test(type = 'trace', cval = 0.05) %>%
      .$r
  }
}

library(parallel)
library(progressr)
cl <- makeCluster(8)
clusterEvalQ(cl, library(tidyverse))
clusterExport(cl, list("simulate_bivariate_0ci", "simulate_bivariate_1ci",
                       "simulate_trivariate_0ci", "simulate_trivariate_1ci",
                       "simulate_trivariate_2ci", "df", "simulate_and_estimate_rank"),
              envir = environment())
evaluated_df <- tibble()

for (i in 1:1000) {

  current_df <- df %>%
    filter(cut(row_number(), 1000, F) == i)

  estimated_rank <- parApply(cl, current_df, 1, FUN = function(x) {
    simulate_and_estimate_rank(f = x[6], as.numeric(x[3]), as.numeric(x[4]),
                               as.numeric(x[5]))
  })

  current_df$estimated_rank <- estimated_rank
  evaluated_df <- bind_rows(evaluated_df, current_df)

  if (i %% 10 == 0) {
    cat(str_c(str_c(rep("=", i/1000*100 - 1), collapse = ""), ">",
              str_c(rep(" ", 100-i/1000*100), collapse = ""), "| ", i/1000*100, "%\n",
              collapse = ""))
  }
}

df_sum <- evaluated_df %>%
  count(n, ci, innov_sd, t, p, estimated_rank) %>%
  arrange(desc(estimated_rank)) %>%
  group_by(n, ci, innov_sd, t, p) %>%
  group_modify(~ mutate(.x, rate = nn / sum(nn), reject_rate = cumsum(rate) - rate)) %>%
  ungroup()


# Simulation with ommited variable

df <- crossing(
  t = 1:20 * 50,
  p = c(1:9 / 10, .92, .94, .96, .98),
  trajectory = 1:1000
)

evaluated_df <- tibble()

for (i in 1:100) {
  current_df <- df %>%
    filter(cut(row_number(), 100, F) == i) %>%
    mutate(
      dgp = pmap(.l = list(t = t, p = p), .f = simulate_trivariate_1ci),
      dgp_xy = map(dgp, ~ select(., x, y)),
      dgp_xz = map(dgp, ~ select(., x, z)),
      dgp_yz = map(dgp, ~ select(., y, z)),
    ) %>%
    pivot_longer(cols = dgp:dgp_yz, names_to = "contained", values_to = "dgp") %>%
    mutate(
      rank = map_dbl(dgp, ~ {tsDyn::VECM(., lag = 0, estim = "ML", include = "none") %>%
          tsDyn::rank.test(type = 'trace', cval = 0.05) %>%
          .$r}
      )
    ) %>%
    select(-dgp) %>%
    pivot_wider(names_from = contained, values_from = rank) %>%
    rename(rank_xyz = dgp) %>%
    rename_all(~ str_replace_all(., "dgp", "rank")) %>%
    mutate(rank_xy_xz = rank_xy + rank_xz)
  evaluated_df <- bind_rows(evaluated_df, current_df)
  print(i)
}

df_sum <- evaluated_df %>%
  count(t, p, rank_xyz, rank_xy_xz)

# Financial performance -----------------------------------------------------------------

source("functions.R")

load("data.Rdata")
```

```r

df <- Bankdata

v <- df %>%
  names() %>%
  .[-1]

evaluated_df <- tibble()

for (window_length in seq(from = 50, to = 1000, by = 50)) {

  current_df <- crossing(stock1 = v, stock2 = v, stock3 = v) %>%
    filter(stock1 < stock2, stock2 < stock3) %>%
    crossing(t = seq(nrow(df) %/% window_length)) %>%
    mutate(stocks = pmap(list(stock1, stock2, stock3, t), .f = function(x, y, z, t) {
      select(df, x, y, z) %>%
        head(-(nrow(df) %% window_length)) %>%
        filter(cut(row_number(), nrow(df) %/% window_length, FALSE) == t)
    })
    ) %>%
    mutate(
      rank_xyz = map_dbl(stocks, ~ {tsDyn::VECM(., lag = 0, estim = "ML",
                                                include = "none") %>%
          tsDyn::rank.test(type = 'trace', cval = 0.05) %>%
          .$r}
      ),
      rank_xy = map_dbl(stocks, ~ {
        .[, 1:2] %>%
          tsDyn::VECM(lag = 0, estim = "ML", include = "none") %>%
          tsDyn::rank.test(type = 'trace', cval = 0.05) %>%
          .$r}
      ),
      rank_xz = map_dbl(stocks, ~ {
        .[, c(1, 3)] %>%
          tsDyn::VECM(lag = 0, estim = "ML", include = "none") %>%
          tsDyn::rank.test(type = 'trace', cval = 0.05) %>%
          .$r}
      ),
      rank_yz = map_dbl(stocks, ~ {
        .[, 2:3] %>%
          tsDyn::VECM(lag = 0, estim = "ML", include = "none") %>%
          tsDyn::rank.test(type = 'trace', cval = 0.05) %>%
          .$r}
      ),
    ) %>%
    mutate(window_length = window_length) %>%
    select(window_length, everything(), -stocks)

  evaluated_df <- bind_rows(evaluated_df, current_df)
  print(window_length)
}


evaluated_date <- tibble()
```

```
for (window_length in seq(from = 50, to = 1000, by = 50)) {

current_date <- df %>%
  head(-(nrow(df) %% window_length)) %>%
  mutate(t = cut(row_number(), nrow(df) %/% window_length, FALSE)) %>%
  group_by(t) %>%
  group_map(~ {head(.x, nrow(.x) / 2) %>%
      pull(Date) %>%
      last()
  }
  ) %>%
  reduce(c) %>%
  enframe() %>%
  set_names("t", "date") %>%
  mutate(
    window_length = window_length,
    date = lubridate::ymd(date)
    )
evaluated_date <- bind_rows(evaluated_date, current_date)
}

evaluated_df <- evaluated_df %>%
  left_join(evaluated_date)

shiny::runGitHub("PairsTrading", "MarcellGranat", ref = "main")
```