Hedge Ratio Calculations

There are various ways to find mean-reverting spread, including distance, cointegration, copula, and ML approach. In the next step, when a trade signal is generated (Z-score above or below some threshold value), a researcher needs to understand how to trade the spread. In other words, how many units of X stock to buy and how many units of Y to sell.

Let's consider an example: a filter system detected a potentially profitable mean-reverting pair of **AMZN** and **AMD** stock. However, AMZN price is $3200, and AMD is only $78. If we trade 1 unit of AMZN vs 1 unit of AMD and both prices revert back, we will face a negative P&L.

Why?

A 1% change in AMZN results in $32 change in your position value, however 1% in AMD results in only $0.78 P&L change. This problem is solved by calculating the *hedge ratio*, which balance dollar value differences between the spread legs. A simple solution is to divid AMZN price by AMD price, and use this resulting value as a hedge ratio.

latest

In this case, for each AMZN stock, we trade 3200/78 = 41 units of AMD stock. This approach is called the **ratio method**. In ArbitrageLab, we have implemented several methods which are used in hedge ratio calculations.

Spread construction methodology

Note

In the ArbitrageLab package, all hedge ratio calculations methods normalize outputs such that a dependent variable asset has a hedge ratio of **1** and a spread is constructed using the following formula:

$$S = leg1 - (hedgeratio_2) * leg2 - (hedgeratio_3) * leg3 - .....$$

Note

All hedge ratio calculation methods assume that the first asset is a dependent variable unless a user specifies which asset should be used as dependent.

One can use the *construct_spread* function from the ArbitrageLab hedge ratio module to construct spread series from generated hedge ratios.

```
>>> import pandas as pd
>>> import numpy as np
>>> from arbitragelab.hedge_ratios import construct_spread
>>> url = "https://raw.githubusercontent.com/hudson-and-thames/example-
data/main/arbitrage_lab_data/sp100_prices.csv"
>>> data = pd.read_csv(url, index_col=0, parse_dates=[0])
>>> hedge_ratios = pd.Series({"A": 1, "AVB": 0.832406370860649})
>>> spread = construct_spread(data[["AVB", "A"]], hedge_ratios=hedge_ratios)
>>> inverted_spread = construct_spread(
...     data[["AVB", "A"]], hedge_ratios=hedge_ratios, dependent_variable="A"
... )
>>> inverted_spread
Date
2017-01-03 -100.529...
```

Ordinary Least Squares (OLS)

One way to find a hedge ratio is to fit a linear regression and use a slope coefficient. In our example, we fit the following regression:

latest

$$AMZN = \beta * AMD$$

In several studies, you may also find that intercept term is included in the linear regression:

$$AMZN = \alpha + \beta * AMD$$

One critic Armstrong (2001) points at the OLS hedge ratios sensitivity to the ordering of variables. It is a possibility that one of the relationships will be cointegrated, while the other will not. This is troublesome because we would expect that if the variables are truly cointegrated, the two equations will yield the same conclusion.

Implementation

Total Least Squares (TLS)

A better solution is proposed and implemented, based on Gregory et al. (2011) to use orthogonal regression – also referred to as Total Least Squares (TLS) – in which the residuals of both dependent and independent variables are taken into account. That way, we incorporate the volatility of both legs of the spread when estimating the relationship so that hedge ratios are consistent, and thus the cointegration estimates will be unaffected by the ordering of variables.

Implementation

Johansen Test Eigenvector

One of the big advantages of the Johansen cointegration test is the resulting eigenvector which serves as a hedge ratio to construct a spread. A researcher can either use the *JohansenPortfolio* class from the ArbitrageLab cointegration module to find all eigenvectors or use a function from hedge ratios module to get the first-order eigenvector which yields the most stationary portfolio.

Implementation

Box-Tiao Canonical Decomposition (BTCD)

First, Box and Tiao introduced a canonical transformation of an N-dimensional stationary autoregressive process. The components of the transformed process can then be ordered from least to most predictable, according to the research by Box and Tiao.

The estimation of this method goes as follows. For the $VAR(L)$ equation, which is called a forecasting equation, this method fits β and estimates $\hat{P}_t$ from the beta. With the estimated P_t, it undergoes a decomposition process and solves for optimal weight. In short, the obj
come up with the matrix of coefficients that deliver a vector of forecasts with the mo
predictive power over the next observation.

$$\sum_{l=1}^{L}\sum_{i=1}^{N}\beta_{i,l,n}P_{t-l,i} + \beta_{n,0}X_{t-1,n} + \varepsilon_{t,n}$$

Implementation

Minimum Half-Life

Half-life of mean reversion is one of the key parameters which describe how often the spread will return to its mean value. As a result, instead of using a linear approach (OLS, TLS) a researcher may want to minimize the spread's half-life of mean-reversion. We have implemented the algorithm which finds the hedge ratio by minimizing half-life of mean-reversion.

Implementation

Minimum ADF Test T-statistic Value

In the same fashion as the minimum half-life is calculated, one can find a hedge ratio that minimizes the t-statistic of the Augmented Dickey-Fuller Test (ADF).

Note

As Minimum Half-Life and Minimum ADF T-statistics algorithms rely on numerical optimization, sometimes output results can be unstable due to the fact the optimization algorithm did not converge. In order to control this issue, *get_minimum_hl_hedge_ratio* and *get_adf_optimal_hedge_ratio* return scipy optimization object, which contains logs and a status flag if the method managed to converge.

Implementation Examples

Code Example

```
>>> import pandas as pd
>>> import numpy as np
>>> from arbitragelab.hedge_ratios import (
...     get_ols_hedge_ratio,
...     get_tls_hedge_ratio,
...     get_johansen_hedge_ratio,
...     get_box_tiao_hedge_ratio,
...     get_minimum_hl_hedge_ratio,
...     get_adf_optimal_hedge_ratio,
... )
>>> # Fetch time series of asset prices
>>> url = "https://raw.githubusercontent.com/hudson-and-thames/example-
data/main/arbitrage_lab_data/gld_gdx_data.csv"
>>> data = pd.read_csv(url, index_col=0, parse_dates=[0])
>>> ols_hedge_ratio, _, _, _ = get_ols_hedge_ratio(
...     data, dependent_variable="GLD", add_constant=False
... )
>>> print(f"OLS hedge ratio for GLD/GDX spread is {ols_hedge_ratio}")
OLS hedge ratio for GLD/GDX spread is {'GLD': 1.0, 'GDX': 7.6...}
>>> tls_hedge_ratio, _, _, _ = get_tls_hedge_ratio(data, dependent_variable="GLD")
>>> print(f"TLS hedge ratio for GLD/GDX spread is {tls_hedge_ratio}")
TLS hedge ratio for GLD/GDX spread is {...}
>>> joh_hedge_ratio, _, _, _ = get_johansen_hedge_ratio(data, dependent_variable="GLD")
>>> print(
...     f"Johansen hedge ratio for GLD/GDX spread is {joh_hedge_ratio}"
... )
Johansen hedge ratio for GLD/GDX spread is {...}
>>> box_tiao_hedge_ratio, _, _, _ = get_box_tiao_hedge_ratio(data, dependent_variable="GLD")
>>> print(
...     f"Box-Tiao hedge ratio for GLD/GDX spread is {box_tiao_hedge_ratio}"
... )
Box-Tiao hedge ratio for GLD/GDX spread is {...}
>>> hl_hedge_ratio, _, _, _, opt_object = get_minimum_hl_hedge_ratio(
...     data, dependent_variable="GLD"
... )
>>> print(
...     f"Minimum HL hedge ratio for GLD/GDX spread is {hl_hedge_ratio}"
... )
Minimum HL hedge ratio for GLD/GDX spread is {...}
>>> print(opt_object.status)
0
>>> adf_hedge_ratio, _, _, _, opt_object = get_adf_optimal_hedge_ratio(
...     data, dependent_variable="GLD"
... )
>>> print(
...     f"Minimum ADF t-statistic hedge ratio for GLD/GDX spread is {adf_hedge_ratio}"
... )
Minimum ADF t-statistic hedge ratio for GLD/GDX spread is {...}
>>> print(opt_object.status)
0
```

Research Notebooks

The following research notebook can be used to better understand the hedge ratios described above.

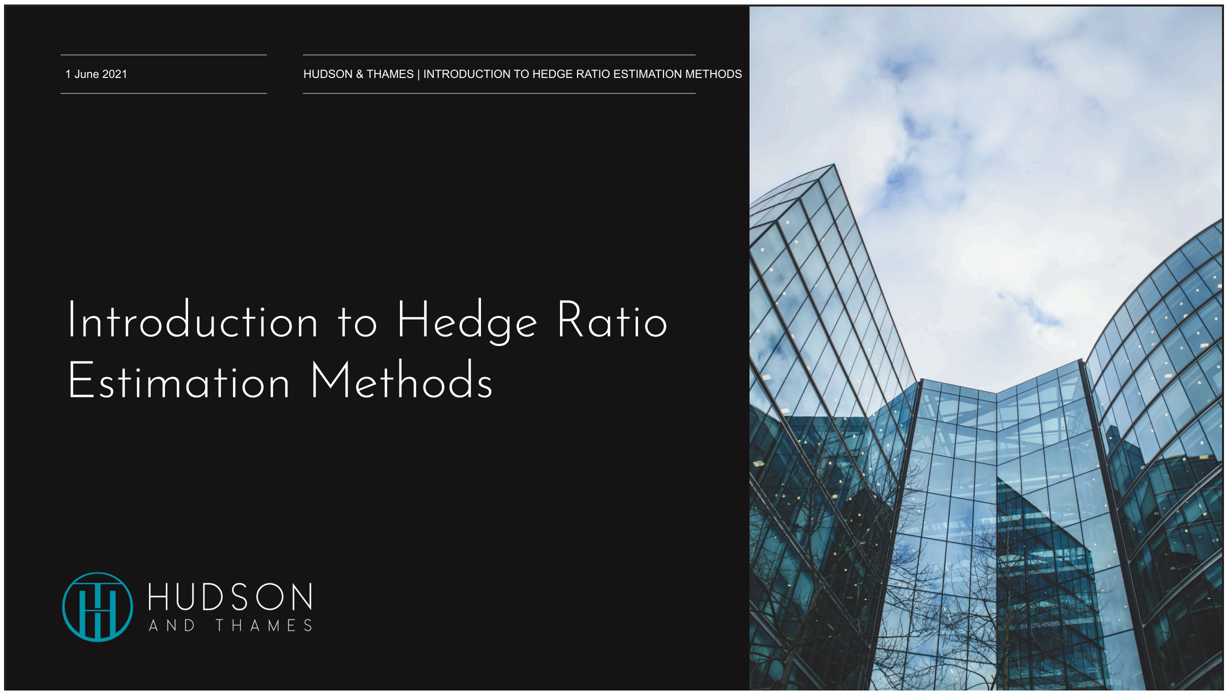
- Hedge Ratios

Download Notebook Download Sample Data

Research Article

Read our article on the topic

Presentation Slides


1 June 2021

HUDSON & THAMES | INTRODUCTION TO HEDGE RATIO ESTIMATION METHODS

Introduction to Hedge Ratio Estimation Methods

HUDSON AND THAMES

< 1 > ⋮ Google Slides

References

- Armstrong, J.S. ed., 2001. Principles of forecasting: a handbook for researchers and practitioners (Vol. 30). Springer Science & Business Media.

latest

- Gregory, I., Ewald, C.O. and Knox, P., 2010, November. Analytical pairs trading under different assumptions on the spread and ratio dynamics. In 23rd Australasian Finance and Banking Conference.