

# Recent Advances in Equity Analytics and the Equities Entity Store: Fractional Integration

This article serves as an extension to my book, *Equity Analytics*, and its accompanying analytical platform, the Equities Entity Store.

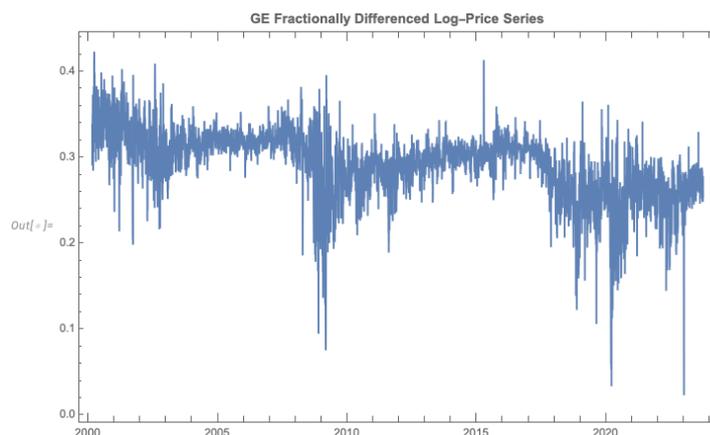
## Fractional Integration and Long Memory

A focal point of my ongoing research is fractional integration, a subject I delved into during my doctoral studies. This formed the basis for several new option pricing models and laid the foundation for my first volatility arbitrage hedge fund ([https://papers.ssrn.com/sol3/cf\\_dev/AbsByAuth.cfm?per\\_id=4178226](https://papers.ssrn.com/sol3/cf_dev/AbsByAuth.cfm?per_id=4178226)). The theory of fractional integration is now well understood, but by no means have all of its potential applications been explored. A couple of new possibilities are illustrated below and in more detail in the book.

The core concept is to model stationary processes, thereby maintaining their key properties consistently over time. One way to do this is to transform the price series into a returns process, which entails taking differences of the log-prices. While this does indeed render a process stationary, it also throws away potentially valuable information by eliminating serial autocorrelation. Fractional integration addresses this concern by taking fractional, rather than integer, differences in the price series. The details of the procedure are set out in the book, but the end result might look something like this:

```
In[ ]:= tsGE = QuantityMagnitude@FinancialData["GE", "Close", {2000, 1, 1}];  
tsLogGE = Log[tsGE];  
In[ ]:= optimalFracDSeries =  
AssociationThread[{"d", "tsFr", "AutoCorrelation"} -> FractionalDOSeries[TimeSeriesWindow[tsLogGE, {tsGE["FirstDate"], {2022, 12, 30}}], 0, 1, 0.001]]  
Out[ ]:= {d -> 0.591309, tsFr -> TimeSeries[ Time: 28 Jul 2000 GMT+1 to 30 Dec 2022 GMT+1  
Data points: 5644 ], AutoCorrelation -> 0.629652 }
```

Our estimation of the optimal difference parameter  $d$  is 0.5913, resulting in a fractionally integrated series with a lag-1 autocorrelation of 0.6297.



## Unit Root Testing, ARMA Modelling

As indicated by the chart, the resulting fractionally differenced log-price series does indeed appear to be stationary. We can test this hypothesis explicitly using the Dickey-Fuller unit root test, which rejects the null hypothesis of a unit root at the 0.1% level:

```
In[ ]:= UnitRootTest[tsGEfr, Automatic, {"AutomaticTest", "PValue"}]
Out[ ]:= {DickeyFullerF, 0.00108497}
```

Consequently, this allows us to apply an ARMA model to the fractionally differenced series, as outlined below.

```
In[ ]:= tsm = TimeSeriesModelFit[tsGEfr]
Out[ ]:= TimeSeriesModel [  Family: ARMA
                          Order: {3, 6} ]

In[ ]:= tsm // Normal
Out[ ]:= ARMAProcess[0.0230163, {0.212048, 0.25984, 0.449318}, {0.228228, 0.00865984, -0.299274, -0.0612032, -0.0566723, -0.0347314}, 0.00046355]

In[ ]:= WeakStationarity[tsm // Normal]
Out[ ]:= True
```

## Trading Strategy

So, we have a fractionally differenced log-price series that is stationary and which we can model as an ARMA process. This suggests a way to exploit the characteristics of the transformed process, by sequentially fitting a new ARMA model at each time step  $t$ , then generating a trading signal if price at time  $t+1$ ,  $P_{t+1}$ , falls outside the confidence interval for the model forecast.

The Equity Analytics book provides the code for a recursive trading strategy based on fitting an ARMA model to the fractionally differenced series, generating 1-step-ahead forecasts and confidence intervals for each time frame. If the value of the differenced series falls outside the confidence interval, we take a long or short position in the stock depending on whether it falls below the lower band, or exceeds the upper band, of the confidence interval. A trade P&L is calculated using the 1-period change in price level in the Close price series (not the differenced series!). The trading outcomes for the period spanning January to September 2023 are as follows:

```
In[ ]:= ListLinePlot[Accumulate[tsPL], Filling -> Axis]
```

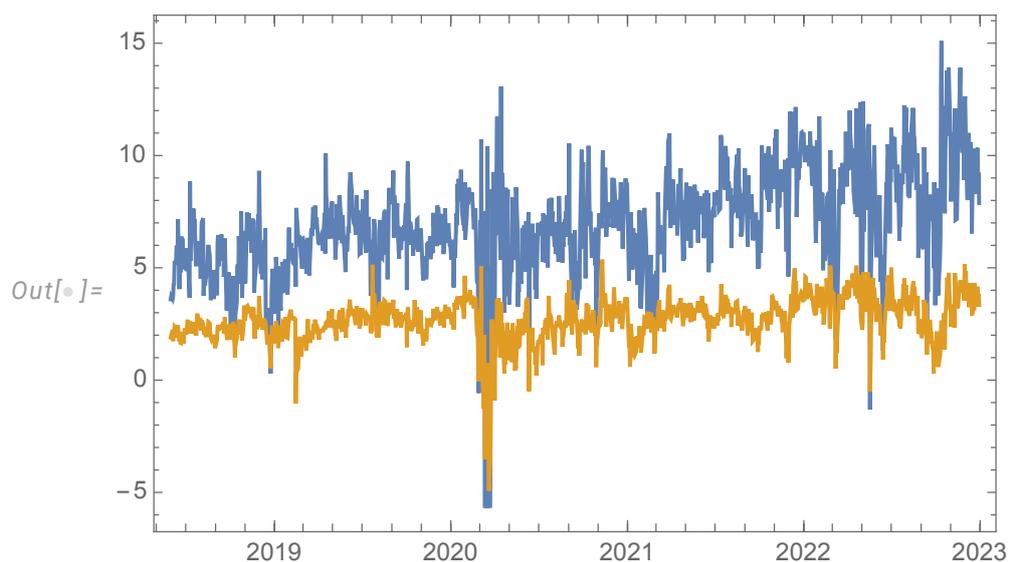


## Fractional Integration in Pairs Trading

Another application where fractional integration has a useful role to play is in statistical arbitrage.

Here we take the case of the PEP-KO pairs trade, where we first fractionally difference each price series, rather than using the returns series as our starting point.

```
In[ ]:= DateListPlot[tsFractionalPriceSeries /@ symbols]
```

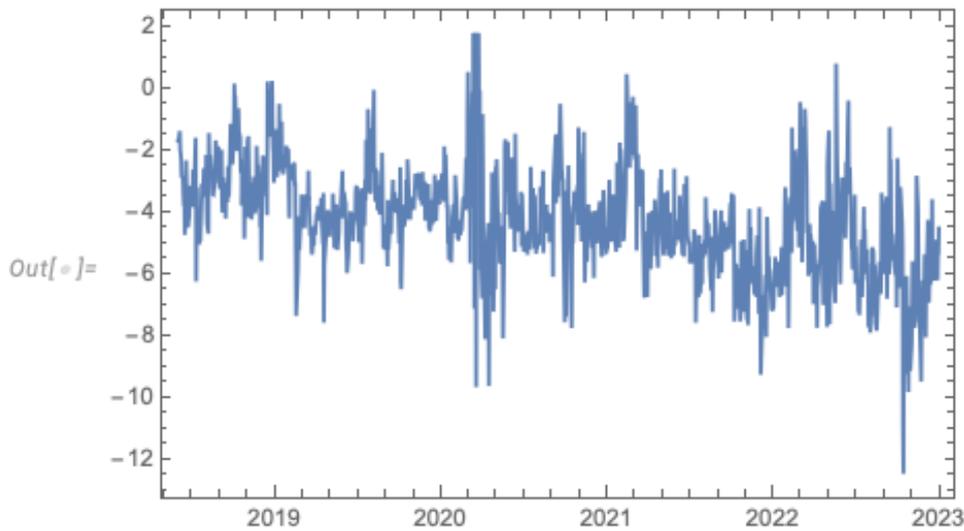


Testing the difference between the two fractionally differences series we find that it too is stationary:

```
In[ ]:= tsFractionalDifferences = First[Differences[tsFractionalPriceSeries[#] & /@ symbols]]
```

```
Out[ ]:= TimeSeries [  Time: 31 May 2018 GMT+1 to 30 Dec 2022 GMT+1  
Data points: 1156 ]
```

```
In[ ]:= DateListPlot[tsFractionalDifferences]
```



The null hypothesis of a unit root in the difference series is rejected at around the 1% level (or less) in all four variants of the tests:

```
In[ ]:= UnitRootTest[tsFractionalDifferences["Values"], Automatic, All]
```

```
Out[ ]:= {7.38223 × 10-8, 3.83325 × 10-8, 0.00137174, 0.000717983}
```

We don't have to generate trade signals using the Zscores from the Kalman Filter model, which is our go-to methodology. Instead, noting that the fractionally differenced series is itself a stationary process, we can simply fit an ARMA model to the series and generate signals based on the standardized residuals, as follows:

```
In[ ]:= tsSignals = TimeSeries[Sign[Values@stdResiduals] * UnitStep[Abs[Values[stdResiduals]] - 1.5], stdResiduals["DateList"]]
```

```
Out[ ]:= TimeSeries[
```

Now we generate trading signals based on the value of the standardized residuals, using a threshold value of +/- 1.5, with results as follows:

Start Date	Fri 29 Dec 2017	Realized Gross Profit	334575.
End Date	Fri 30 Dec 2022	Realized Gross Loss	-192294.
Investment Period	5. yr	Realized Net Profit	\$142280.84
Total Realized Profit	\$142167.33	Profit Factor	1.74
Total Unrealized Profit	\$20100.84	No. Trades	58
Total Net Profit	\$162268.17	Avg. Trade	\$2453.12
Transaction Costs	\$11872.65	Avg. Trade Duration	28.6 days
Total Return	162.27 %	No. Winners	36
CAGR	21.27 %	No. Losers	22
Annual Volatility	32.16 %	Win Rate	62.07 %
Annual Semi-Deviation	20.63 %	Av. Winning Trade	\$9293.75
Sharpe Ratio	0.66	Av. Losing Trade	\$-8740.65
Sortino Ratio	1.03	Win/Loss Ratio	1.06
Calmar Ratio	0.56	Slippage	\$7915.10
Max Drawdown Date	Wed 24 Mar 2021	Commission	\$3957.55
Max Drawdown	\$-62321.88		
Max Drawdown Pct	-38.18 %		



Among the methods tested for the PEP-KO pair from 2017 to 2022, the fractional differencing method showed the highest risk-adjusted returns, when signals were generated from the standardized residuals of an ARMA model fitted to the fractionally-differenced series.

For more details and additional insights into quantitative investing, refer to 'Equity Analytics and the Equities Entity Store: <https://equityanalytics.store/>, or contact me at [jkinlay@gmail.com](mailto:jkinlay@gmail.com).