

## 2.2 Trading Anomalies

*Anomaly detection, also known as outlier detection, is a method used to identify data points that are significantly different from the majority of the data. These data points are referred to as anomalies or outliers. Anomaly detection techniques are used in various fields such as fraud detection, network intrusion detection, and manufacturing quality control.*

*There are several techniques used for anomaly detection, including :*

**Statistical methods** : These methods rely on summary statistics of the data to detect anomalies, such as mean, median, and standard deviation. For example, Z - score method considers data points with a Z - score greater than a certain threshold to be anomalies.

**Clustering algorithms** : Clustering algorithms group similar data points together and anomalies are detected as data points that are far away from their cluster center. For example, K - Means clustering is a popular clustering algorithm used for anomaly detection.

**Density - based methods** : These methods use a density function to estimate the probability of a data point and identify data points with a low probability as anomalies. For example, the Gaussian Mixture Model can be used to estimate the probability density of the data.

**Machine learning - based methods** : These methods use machine learning algorithms, such as decision trees and random forests, to model the normal behavior of the data and identify data points that deviate from this behavior as anomalies.

*The choice of method for anomaly detection depends on the specific requirements of the task, such as the size and structure of the data, and the desired level of accuracy and computational efficiency.*

---

## Introduction

One very interesting application of machine learning algorithms is in anomaly detection, in which the aim is to identify unusual observations in a set of data. These methods are now widely used for real-time quality control in manufacturing processes, for example, to ensure that the dimensions of products on the production line are within stipulated tolerances.

Similar techniques can be applied to time series, including financial time series. In the Equities Entity Store we have implemented algorithms that will identify unusual patterns in the sequences of equity returns and convert these into trading signals. So far as I am aware, there has been no published research on this or any similar methodology and it has been made available for the first time in the initial release of the Equities Entity Store product.

---

## Trading Anomalies in Financial Time Series

Let's illustrate the application of the model using a sample of the daily returns for AAPL, over the period from 2020-2022.

In the plot below there are several sequences in the returns process that are detected as anomalous, which tend to be unusually large positive or negative returns during periods of higher volatility:

```
In[ ]:= Entity["Equities", "AAPL"]
```

```
Out[ ]:=
```

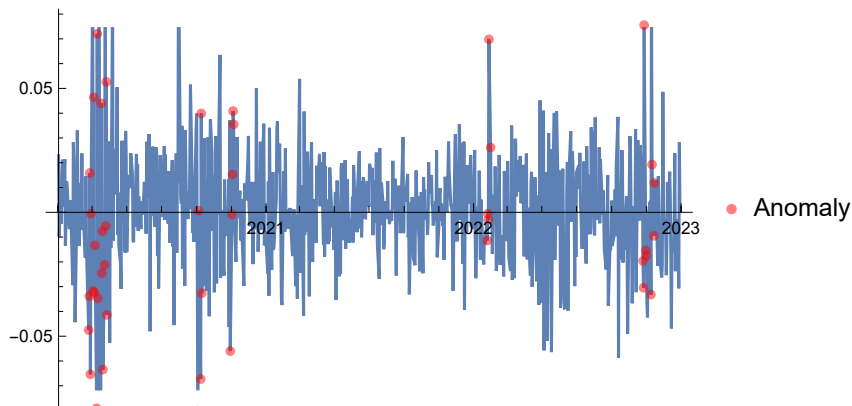
AAPL

```
tsStockReturns = EntityDailyReturns[ AAPL EQUITIES ];
```

```
tsSampleReturns = TimeSeriesWindow[tsStockReturns, {{2020, 1, 2}, {2022, 12, 30}}];
```

```
In[ ]:= DisplayAnomalies[tsSampleReturns]
```

```
Out[ ]:=
```

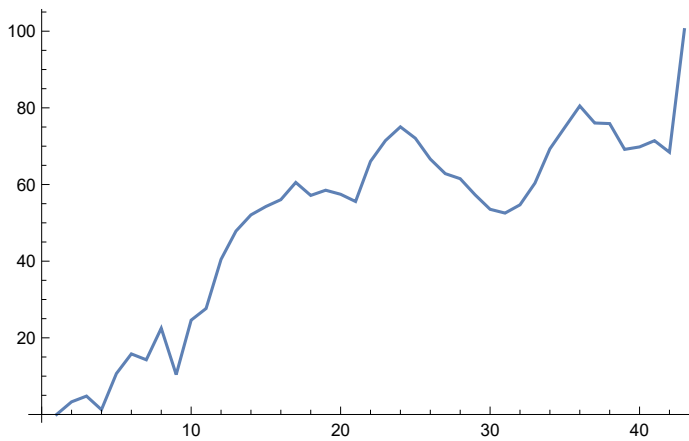


The anomaly trading algorithm uses machine learning to learn patterns of anomalies in the time series. It then identifies these anomalies out of sample and generates long or short trading signals. This process is captured in the “AnomalyTradingStrategy” function, which uses parallel processing to speed up the computationally intensive fitting and application of anomaly detection algorithms.

The process is illustrated below for AAPL, with out-of-sample results for the period from Jan 2020 to Dec 2022. The resulting strategy has a 60% trade win rate, total return of just over 100%, CAGR of 28.4% and Sharpe ratio of 1.46:

```
In[ ]:= AnomalyStrategyAAPL = AnomalyTradingStrategy["AAPL"];
```

```
In[ ]:= ListLinePlot[
  FoldList[Times, 100, 1 + AnomalyStrategyAAPL["AAPL", "Strategy Returns"] ["Values"]] - 100]
Out[ ]:=
```



```
In[ ]:= AnomalyStrategyAAPL["AAPL", "Strategy Performance"]
Out[ ]:=
{ | Start Date → Tue 4 Feb 2020 , End Date → Mon 14 Nov 2022 ,
  Interval → 2.77534 yr , No. Trades → 42, Total Return → 1.004, CAGR → 0.28463,
  Annual Volatility → 0.194191, Annual SemiDeviation → 0.0853749,
  Max Drawdown → -0.128516, Max Drawdown Date → Thu 17 Mar 2022 , Sharpe ratio → 1.46572,
  Sortino ratio → 3.33389, Calmar ratio → 2.21474, Avg. Return → 0.0239047,
  Win/Loss ratio → 1.9452, Profit Factor → 2.86059, Win Rate → 0.595238 | }
```

## Portfolio Strategy

The anomaly trading strategy is intriguing, but it becomes even more so when applied to a small portfolio of stocks. In this example, a group of seven stocks from various industries, including tech stocks and stocks from traditional industries such as retail, construction, and oil and gas, have been selected.

```
In[ ]:= stocks
Out[ ]:=
{AAPL, AZO, FLR, GOOG, LLY, V, XOM}

In[ ]:= <|# → #["Sector Information"] ["GICS Industry"] & /@ (Entity["Equities", #] & /@ stocks) |>
Out[ ]:=
{ | AAPL → Technology Hardware, Storage & Peripherals, AZO → Specialty Retail,
  FLR → Construction & Engineering, GOOG → Interactive Media & Services,
  LLY → Pharmaceuticals, V → IT Services, XOM → Oil, Gas & Consumable Fuels | }
```

We use the same `AnomalyTradingStrategy` as above, applied to each stock in turn, and collect the results in an association named `AnomalyTradingStrategyResults`:

```
AnomalyTradingStrategyResults = <| |>;
```

```
Do[AppendTo[AnomalyTradingStrategyResults,
  AnomalyTradingStrategy[stock]], {stock, stocks}];
```

There is considerable variation in the performance of the individual strategies, with AAPL being the best performing strategy in several categories. But the portfolio also includes stocks like FLR and LLY, that produce what appear to be relatively poor results:

```
In[ ]:= AnomalyTradingStrategyResults[[All, "Strategy Performance"]] // Dataset // Transpose
Out[ ]:=
```

	AAPL	AZO	FLR
Start Date	Tue 4 Feb 2020	Fri 6 Mar 2020	Thu 16 Jan 2020
End Date	Mon 14 Nov 2022	Wed 25 May 2022	Tue 18 Oct 2022
Interval	2.77534 yr	2.21918 yr	2.75342 yr
No. Trades	42	38	34
Total Return	0.983619	0.475996	0.00916385
CAGR	0.279909	0.191771	0.0033185
Annual Volatility	0.197455	0.205558	0.260267
Annual SemiDeviation	0.0901118	0.0999082	0.213127
Max Drawdown	-0.12138	-0.129693	-0.378721
Max Drawdown Date	Thu 17 Mar 2022	Thu 12 Mar 2020	Mon 30 Mar 2020
Sharpe ratio	1.41758	0.93293	0.0127504
Sortino ratio	3.10624	1.91947	0.0155706
Calmar ratio	2.30605	1.47866	0.0087624
Avg. Return	0.0234195	0.0125262	0.000269525
Win/Loss ratio	1.69039	1.4714	0.882123
Profit Factor	2.74688	2.02317	1.11736
Win Rate	0.619048	0.578947	0.558824



Next, we aggregate all of the individual strategies into an equally-weighted portfolio and review the results.

The combined portfolio performed exceptionally well, with a total return of 190% from 2020-2022, a yearly growth rate of 45.6%, and a Sharpe ratio of 3.46. It had moderate volatility and drawdown,

resulting in an outstanding Calmar ratio of 4.67.

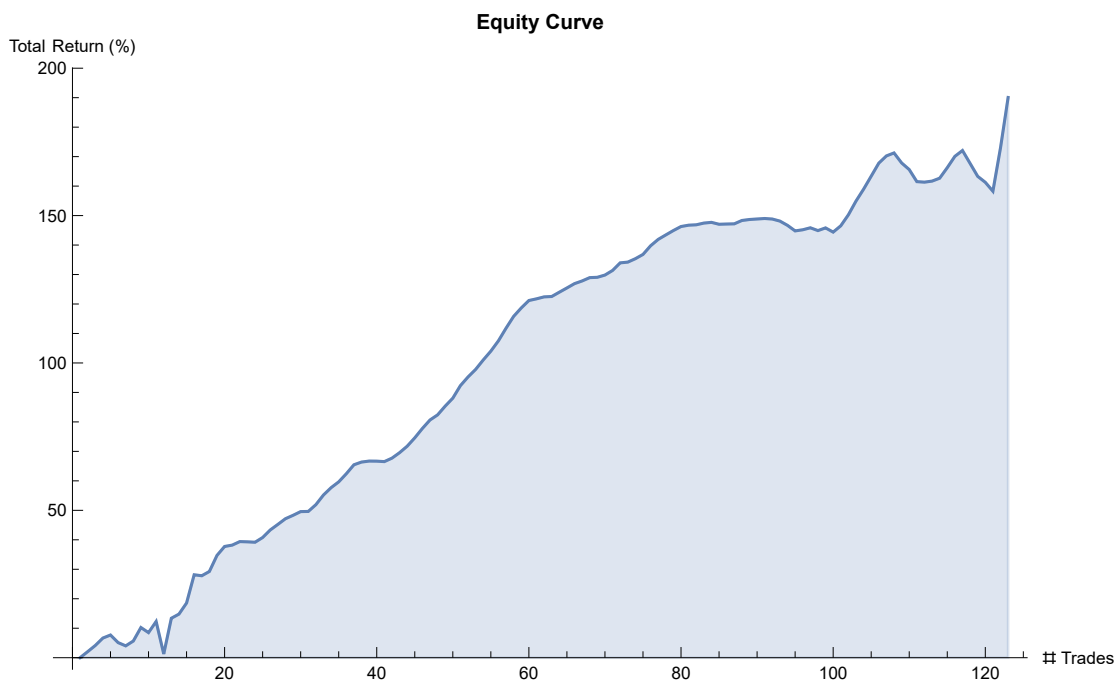
```
In[ ]:= {tsStrategyReturns, td} =  
PortfolioReturns[AnomalyTradingStrategyResults[All, "Strategy Returns"]]
```

Out[ ]:=

```
{TimeSeries[ Time: 16 Jan 2020 to 15 Nov 2022  
Data points: 122 ],  
TemporalData[ Time: 16 Jan 2020 to 15 Nov 2022  
Data points: 854 Paths: 7 ]}
```

```
In[ ]:= ListLinePlot[FoldList[Times, 100, 1 + tsStrategyReturns["Values"]] - 100,  
Filling -> Axis, AxesLabel -> {"# Trades", "Total Return (%)"},  
PlotLabel -> Style["Equity Curve", Bold], ImageSize -> Large]
```

Out[ ]:=



```
In[*]:= PerformanceStats[tsStrategyReturns] // Dataset
Out[*]=
```

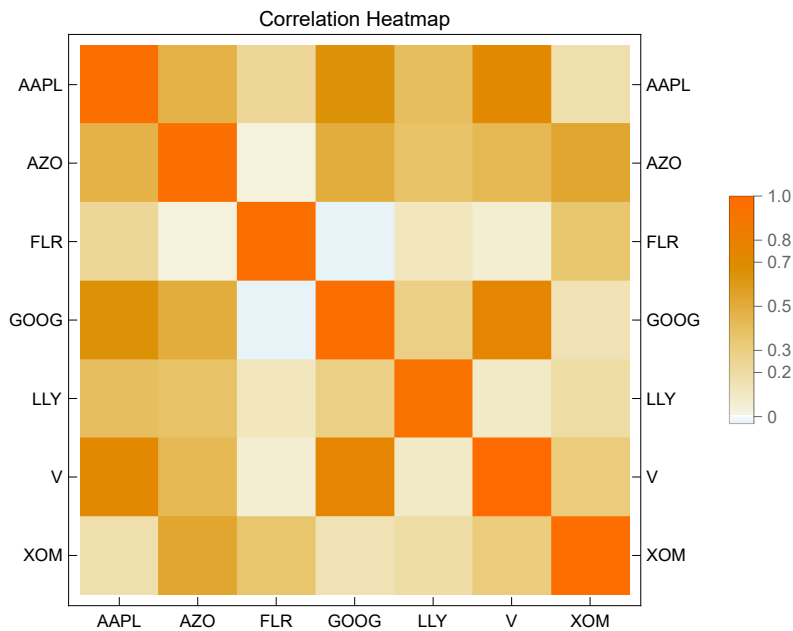
Start Date	Thu 16 Jan 2020
End Date	Tue 15 Nov 2022
Interval	2.83014 yr
No. Trades	122
Total Return	1.9006
CAGR	0.456852
Annual Volatility	0.131946
Annual SemiDeviation	0.128651
Max Drawdown	-0.0980989
Max Drawdown Date	Thu 12 Mar 2020
Sharpe ratio	3.46243
Sortino ratio	3.5511
Calmar ratio	4.65706
Avg. Return	0.0155787
Win/Loss ratio	1.23162
Profit Factor	5.02912
Win Rate	0.803279

---

## Impact of Diversification

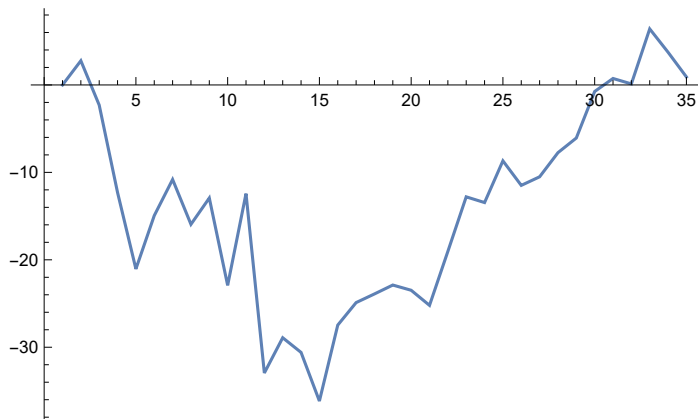
The secret to the portfolio's success is reducing risk through diversification. Although stock returns may be highly correlated, what matters here is the correlation between anomaly strategies. The heat map shows that many strategies have low correlations, usually lower than the correlations between the stock returns. This is because anomalies in the returns of different stocks occur at different times.

```
In[ ]:= CorrelationHeatmap[td, stocks]
Out[ ]:=
```



To make the point clear, let's look at the worst performing strategy in FLR:

```
In[ ]:= ListLinePlot[FoldList[Times, 100,
    1 + AnomalyTradingStrategyResults[["FLR", "Strategy Returns"]]["Values"]] - 100]
Out[ ]:=
```



The poor performance of the FLR strategy doesn't mean it should be excluded from the portfolio, however. It turns out that if we don't include FLR, portfolio performance deteriorates, producing lower total return, CAGR, and Sharpe ratio. This is because the FLR strategy had a strong performance in the second half of the test, compensating for other strategies that performed relatively poorly in that period.

We can quantify the impact of the FLR strategy by eliminating it from the portfolio:

```
In[ ]:= reducedStocklist = {"AAPL", "AZO", "GOOG", "LLY", "V", "XOM"};
```

```
In[ ]:= {tsStrategyReturnsNoFLR, tdNoFLR} =  
PortfolioReturns[AnomalyTradingStrategyResults[reducedStocklist, "Strategy Returns"]];
```

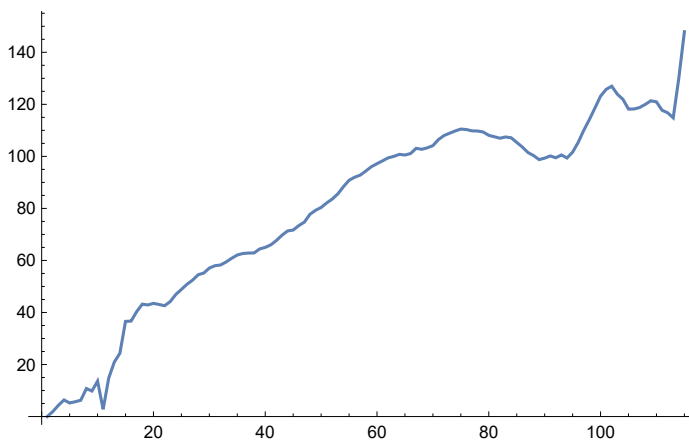
```
In[ ]:= PerformanceStats[tsStrategyReturnsNoFLR] // Dataset
```

```
Out[ ]:=
```

Start Date	Tue 4 Feb 2020
End Date	Tue 15 Nov 2022
Interval	2.77808 yr
No. Trades	114
Total Return	1.47912
CAGR	0.386537
Annual Volatility	0.138652
Annual SemiDeviation	0.108419
Max Drawdown	-0.0953269
Max Drawdown Date	Thu 12 Mar 2020
Sharpe ratio	2.78783
Sortino ratio	3.56522
Calmar ratio	4.05486
Avg. Return	0.0129747
Win/Loss ratio	1.62454
Profit Factor	4.54871
Win Rate	0.736842

```
In[ ]:= ListLinePlot[FoldList[Times, 100, 1 + tsStrategyReturnsNoFLR["Values"]] - 100]
```

```
Out[ ]:=
```





---

## Conclusion

Anomaly detection is a new strategy R&D method that shows great promise. It works by finding strategies in individual stocks with low correlations due to different times of occurrence of anomalies in the returns. This leads to excellent performance in portfolios of a few stocks.

However, the algorithms are demanding computationally, making this approach impractical for large stock portfolios. Additionally, it's harder to control the market or sector exposure of the portfolio compared to cross-sectional strategies.

## Notes

- More on anomaly detection techniques
- More on anomaly detection in Mathematica from Wolfram U