# Short Term Trading Models – Mean Reversion Trading Strategies and the Black Swan Events

Author Names:
*Mark Babayev (markb.quant@gmail.com)*
*Folakemi Lotun (mailflakie@gmail.com)*
*Goodwill Tatenda Mumvenge (gmvenge@gmail.com)*
*Ritabrata Bhattacharyya (ritabrata.bhattacharyya@wqu.org)*

***Abstract***

*This research analyzed the effectiveness of Black Swan strategies for the Short-Term Mean-Reversion systems, the risks and rewards profiles of such betting systems based on the S&P500 index. In determining the Black Swan events, the research made use of multiple strategies against two portfolios. By utilizing the python notebooks, signals created by the Black Swan and Bollinger Bands trading strategies were compared for performance against the baseline index (buy-and-hold strategy). This was followed by a validation of how risk mitigation techniques like the stop-loss affect the trading performance. The research concluded that it is possible to construct a Mean-Reverse strategy that outperforms the market over time.*

***Keywords****: Mean-Reverse, Black Swans, Grey Swans, White Swans, Jarque Bera test, IsolationForest, outliers, Bollinger Bands, buy-and-hold, stop-loss.*

# 1. Introduction

In the presence of Mean Reversion which implies the tendency of variables like stock prices to converge to an average value over time (Ronald Balvers et al. (2000)), various trading strategies have been developed to take advantage of mean reverting behaviors (Fischer and Riedler (2014)). While mean reversion has been ridiculed as being unrealistic (Lo and MacKinlay (1988), practical examples of mean reversion in prices have been found in a number of empirical studies including Gatev et al. (2006), Leung and Li (2016) as well as Avellaneda and Lee (2010).

Another study found no statistically significant evidence that price movements in calendar time scale consistently deviate from randomness. There were only limited departures split between trend and mean-reversion depending on the time frame, prices and assets. [5]

A Black Swan event, according to Brunåker and Nordqvist (2013:5), is “one which could not be predicted in advance by (all but a very few of) the observers”. Because of Mean Reversion and Black Swan events, trading strategies need to be built that can take advantage of the two. In light of that, this research is carried out to analyze the effectiveness of Black Swan strategies for Short Term Mean Reversion systems, the risks and rewards profiles of such betting systems based on the S&P 500 index.

## 1.1. Problem Statement

Financial time series are multifractal, thus exhibiting non-Gaussian distribution, the presence of extreme values (outliers), and long-range dependent dynamics. A constant challenge for quantitative traders is these frequent adjustments of financial markets, often abruptly, due to surges such as government policy, influences of regulatory bodies and other macroeconomic effects. Predicting such changes is a daunting process albeit undertaken by quantitative traders. As pointed out by Barber and Odean (2000), trading is indeed hazardous to your wealth. After studying 66,465 households from 1991 to 1996, individuals who traded the most underperformed the market returns significantly even without considering transaction costs. While those results advocate for buy and hold strategy, we see traders buying and selling stocks frequently because they believe they may beat the market.

While it may be possible to beat the market, there's a need to appreciate the continuous change in trends of assets leading to adjustments in returns via shifts in their means, volatilities and correlation which impact the effectiveness of time series methods that rely on stationarity. These steady changes eventually lead to dynamically-varying correlation, excess kurtosis (fat tails), heteroskedasticity as well as skewed returns.

The need for accurate quantitative modeling necessitates effective detection and categorization of these trends in order to optimally select quantitative trading strategies and select the best parameters within them. The modeling task then becomes an attempt to identify new trends and adjust strategy deployment, risk management and position sizing criteria accordingly.

Technical analysis uses indicators designed to help a trader determine whether current behavior is indicative of a particular trend and also the timing of a potential future trend. However, a technical indicator always needs to be used within a time frame and the problem of determining the best time frame can be the solution to most quantitative modeling problems (Fernandez-Blanco et al. [2008]), which requires sophisticated research.

### 1.2. Background

This work has been inspired by the thesis of Fabian Brunåker & Andreas Nordqvist, "A Performance Evaluation of Black Swan Investments". [2]

Their thesis evaluated an investment strategy that was exploiting the market's reaction to unpredictable events, called Black Swans. They created a portfolio, consisting of ten of the stocks from the Stockholm Stock Exchange with the largest price change after days with extreme negative returns and the ten of the stocks with the least change in price after extreme positive returns. The authors used various factors to define the Black Swan events, among them the Jarque-Bera statistics. In their work they used the mean-reversion strategy based on the law of large numbers that states that the sample mean approaches the true mean if the number of observations is large enough.

### 1.3. Black, White and Grey Swan Events

According to Nassim Taleb, the originator of the catchphrase "Black Swans" through his book, "The Black Swan: The Impact of the Highly Improbable". A black swan event means a highly improbable event with three principal characteristics: its unpredictable; its effect is massive; and, after the incident had occurred, we formulate an explanation that makes it appear less random and more predictable than it was. Recently, especially since the financial crisis, the wave of bad news in markets suggests unexpected events with a major impact are happening increasingly frequently.

Grey Swan on the other hand depicts a highly probable event with three principal characteristics: its predictable; its impact can easily escalate; and, after the incident had occurred, we formulate an explanation that recognizes the probability of occurrence but shifts our focus to errors in judgment or some other human form of causation. As events earlier classified as highly improbable i.e Black Swans are increasing in frequency, most of those events earlier referred to as Black Swan are now reclassified as Grey Swan ( the risk team at PwC put out a paper in 2012 called: "Black swans turn grey" [15]). This is not because these events have less severe impacts but because they have become more probable than before. Recent examples are; the US deficit, the eurozone crisis, political concerns in the Middle East, and a slowdown in China.

Lastly, a white swan event is a highly certain event with three principal characteristics: its certain; it affects an impact that can easily be estimated; and, after its occurrence, we formulate an explanation that agrees with the certainty of occurrence though we tend to shift our focus to errors in judgment or some other human form of causation. Some examples of common white swan events are Healthcare, economic growth and increase in population.

For our study, we chose Black Swan events to demonstrate the working of the models. However, our models can work seamlessly with white and grey swan events, few parameters only need to be changed to accommodate more events that are not absolutely outlier

### 1.4. Research Questions

This project should answer the following hypothesis:

**Hypothesis 1:**

**$H_{0:}$** It is not possible to construct a Mean-Reverse strategy solely based on the BB indicator that outperforms the market over time.

**$H_{1:}$** It is possible to construct a Mean-Reverse strategy solely based on the BB indicator that outperforms the market over time

**Hypothesis II:**

**$H_0$**: It is not possible to construct a Mean-Reverse strategy based on the Black Swan events that outperforms the market over time.

**$H_1$**: It is possible to construct a Mean-Reverse strategy based on the Black Swan events that outperforms the market over time.

## 2. Theoretical Framework

### 2.1. Mean-Reversion

Mean reversion trading is the theory which suggests that prices, returns, or various economic indicators tend to move to the historical average or mean over time. The research is still inconclusive about whether stock prices revert to the mean. Some studies show the mean reversion in some data sets over some periods, but many others do not. For example, in 2000, Ronald Balvers, Yangru Wu and Erik Gilliland found some evidence of mean reversion over long investment horizons.[3]

### 2.2. Bollinger Bands

The BB indicator is a basic indicator that uses the Mean-Reverse concept.
John Bollinger, creator of the Bollinger Bands® defines Bollinger bands as "a technical analysis tool, they are a type of trading band or envelope". Bollinger bands use a statistical measure known as the standard deviation, to establish where a band of likely support or resistance levels might lie. This is a specific utilization of a broader concept known as a volatility channel.

A volatility channel plots lines above and below a central measure of price. These lines, also known as envelopes or bands, widen or contract according to how volatile or nonvolatile a market is.[1]

Further we are going to show that the Bollinger Bands alone cannot be considered a sign for buy or sell signals, they just show when the asset is overbought or oversold.

### 2.3. Black Swan events

A Black Swan event is “one which could not be predicted in advance by (all but a very few of) the observers”, F. Brunåker & A. Nordqvist (2013:5). It’s important to note that Black Swan events are considered outliers in econometric meaning values that deviate from the rest of the data (Newbold, 2013). According to Taleb (2007a), “most life changing moments in our lives are a result of randomness and uncertainty, our inability to predict these events, even by so-called “experts”, and our invariable reaction to these events (surprise)”. Thus, a Black Swan also includes “the absence of an expected event”.

For an event to be termed a black swan, some conditions must be met:

- The event must be unpredictable.
- The event causes a major impact. It should be noted that event itself and its consequences may either be positive or negative for example wars, natural disasters and technical innovations. Also, black swan theory also applies to events happening to an individual and not necessarily a large number of people all the time.
- The ex post explanation where there must be interests in finding explanations for the occurrence of the event after it has happened.

Thus, the Black Swan events can be defined using the following techniques[2]:

- ≥5% monthly decrease/increase [3],
- ≥1.5% daily decrease/increase [4],
- > three standard deviations from the mean,
- Fails Jarque-Bera test for normality

**Jarque-Bera statistics**

The null hypothesis, that the sample is normally distributed, is rejected if the JB statistic is larger than the significance points [5].

### 2.4. Black Swan Investment Strategy

We invested in stocks which have had the largest percentage decrease or increase after an extreme event. Based on the mean reversion assumption, we expect these stocks to come back to their fundamental mean values thus providing us the highest profit.

We can go either long or short relevant to the negative or positive Black Swan event.

The Black Swan events should give much stronger signals for the trade than the regular Bollinger Bands strategy, that’s why we expect to see better results here.

## 3. Methodology

### 3.1. Data

We took 500 stocks from S&P500 index, choose from them a dozen stocks with the largest number of extreme events and a dozen stocks with the lowest value of the Jarque Bera test that validates the asset normality. All of them traded by the various trading strategies, among them the BB indicators. From the S&P500 assets we excluded the Netflix Inc. (NASDAQ:NFLX) because when it first listed on the stock exchange it has had an enormous performance and could solely influence the whole portfolio, as well as other assets that doesn’t exist in Quandle before year 2010, for example Alphabet Inc. (NASDAQ:GOOG). Some assets inside the portfolio are highly correlated among themselves. We could remove them too but still left them as is for the sake of simplicity.


**Figure 1. Netflix Inc.**

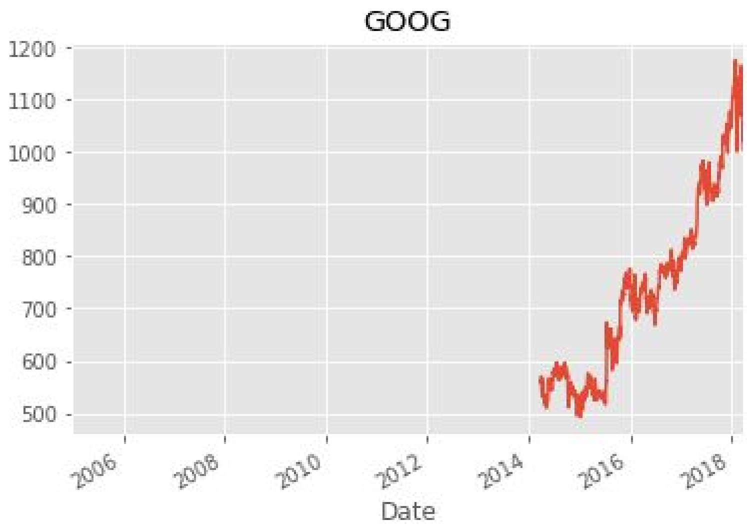


**Figure 2. Alphabet Inc.**

### 3.2. Selecting assets with Black Swan events

We selected the assets based on the 3 types of Black Swan events:

1. **Daily Events**: The current asset price is different in 10% from the 5-day rolling mean
2. **Monthly Events**: The current asset price is different in 20% from the 20-day rolling mean
3. **3-STD Events**: The current asset price lays out of the 3 standard deviations with the 20-day rolling mean.

In addition, we calculated the asset's Jarque Bera test to see how close they are to the normal distribution.

**Table 1. Assets with the highest number of Black Swan events**

| | Daily events | Monthly events | 3 STD | JB test |
|---|---|---|---|---|
| **UAL** | 126 | 166 | 22 | 155 |
| **ISRG** | 52 | 68 | 45 | 1025 |
| **RMD** | 8 | 1 | 40 | 489 |
| **AMD** | 95 | 135 | 21 | 2941 |
| **LVS** | 101 | 151 | 14 | |
| **MNST** | 46 | 60 | 37 | 487 |
| **MGM** | 101 | 133 | 21 | 1762 |
| **ABMD** | 62 | 105 | 35 | 6020 |
| **ETFC** | 111 | 115 | 18 | 1536 |
| **AAL** | 162 | 245 | 15 | 2035 |
| **DGX** | 3 | 1 | 35 | 1767 |
| **AIG** | 105 | 130 | 24 | 756 |

**Table 2. Assets with the minimum value of the Jarque Bera test**

| | JB test |
|---|---|
| **BEN** | 54 |
| **COF** | 61 |
| **EXPD** | 71 |
| **HPQ** | 51 |
| **JWN** | 55 |
| **KIM** | 67 |
| **LNC** | 75 |
| **NEM** | 34 |
| **PFG** | 69 |
| **SLB** | 73 |
| **TXT** | 50 |
| **VNO** | 40 |

The Jarque Bera test shows that the assets with the lowest number of the Black Swan events are more normal than their counterparts. The p-value for all tests is 0.0 except for the LVS that has p-value of 0.86.

All assets were normalized and added into 2 equally weighted portfolios with the highest number of Black Swan events and the lowest value of the Jarque Bera test.

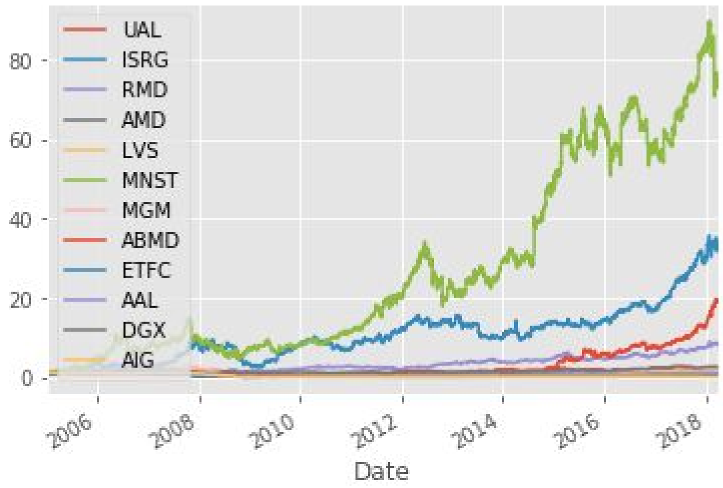


**Figure 3. Assets with Black Swan events**

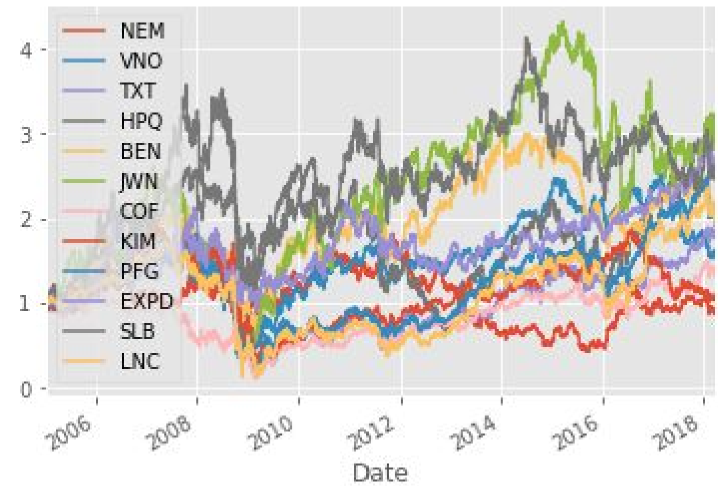


**Figure 4. Assets with the lowest value of the Jarque Bera test**

It’s interesting to note that the absolute majority of the extreme events occurred at the time of the 2008 financial crisis. For example, the American Airlines Group Inc. (NASDAQ:AAL) has been affected by the 2008 financial crisis:

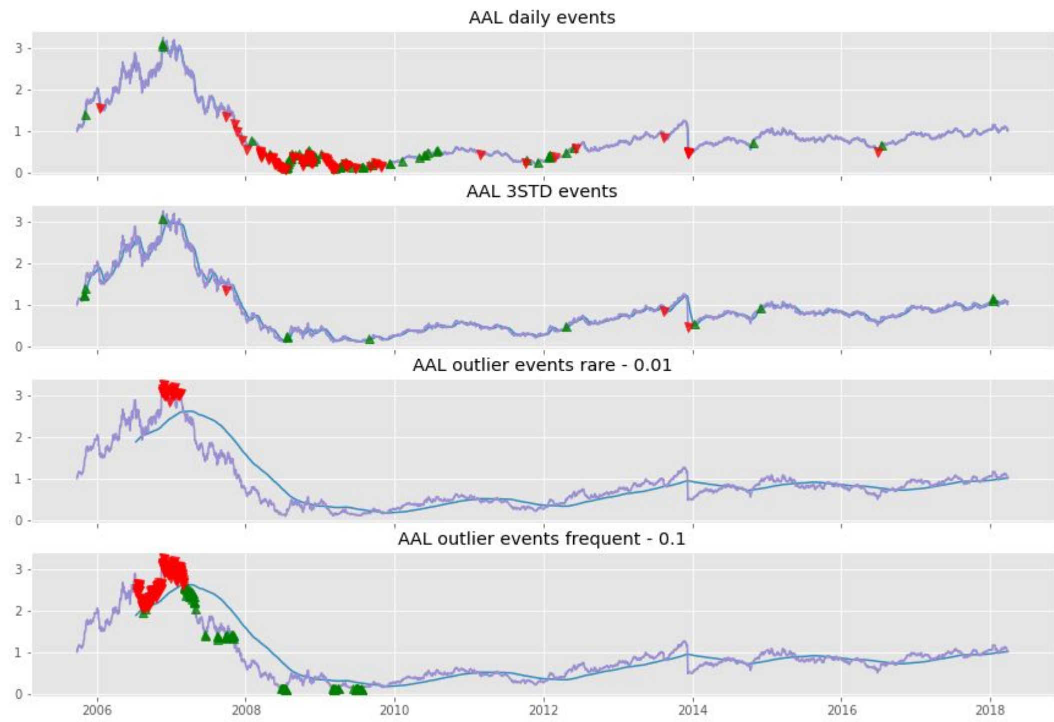


**Figure 5. AAL asset with Black Swan trade signals**

### 3.3 The Black Swans Trading Strategy

We used 3 different Black Swan strategies:

1. **Daily Events**: The current asset price is different in N% from the K-day rolling mean
2. **3-STD Events**: The current asset price lays out of the 3 standard deviations with the K-day rolling mean.
3. **Outlier Events**: The trading signals were selected from the IsolationForest outliers with the contamination of N. The IsolationForest is the Machine Learning algorithm implemented in the python's sklearn library. The outliers are compared against the asset's K-day mean.

*The IsolationForest 'isolates' observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature.* [13]

When Black Swan event is located above the mean we consider it the short event, and when it is below the mean - we consider it the long event. The long/short trades are closed only when opposite trade starts without considering any other conditions, like crossing means.

### 3.4. The Bollinger Bands Trading Strategy

In BB indicators we compared rolling mean with STDs against another quick rolling mean instead of the real asset in order to decrease the number of trades. For example, the quick rolling mean can be the 5-day and the STD can be calculated based on the 20-day rolling mean.

The trading signals for the BB indicator:

- When asset crosses the **Upper Bollinger Band** from below means that the asset is overbought and we open a short position and sell one unit of the asset.
- When asset crosses the **Lower Bollinger Band** from above means that the asset is oversold and we open a long position and buy one unit of the asset.
- When asset crosses the **Bollinger Mean** we close all existing positions both long and short.

All trading strategies generate many identical sequential signals. In that case we take into account only the first signal and the rest identical signals remain unattended.

Please see the Appendix for an example of trading the whole S&P500 index (Yahoo ^GSPC).

### 3.5 Tuning Hyperparameters

We used the Bayesian Optimization algorithm for choosing the best hyperparameters for our trading strategies. The optimization function goal was to reach the maximum portfolio return after the 500 iterations. We know that the professional traders tend to use other targets for the optimization functions like, for example Sharpe ratio, VaR or CVaR. We will be using here the total asset returns for the sake of simplicity.

We are not using the cross-validations (separating dataset to the train/test datasets) for the sake of simplicity, so the results may be quite overfitted. We also neglect the trading costs and the costs of borrowing money for long trades and borrowing assets for short trades. We also don't rebalance the portfolio and calculate all future signals based on the whole historical data without yearly or quarterly cropping it.

The idea of Bayesian Optimization by using the Hyperopt library described in the [14]

- *Bayesian optimization chooses the next hyperparameters in an informed way, and as such spends more time evaluating areas of the parameter distribution it believes have the highest chance of bringing a cross-validation score improvement versus previous iterations.*
- *This can result in fewer evaluations of the objective function and better generalization performance on the test set compared to random or grid search.*
- *The relative benefits of Bayesian Optimization differ with the number of dimensions of the dataset and the size of the parameter grid. The larger the dataset and / or the parameter grid, the higher the potential for efficacy gains.*
- *Random Search can still outperform Bayesian Optimization as it could bump onto the optimal set of hyperparameters right at the start — just by sheer luck.*

The hyperparameters for each asset are calculated separately and all assets then combined together into an equally-weighted portfolio.

These are the used hyperparameters limits for every trading strategy:

- **Bollinger Bands**
    - stop-loss: from 0 to 0.6, step 0.05
    - asset mean: from 1 to 5, step 1
    - BB mean: from 10 to 900, step 5
- **Daily Black Swans**
    - stop-loss: from 0 to 0.6, step 0.05
    - asset mean: from 2 to 200, step 1

- break event: from 10% to 50%, step 5%

- **3-STD Black Swans**
  - stop-loss: from 0 to 0.6, step 0.05
  - asset mean: from 1 to 100, step 1
- **Outliers Black Swans**
  - stop-loss: from 0 to 0.6, step 0.05
  - asset mean: from 50 to 500, step 5
  - contamination: from 0.01 to 0.2, step 0.005

### 3.6. Evaluation methods

The Hypothesis proved by comparing the BB/BS total return against the baseline index total return. In addition, we also take into account the strategies' Sharpe ratios, Calmar ratios and the Win/Loss rates.

We are using the Calmar ratio because the Sharpe ratio assumes the normal distribution and therefore doesn't take into account tail risk, Black Swans etc. On the other hand, the Calmar ratio uses the maximum returns drawdown instead of the standard deviation. The maximum drawdown is the largest drop of returns, when considering sequential tops and bottoms. It is the largest peak to trough decline. [10][11]

The Win/Loss rate determines whether the strategy perform consistently, or the profit only concentrate on a few trades that is very profitable [12]

## 4. Results

### 4.1. Hypothesis I

It is not possible to construct a Mean-Reverse strategy solely based on the BB indicator that outperforms the market over time.

**The hypothesis is rejected** - the BB strategy profits for "Black Swans" portfolio (low normality) didn't reach the buy-and-hold values but outperformed in the "Jarque Bera" portfolio (high normality).

This is the comparison of the baseline equally weighted index against 2 portfolios with the "Black Swans" and the "Jarque Beras" assets, showing profit after the BB strategy.

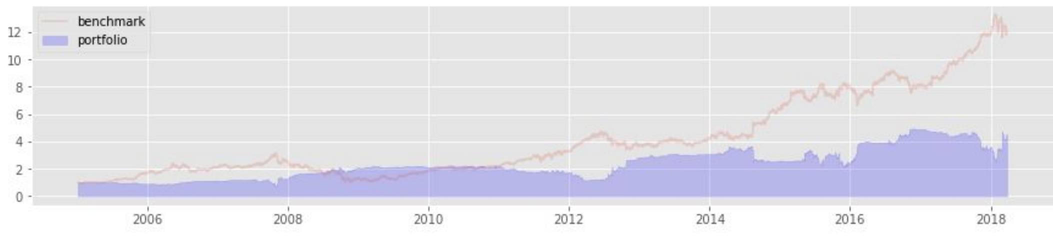


**Figure 7. The Black Swan events portfolio, comparing against the BB strategy**

The BB strategy for the "Jarque Bera" portfolio outperforms the portfolio's baseline index. We assume that it happens because of the relatively high normality of the containing assets.


**Figure 8. The Lowest Jarque Bera test portfolio, comparing against the BB strategy**

These are some examples of the asset trade signals and profits from the "Black Swans" portfolio.

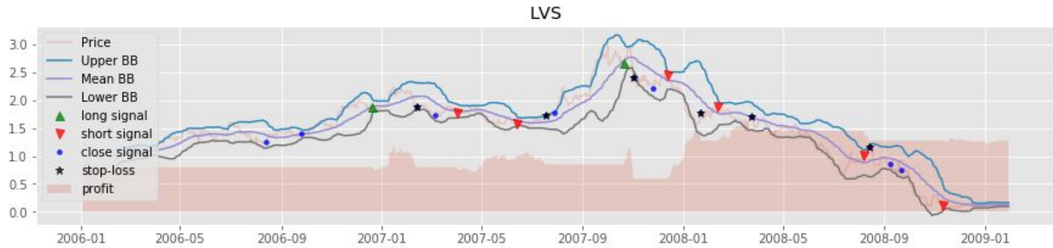


**Figure 9. LVS asset with BB signals and profit, "Black Swans" portfolio, years 2006-2009**

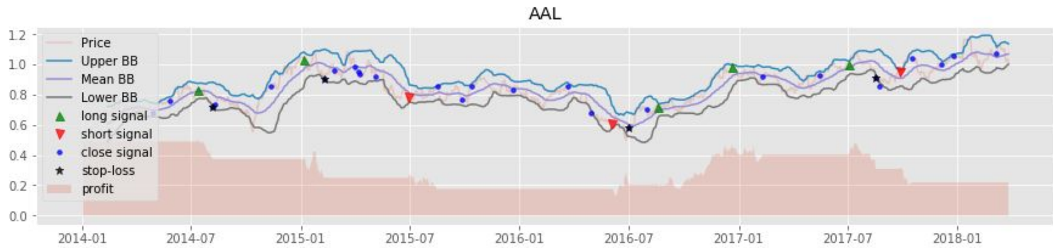


**Figure 10. AAL asset with BB signals and profit, "Black Swans" portfolio, years 2014-2018**

The next table shows the performance analysis of the BB trading strategy for both "Black Swans" and "Jarque Beras" portfolios.

**Table 3. BB performance analysis - "Black Swans" portfolio**

| | Total Return | Maximum Drawdown | Drawdown Duration | Sharpe ratio | Calmar ratio | Win/Loss ratio | Number of trades |
|---|---|---|---|---|---|---|---|
| **Buy-and-hold** | 10.8971 | 2.1413 | 999 | 0.0504 | 0.004 | | |
| **BB strategy** | 3.5022 | 2.3857 | 550 | 0.0321 | 0.002 | 1.756 | 350 |

**Table 4. BB performance analysis - "Jarque Beras" portfolio**

| | Total Return | Maximum Drawdown | Drawdown Duration | Sharpe ratio | Calmar ratio | Win/Loss ratio | Number of trades |
|---|---|---|---|---|---|---|---|
| **Buy-and-hold** | 0.9437 | 1.3275 | 1674 | 0.0203 | 0.0016 | | |
| **BB strategy** | 1.8724 | 0.206 | 333 | 0.0712 | 0.005 | 2.2926 | 540 |

### 4.2. Hypothesis II

It is not possible to construct a Mean-Reverse strategy based on the Black Swan events that outperforms the market over time.

**The hypothesis is rejected -** both, the "Daily" and "3-STD" Black Swan strategies outperform the buy-and-hold index.


**Figure 11. Comparing the "Black Swans" portfolio against the different types of Black Swan strategies**

This is an example of the Las Vegas Sands Corp. (NASDAQ:LVS) trade signals with profits after default and optimized hyperparameters:

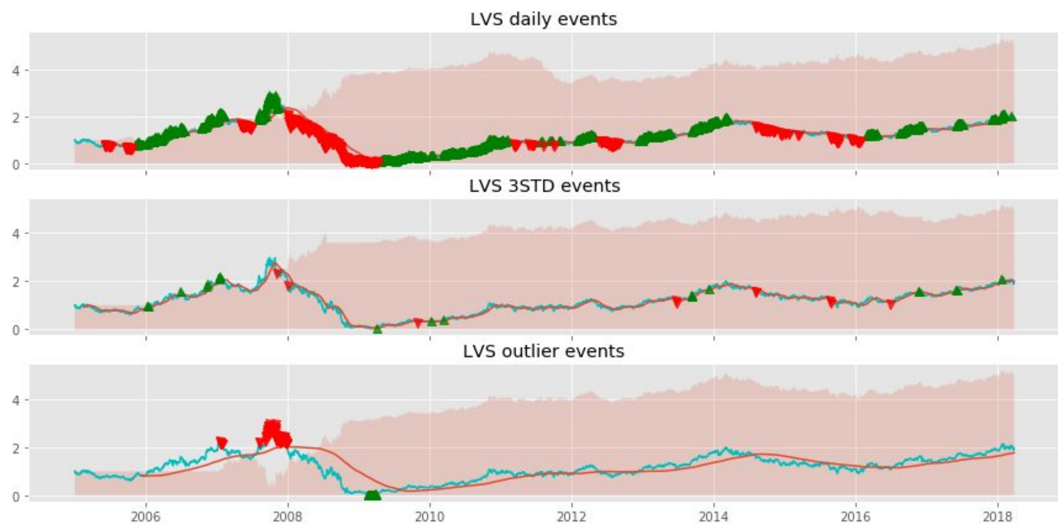


**Figure 12. LVS asset with BS signals and profit (optimized hyperparameters)**

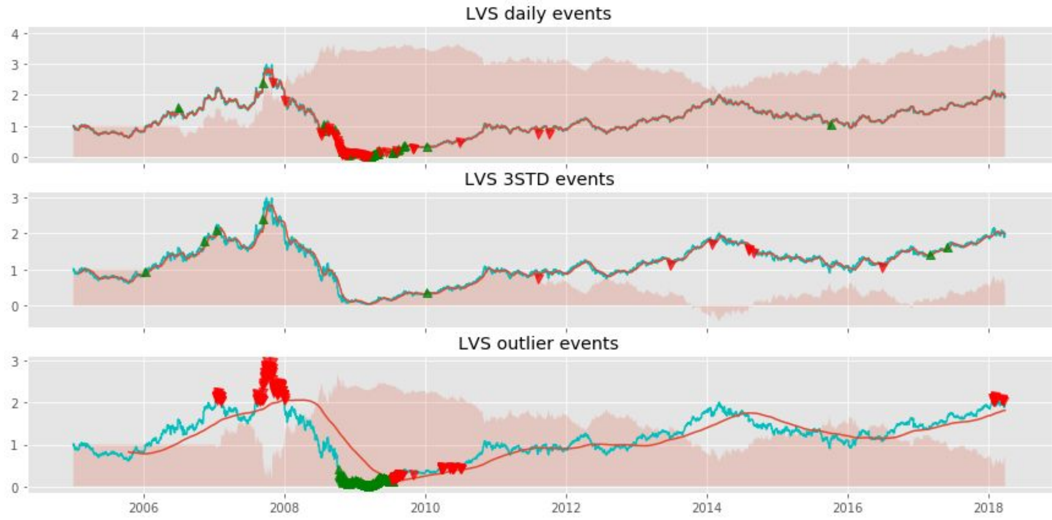


**Figure 13. LVS asset with BS signals and profit**
**(default hyperparameters - for details please see Jupyter Notebooks)**

An interesting case occurred with Monster Beverage Corporation (NASDAQ:MNST) asset. We've bought it at the beginning of the trades and all the way on just kept the asset untouched without any move:

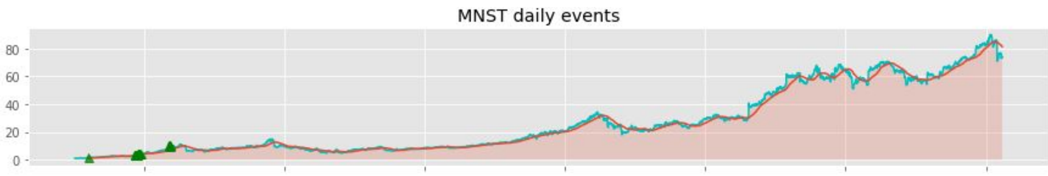


**Figure 14. MNST asset with BS Daily signals and profit**

The next table shows the performance analysis of the Black Swan trading strategies and the baseline "Black Swans" portfolio.

**Table 5. BS performance analysis - values**

| | **Total Return** | **Maximum Drawdown** | **Drawdown Duration** | **Sharpe ratio** | **Calmar ratio** | **Win/Loss ratio** | **Number of trades** |
|---|---|---|---|---|---|---|---|
| **Buy-and-hold** | 10.8971 | 2.1413 | 999 | 0.0504 | 0.004 | | |
| **BB** | 3.5022 | 2.3857 | 550 | 0.0321 | 0.002 | 1.756 | 350 |
| **BS Daily** | **12.3826** | **1.7265** | **542** | **0.073** | **0.0056** | **0.9354** | **60** |
| **BS 3-std** | **12.4733** | **1.9017** | **377** | **0.0774** | **0.006** | **1.3333** | **42** |
| **BS Outliers** | 5.8877 | 3.1484 | 972 | 0.0428 | 0.0025 | 1.7 | 54 |

**Table 6. BS performance analysis - ranks**

| | Total Return | Maximum Drawdown | Drawdown Duration | Sharpe ratio | Calmar ratio | Win/Loss ratio | Number of trades |
|---|---|---|---|---|---|---|---|
| **Buy-and-hold** | 3 | 3 | 5 | 3 | 3 | | |
| **BB** | 5 | 4 | 3 | 5 | 5 | 1 | 4 |
| **BS Daily** | **2** | **1** | **2** | **2** | **2** | **4** | **3** |
| **BS 3-std** | **1** | **2** | **1** | **1** | **1** | **3** | **1** |
| **BS Outliers** | 4 | 5 | 4 | 4 | 4 | 2 | 2 |

If we stack different strategies together and use for the asset only the strategy with the highest total return we would achieve even better results.

**Table 7. All assets total returns for BB/BS strategies**

| | **Benchmark** | **BB** | **BS daily** | **BS 3std** | **BS outliers** | **Best Strategy** |
|---|---|---|---|---|---|---|
| **UAL** | 1.1593 | 1.9382 | 2.8210 | 2.3456 | 3.4133 | BS outliers |
| **ISRG** | 31.22 | 9.1893 | 34.0413 | 46.0914 | 13.6798 | BS 3std |
| **RMD** | 7.4848 | 2.8881 | 6.0168 | 4.8792 | 2.9798 | BS daily |
| **AMD** | -0.5329 | 0.5478 | 2.2849 | 1.8939 | 0.3549 | BS daily |
| **LVS** | 0.951 | 1.5198 | 4.1763 | 4.0143 | 4.0563 | BS daily |
| **MNST** | 72.4893 | 17.0081 | 71.8984 | 61.067 | 46.5253 | BS daily |
| **MGM** | -0.0165 | 0.8435 | 2.6377 | 2.6216 | 3.3987 | BS outliers |
| **ABMD** | 18.0826 | * | 18.5389 | 18.7161 | -8.5689 | BS 3std |
| **ETFC** | -0.6321 | 0.1994 | 1.7886 | 2.1127 | -0.2763 | BS 3std |
| **AAL** | 0.015 | 2.6122 | 2.2107 | 3.5952 | 4.0927 | BS outliers |
| **DGX** | 1.4911 | 1.6529 | 1.2069 | 1.1837 | 1.0615 | BB |
| **AIG** | -0.9456 | 0.1248 | 0.9694 | 1.1586 | -0.0648 | BS 3std |

[*] The best ABMD return for the Bollinger Bands trading strategy occurs when we don't trade it all. Any other hyperparameter combination returns negative results.

## 5. Discussion

From the above analysis and results, it has been made very clear that the BB strategy tries to recognise an ending point of a downward or an upward trend and does not chase momentum. Thus, it speculates on reversal to the fundamental value which is mean reversion. According to this strategy, the stock price should fall when a stock is overbought, but would only signal a 'sell' as soon as the downward trend begins. Since the bands are two standard deviations below and above the moving average, this alternatively provides a relative definition of high and low. However, because stock prices always reflect the true value of firms as per the efficient market hypothesis (Fama, 1995), no overbuying or overselling can be witnessed. As expected, in an efficient market, like the S&P 500, the BB strategy could not generate abnormal returns.

As pointed out by Vargas and Estrada (2012), mean reversion must be present for the strategy to outperform the market. Their assumption was that indices do not follow random walk but are mean reverting. We assumed mean reversion for our trading strategy. Although there's inconclusive evidence about mean reversion, DeBondt and Thaler (1985) found "mean reversion to occur in the second and third year of the test period, with only little mean reversion within the first 12 months". In addition, both Bali et al. 2008 and Spierdijk et al. 2012 found evidence of mean reversion absorbing half of a shock after four to eight years. Furthermore, mean reversion was proved by Gatev et al. (2006), Leung and Li (2016) as well as Avellaneda and Lee (2010). The findings from the above-mentioned researchers therefore supports our findings. If mean reversion did not exist, then our findings would have no empirical backbone which is that of mean reversion.

Our findings tally with Joost de Man (2014) who looked at the STOXX Europe 600 and assessed the impact of the contrary BB strategy. His findings were that the BB strategy did not outperform the market. In addition, the BB strategy with the shortest-term moving average was found to generate the greatest abnormal returns, considering transaction costs.

Furthermore, the research has found that it is possible to construct a Mean-Reverse strategy based on the Black Swan events that outperforms the market over time. Before the 2008 financial crisis, the prices of stocks were highly overbought. Because the BB strategy recognized those overbought stocks, it provided investors with a sell signal. Thus, this strategy could outperform the market in the presence of black events. As noted by Martin & Ventura (2011), most crises are preceded by speculative bubbles where asset prices are far above the intrinsic values due to investor sentiments. However, bubbles do not last forever, as they will burst one day. Thus, in the event of a Black Swan event, the strategy generated abnormal returns and outperformed the market as it generated the buy-signals during economic decline as well as the sell-signals when it bottomed and began to recover (Chuen & Gregorio, 2014).

### 5.1. Contributions to literature

This research contributes to literature in several ways. Firstly, this study investigated mean reversion trading strategies and the black swan events through constructing portfolios on the S&P 500 as highlighted in the methodology. This trading strategy provided can be applied by the investors who prefer sensation seeking and are less than fully rational. Moreover, the Bollinger Bands strategy applied on S&P 500 contributes to the existing literature which is very limited. Apart from that, it has been proven that the BB strategy generates higher returns than the Buy and Hold strategy before transaction costs which contradicts Barber and Odean (2000) who concluded that "excessive trading by investors will underperform the B&H strategy, even before transaction costs" Last but not least, this research can help traders to build Black swan trading strategies to exploit mean reversion in security prices as it has been proven.

# 6. Summary of research findings, conclusions and recommendations

## 6.1. Introduction

This chapter outlines the summary of research findings and gives the conclusions that can be drawn from the study. In addition, it also gives recommendations to traders and paves the way for further studies in the subject.

## 6.2. Summary of research findings

This research analysed the effectiveness of Black Swan strategies for the Short Term Mean Reversion systems, the risks and rewards profiles of such betting systems based on the S&P500 index. The research concluded that it is possible to construct a Mean-Reverse strategy solely based on the BB indicator that outperforms the market over time but only for the portfolio consisting from naturally distributed assets and that one can outperform the market with a mean reverse strategy based on the black swan events.

## 6.3. Recommendations to traders

As we clearly see Bollinger Bands indicator as an implementation of the Mean-Reverse strategy alone can be considered a sign for buy or sell signals for naturally distributed assets. For the rest of the assets, in addition to the BB signs, traders must also look for a general trend direction, that means it’s unclear exactly what kind of trading position each signal means, long or short. In any case the Black Swan events give much stronger signals and can be used for opening long or short trading positions.

## 6.4. Recommendations for Future Study

This research does endorse the existence of mean reversion on the S&P500, as it was assumed to exist. We recommend an investigation of mean reversion on the S&P500 index as well as the speed of reversion. Researchers like Lo and MacKinlay (1988), have found the evidence against mean reversion.

It is also worth adding to the study of Grey and White events. They are not as strong as the Black ones but can be predicted with greater probability, opposite to the Black that cannot be predicted at all.

# Disclaimer

# Appendix

## 1. Analysing S&P500 index

The S&P500 is a value-weighted index opposite to the equally-weighted portfolios that we concluded in this study.

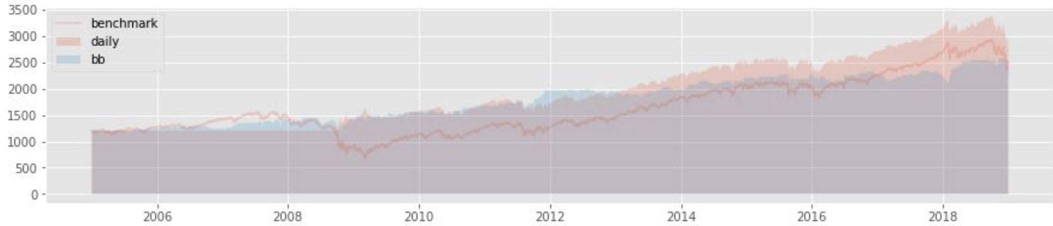


**Figure 14. S&P500 with BB/BS signals and their corresponding profit**

The original S&P500 index is outperformed by both Bollinger Bands and Daily Black Swan events.

**Table 7. BB/BS performance analysis for S&P500 index**

| | Total Return | Maximum Drawdown | Drawdown Duration | Sharpe ratio | Calmar ratio | Win/Loss ratio | Number of trades |
|---|---|---|---|---|---|---|---|
| **Baseline index** | 1.0854 | 888.62 | 1375 | 0.0235 | 0.0013 | | |
| **BB** | 1.1126 | 279.79 | 284 | 0.0335 | 0.0025 | 2.56 | 89 |
| **BS Daily** | 1.4592 | 579.65 | 285 | 0.0382 | 0.0021 | 0 | 2 |

**2. Analysing the Black Swans strategies with default hyperparameters without stop-losses and without doing the Bayesian Optimization.**

- Daily Black Swans:
  - asset mean: 5
  - break event: 0.1
- 3-STD Black Swans:
  - asset mean: 20
- Outliers Black Swans:
  - asset mean: 200
  - contamination: 0.1

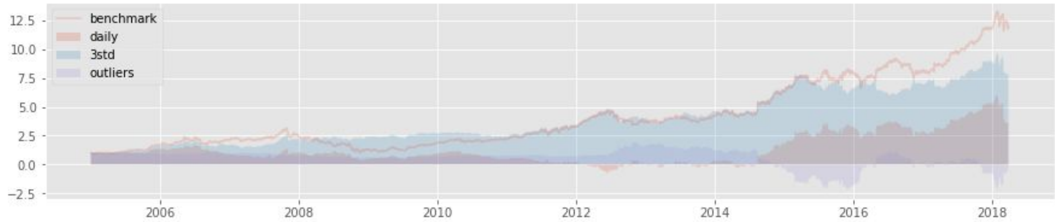


**Figure 15. Comparing the "Black Swans" portfolio against the different types of Black Swan strategies without the Bayesian Optimization**

As we can see, we receive much worse results when we are using the default well known parameters for the Mean-Reverse strategy.

**3. Selecting the portfolios (python source code)**

```
# Download the list of the S&P500 tickets from Wikipedia website
def save_sp500_tickers():
  resp = requests.get('http://en.wikipedia.org/wiki/List_of_S%26P_500_companies')
  soup = bs.BeautifulSoup(resp.text, 'lxml')
  table = soup.find('table', {'class': 'wikitable sortable'})
  tickers = []
  for row in table.findAll('tr')[1:]:
    ticker = row.findAll('td')[0].text.strip('\n')
    tickers.append(ticker)
  with open("sp500tickers.pkl", "wb") as f:
    pickle.dump(tickers, f)
  return tickers

# Upload all S&P500 assets and save them as separate DataFrame pickle files
def load_data(reload_sp500=False):
  if reload_sp500:
    tickers = save_sp500_tickers()
  else:
    with open("sp500tickers.pkl", "rb") as f:
```

```
      tickers = pickle.load(f)
  if not os.path.exists('stock_dfs'):
    os.makedirs('stock_dfs')

  start = dt.datetime(2005, 1, 1)
  end = dt.datetime.now()
  for idx, ticker in enumerate(tickers):
    # just in case your connection breaks, we'd like to save our progress!
    if not os.path.exists('stock_dfs/{}.pkl'.format(ticker)):
      time.sleep(5)
      try:
        df = web.DataReader(ticker, 'quandl', start, end)
        print('%s Obtained: %s' % (idx, ticker))
        df.reset_index(inplace=True)
        df.set_index("Date", inplace=True)
        if 'Symbol' in df.columns: df = df.drop("Symbol", axis=1)
        df.to_pickle('stock_dfs/{}.pkl'.format(ticker))
      except Exception as ex:
        print('Exception: ', ex)
    else:
      print('%s Already have: %s' % (idx, ticker))

# Take files saved in a previous function and and create from them one DataFrame with
their "Adj Close" prices
def compile_data():
  with open("sp500tickers.pkl", "rb") as f:
    tickers = pickle.load(f)

  main_df = pd.DataFrame()
  for idx, ticker in enumerate(tickers):
    print(idx)
    if os.path.exists('stock_dfs/{}.pkl'.format(ticker)):
      df = pd.read_pickle('stock_dfs/{}.pkl'.format(ticker))
      df = df[['AdjClose']]
      df.rename(columns={'AdjClose': ticker}, inplace=True)
      if main_df.empty:
        main_df = df
      else:
        main_df = main_df.join(df, how='outer')

  print('Obtained %s stocks. \n %s' % (len(main_df.columns), main_df.head()))
```

```
  main_df.to_pickle('sp500_joined_closes.pkl')

# Create dataset with assets that have the highest number of 3 types of Black Swan
events
def locate_black_swans(df):
  events = pd.DataFrame(index=df.columns, columns=('daily','monthly','3std'))
  for col in df.columns:
    asset = df[col].dropna()
    daymean = asset.rolling(window=5).mean()
    monmean = asset.rolling(window=20).mean()
    monstd = asset.rolling(window=20).std()
    events1 = asset[np.abs(asset/daymean - 1) > 0.1].size
    events2 = asset[np.abs(asset/monmean - 1) > 0.2].size
    events3 = asset[(asset>monmean+3*monstd) | (asset<monmean-3*monstd)].size
    events.loc[col,:] = (events1,events2,events3)

  evt1 = events.sort_values('daily', ascending=False)
  evt2 = events.sort_values('monthly', ascending=False)
  evt3 = events.sort_values('3std', ascending=False)
  joined_evt_max =
list(set(evt1.head(5).index.tolist()+evt2.head(5).index.tolist()+evt3.head(5).index.toli
st()))
  joined_evt_max = events.loc[joined_evt_max,]
  joined_evt_min =
list(set(evt1.tail(5).index.tolist()+evt2.tail(5).index.tolist()+evt3.tail(5).index.toli
st()))
  joined_evt_min = events.loc[joined_evt_min,]
  return joined_evt_max, joined_evt_min

# Because we will upload 500 assets in a row there is a need to put there the Quandle
API key
os.environ['QUANDL_API_KEY'] = '...'

# Go to the website "http://en.wikipedia.org/wiki/List_of_S%26P_500_companies" and take
from there the list of the S&P500 assets
save_sp500_tickers()

# Upload all S&P500 assets and save them as separate DataFrame pickle files
load_data()

# Take files saved in a previous function and and create from them one DataFrame with
their "Adj Close" prices
```

```
compile_data()

# Create dataset with assets that have the highest number of 3 types of Black Swan
events
joined_evt = locate_black_swans(df_joined)

# Save dataset to pickle file
df = df_joined[joined_evt_max.index]
df.dropna(axis=0, how='all', inplace=True)
# Normalize assets to make them all start with 1. We need it for building the
equally weighted index.
for col in df.columns:
  first_value = df.loc[df[col]>0,col]
  df[col] = df[col] / first_value.values[0]

df.to_pickle('sp500_swans.pkl')

# Create dataset with assets that have the lowest value of Jarque Bera test - the most
normally distributed assets
# Drop NFLX because it is too volatile and other assets that are missing data before
year 2007
df_joined.drop('NFLX', axis=1, inplace=True)
late_entry = df_joined.columns.difference(df_joined.loc[:'2007',:].dropna(axis=1,
how='all').columns)
df_joined.drop(late_entry, axis=1, inplace=True)

jbtests = df_joined.apply(lambda x:
stats.jarque_bera(x.dropna().values),axis=0).sort_values()
jbtests2 = pd.DataFrame(index=jbtests.index, columns=('JB test','JB p-value'))
jbtests2['JB test'] = jbtests.apply(lambda x:x[0])
jbtests2['JB p-value'] = jbtests.apply(lambda x:x[1])
jbtests2 = jbtests2[jbtests2['JB p-value']<0.0000001].sort_values('JB test').head(12)

# Save dataset to pickle file
df_jbtest = df_joined[jbtests2.index]
df_jbtest.dropna(axis=0, how='all', inplace=True)
# Normalize assets to make them all start with 1. We need it for building the
equally weighted index.
for col in df_jbtest.columns:
  first_value = df_jbtest.loc[df[col]>0,col]
  df_jbtest[col] = df_jbtest[col] / first_value.values[0]
```

```
df_jbtest.to_pickle('sp500_jbtest.pkl')
```

### 4. Creating the trading strategy (python source code)

```
"""
Calculate the strategy's profits based on the long and short signals.
We don't take into account repetitive signals, only the first one is considered.
We start our trading with a zero capital and a zero number of owned assets.
When we open a long position we borrow money to buy an asset and we extract the asset
cost from our capital.
When we open a short position we borrow an asset to sell it and we add an asset cost to
our capital and extract the number of owned assets.
When we finish the whole process and reach final trade we close open long positions if we
have owned assets or short positions if we still have borrowed assets.
The calculated profit value shows our total capital with the owned asset costs at every
moment of time.
"""
def BacktestAlphaSignal(asset, long_signal, short_signal, allow_repeat=False):
  long_position = 0
  short_position = 0
  # first index when the asset appeared in the stock exchange
  first_index = asset[asset>0].index[0]
  owned_capital = 0

  profit = pd.Series(index=asset.index)
  for idx in profit[first_index:].index:
    if long_signal is not None and idx in long_signal and long_signal[idx] != 0 and
(allow_repeat or long_position != long_signal[idx]):
      # when we open new long position our owned capital is declining
      long_position += long_signal[idx]
      if long_position < 0:
        long_position = 0
      else:
        owned_capital += -long_signal[idx]*asset[idx]

    if short_signal is not None and idx in short_signal and short_signal[idx] != 0 and
(allow_repeat or short_position != short_signal[idx]):
      # when we open new short position our owned capital is increasing
      short_position += short_signal[idx]
      if short_position < 0:
        short_position = 0
```

```
      else:
        owned_capital += short_signal[idx]*asset[idx]

    # current total capital including owned stock shares
    profit[idx] = (long_position-short_position)*asset[idx] + owned_capital

  # After finalizing trades we add to all calculated profit values the asset's starting
value to make the results more intuitive.
  if profit[profit!=0].size == 0:
    profit[:] = None
  else:
    profit += asset[first_index]

  return profit

# Split signal of type [1 - close short and open long, -1 - close long and open short]
# into 2 signal arrays, one for long signals and one for short signals.
def SplitSignals(signal):
  close_signal = signal[signal!=0].apply(np.abs)*-1
  long_signal = signal[signal>0].combine_first(close_signal)
  short_signal = (signal[signal<0]*-1).combine_first(close_signal)
  return long_signal, short_signal

# Taka long and short signals and generate table with trades (2 columns for opening and
closing position)
def GenerateTrades(asset, long_signal, short_signal):
  long_signal = NormalizeSignals(long_signal.copy(), asset.index[-1])
  short_signal = NormalizeSignals(short_signal.copy(), asset.index[-1])
  joined_trades =
pd.DataFrame(columns=('open','close','win'),index=long_signal.index.union(short_signal.in
dex))
  joined_trades.loc[long_signal.index, 'open'] = long_signal[long_signal>0]*asset
  joined_trades.loc[long_signal.index, 'close'] = long_signal[long_signal<0]*asset
  joined_trades.loc[short_signal.index, 'open'] = short_signal[short_signal>0]*-1*asset
  joined_trades.loc[short_signal.index, 'close'] = short_signal[short_signal<0]*-1*asset
  joined_trades['close'] = joined_trades['close'].shift(-1)
  joined_trades['win'] = (joined_trades['open']+joined_trades['close']).dropna() < 0
  return joined_trades.dropna(how='all')

# Remove repetitive signals because we don't use them in any case.
def NormalizeSignals(signal, last_index):
```

```
  value = 0
  for idx in signal.index:
    if signal[idx] == value:
      signal[idx] = 0
    else:
      value = signal[idx]

  signal = signal[signal!=0]
  if signal.size>0 and signal[0]<0:
    signal = signal[1:]
  if signal.size>0 and signal[-1]>0:
    signal.loc[last_index] = -1

  return signal

# Generate stop-loss signals. They are treated as regular close signals.
def StopLossSignal(asset, signal, is_long, stop_loss):
  stop_loss_signal = pd.Series()
  old_trade_period = None
  index = signal[signal!=0].index
  for i in range(index.size-1):
    if signal[index[i]]<0:
      continue

    trade_period = asset[index[i]:index[i+1]]
    if old_trade_period is not None:
      trade_period = old_trade_period.combine_first(trade_period)
      old_trade_period = None

    if is_long:
      cum_period = trade_period.cummax()*(1-stop_loss)-trade_period
      cum_period = cum_period[cum_period>0]
    else:
      cum_period = trade_period.cummin()*(1+stop_loss)-trade_period
      cum_period = cum_period[cum_period<0]

    if cum_period.size>0:
      stop_loss_signal[cum_period.index.values[0]] = -1
    elif signal[index[i+1]]>0:
      old_trade_period = trade_period
```

```
  return stop_loss_signal

# Calculate performance metrics: Sharpe ratio, Calmar ratio
def GetSharpeCalmar(asset):
  rets = asset/asset.shift(1)-1
  rets.fillna(0, inplace=True)
  rets.replace([np.inf, -np.inf], 0, inplace=True)
  sharpe = rets.mean()/rets.std()
  mins = np.ravel(argrelmin(rets.values))
  maxs = np.ravel(argrelmax(rets.values))
  extrema = np.concatenate((mins, maxs))
  extrema.sort()
  calmar = -rets.mean()/np.diff(rets[extrema]).min()
  return sharpe, calmar

# Calculate performance metrics: Total Return, Maximum Drawdown, Drawdown Duration
def PerformanceMetrics(asset):
  asset = asset[asset!=0]
  res = pd.Series()
  res['total_return'] = asset[-1]/asset[0]-1
  res['max_drawdown'], res['drawdown_duration'] = create_drawdowns(asset)
  res['sharpe'], res['calmar'] = GetSharpeCalmar(asset)
  return res

# Helper function for creating Maximum Drawdown metrics
def create_drawdowns(asset):
  hwm = [0]
  eq_idx = asset.index
  drawdown = pd.Series(index = eq_idx)
  duration = pd.Series(index = eq_idx)
  for t in range(1, len(eq_idx)):
    cur_hwm = max(hwm[t-1], asset[t])
    hwm.append(cur_hwm)
    drawdown[t]= hwm[t] - asset[t]
    duration[t]= 0 if drawdown[t] == 0 else duration[t-1] + 1
  return drawdown.max(), duration.max()

# Generate "Daily" Black Swan events
def GenerateBlackSwanSignalDaily(asset, day_ma, break_event):
  daymean = asset.rolling(window=int(day_ma)).mean()
```

```
  events1 = (asset[np.abs(asset/daymean - 1) >
break_event]-daymean).dropna().apply(np.sign)
  long_signal, short_signal = SplitSignals(events1)
  return long_signal, short_signal, daymean

# Generate "3-STD" Black Swan events
def GenerateBlackSwanSignal3STD(asset, mon_ma):
  monmean = asset.rolling(window=int(mon_ma)).mean()
  monstd = asset.rolling(window=int(mon_ma)).std()
  events3 = (asset[(asset>monmean+3*monstd) |
(asset<monmean-3*monstd)]-monmean).dropna().apply(np.sign)
  long_signal, short_signal = SplitSignals(events3)
  return long_signal, short_signal, monmean

# Generate Black Swan events based on the IsolationForest outliers
def GenerateBlackSwanSignalOutliers(asset, mon_ma, contamination):
  asset = asset.dropna()
  monmean = asset.rolling(window=int(mon_ma)).mean()
  isolation_forest = IsolationForest(n_estimators=100, contamination=contamination,
behaviour="new", n_jobs=-1)
  outlier = isolation_forest.fit_predict(asset.values.reshape(-1, 1))
  events4 = (monmean[outlier==-1]-asset[outlier==-1]).dropna().apply(np.sign)
  long_signal, short_signal = SplitSignals(events4)
  return long_signal, short_signal, monmean

# Generate regular open/close signals and append the them the stop-loss signals
def GenerateSignalsStopLoss(asset, func, stop_loss, *args, **kwargs):
  long_signal, short_signal, param = func(asset, *args, **kwargs)
  if stop_loss>0:
    stop_loss_long = StopLossSignal(asset, long_signal, True, stop_loss)
    long_signal = long_signal.combine_first(stop_loss_long)
    stop_loss_short = StopLossSignal(asset, short_signal, False, stop_loss)
    short_signal = short_signal.combine_first(stop_loss_short)
  return long_signal, short_signal, param

"""
Generate short and long signals for a BB strategy.
Both long and short signals behave the same: 1 - open position, -1 - close position.
For the sake of simplicity we consider the following rules:
- It's impossible to open/close different number of positions, only +1/-1
```

```
- The long and short positions don't overlap. That means, for example, when a short
position starts the long position ends.
"""
def GenerateBBSignals(asset, asset_ma, bb_ma):
  rolling_asset = asset.rolling(window=int(asset_ma)).mean() if asset_ma>1 else asset
  rolling_ma = asset.rolling(window=int(bb_ma)).mean()
  rolling_std = asset.rolling(window=int(bb_ma)).std()
  bb_upper1 = rolling_ma + 2*rolling_std
  bb_lower1 = rolling_ma - 2*rolling_std

  short_signal = (rolling_asset-bb_upper1).dropna().apply(np.sign).diff().dropna()//2
  long_signal = (bb_lower1-rolling_asset).dropna().apply(np.sign).diff().dropna()//2
  close_signal = (rolling_asset-rolling_ma).dropna().apply(np.sign).diff().dropna()//2
  close_signal = close_signal[close_signal!=0].apply(np.abs)*-1
  long_signal = long_signal[long_signal>0].combine_first(close_signal)
  short_signal = short_signal[short_signal>0].combine_first(close_signal)
  return long_signal, short_signal, (rolling_ma, bb_upper1, bb_lower1)

# Analyze portfolio performance
def AnalyzePortfolioPerformance(df, func, best_params):
  if not 'stop_loss' in best_params: best_params['stop_loss'] = 0
  returns = pd.Series(index=df.columns)
  assets_profit = pd.DataFrame(columns=df.columns, index=df.index)
  winloss_ratio = pd.DataFrame(index=df.columns, columns=('wins','losses'))
  for col in df.columns:
    # Generate regular and stop-loss signals
    long_signal, short_signal, _ = GenerateSignalsStopLoss(df[col], func,
**best_params.loc[col,:].to_dict())
    # Backtest trading strategy and calculate historical profits
    profit = BacktestAlphaSignal(df[col], long_signal, short_signal).dropna()
    if profit.size>0 and profit[0]!=0:
      returns[col] = profit[-1]/profit[0]-1
    assets_profit.loc[:,col] = profit
    # Calculate the total number of trades and the Win/Loss ratio
    joined_trades = GenerateTrades(df[col], long_signal, short_signal)
    winloss_ratio.loc[col,:] = (joined_trades[joined_trades['win']==True].shape[0],
joined_trades[joined_trades['win']==False].shape[0])

  portfolio = assets_profit.mean(axis=1).dropna()
  portfolio_performance = PerformanceMetrics(portfolio)
```

```
  portfolio_performance['trades'] =
winloss_ratio['wins'].sum()+winloss_ratio['losses'].sum()
  portfolio_performance['win/loss'] =
winloss_ratio['wins'].sum()/winloss_ratio['losses'].sum() if
winloss_ratio['losses'].sum()>0 else 0
  return portfolio, portfolio_performance, returns

# Bayesian Optimization function for finding the best hyperparameters
def BayesOptimize(df_bayes, func, param_hyperopt):
  # Objective function for the Bayesian Optimization
  def objective_function(asset, params):
    long_signal, short_signal, _ = GenerateSignalsStopLoss(asset, func, **params)
    profit = BacktestAlphaSignal(asset, long_signal, short_signal)
    profit = profit.dropna()
    if profit.size>0 and profit[0]!=0:
      score = profit[-1]/profit[0]-1
    else:
      score = 0
    return {'loss': -score, 'status': STATUS_OK}

  if not 'stop_loss' in param_hyperopt: param_hyperopt['stop_loss'] = 0
  best_params = pd.DataFrame(index=df_bayes.columns, columns=param_hyperopt.keys())
  for col in df_bayes.columns:
    # trials - is the temporary results storage in memory. We can store results also in
MongoDB or other databases.
    trials = Trials()
    # fmin - function from "hyperopt" library that runs optimization for a maximum of 500
steps
    best_param = fmin(partial(objective_function, df_bayes[col]),
                      param_hyperopt,
                      max_evals=500,
                      algo=tpe.suggest,
                      trials=trials,
                      rstate= np.random.RandomState(1))

    # Printing the best params and a total return (total return has "-" sign because it
is a loss function)
    #loss = [x['result']['loss'] for x in trials.trials]
  best_params.loc[col,:] = pd.Series(best_param)
  return best_params
```

**5. Generate hyperparameters with the Bayesian Optimization algorithm (python source code)**

```
func = GenerateBBSignals
param_hyperopt = {
    'stop_loss': hp.quniform('stop_loss', 0, 0.6, 0.05),
    'asset_ma': scope.int(hp.quniform('asset_ma', 1, 5, 1)),
    'bb_ma': scope.int(hp.quniform('bb_ma', 20, 900, 5))
}

# Optimize hyperparameters with Bayesian Optimization algorithm
best_params = BayesOptimize(df, func, param_hyperopt)
best_params.to_pickle('sp500_params.pkl')
```

**6. Analyse portfolio performance - Bollinger Bands strategy (python source code)**

```
# Analyze portfolio performance of "Black Swans" assets with the Bollinger Bands strategy
performance = pd.DataFrame()

# The following function calculates the original baseline index performance metrics:
Total Return, Maximum Drawdown, Drawdown Duration, Sharpe Ratio, Calmar Ratio
performance['benchmark BS'] = PerformanceMetrics(benchmark)
performance.loc['win/loss',:] = None
performance.loc['trades',:] = None
returns = pd.DataFrame(index=df.columns)
returns['benchmark'] = df.apply(lambda x: x[-1]/x.dropna()[0]-1)

# The following function calculates the "Black Swans" portfolio profits performance
metrics
# In addition to previous metrics we also have the Win/Loss ratio and the total trades
number.
# For some reason the BB strategy had no trading signals for the ABMD asset
portfolio, performance['BB'], returns['BB'] = AnalyzePortfolioPerformance(df,
GenerateBBSignals, best_bb_params)

plt.figure(figsize=(15,3))
# Draw the baseline portfolio
plt.plot(benchmark, label='benchmark', alpha=.2)
# Draw the portfolio profits at every moment of time
plt.fill_between(portfolio.index, 0, portfolio, label='portfolio', alpha=.2, color='b')
plt.legend(loc='upper left')
plt.show()
print(performance.T)
```

```
print(returns)
```

**7. Analyse portfolio performance - Black Swans strategy (python source code)**

```
# Analyze the "Black Swans" portfolio performance with 3 Black Swan strategies: daliy,
3-std, IsolationForest outliers
portfolio1, performance['BS daily'], returns['BS daily'] =
AnalyzePortfolioPerformance(df, GenerateBlackSwanSignalDaily, best_bsd_params)
portfolio2, performance['BS 3std'], returns['BS 3std'] = AnalyzePortfolioPerformance(df,
GenerateBlackSwanSignal3STD, best_bs3std_params)
portfolio3, performance['BS outliers'], returns['BS outliers'] =
AnalyzePortfolioPerformance(df, GenerateBlackSwanSignalOutliers, best_bsifo_params)
returns['max'] = returns.drop('benchmark',axis=1).idxmax(axis=1)

plt.figure(figsize=(15,3))
plt.plot(benchmark, label='benchmark', alpha=.2)
plt.fill_between(portfolio1.index, 0, portfolio1, label='daily', alpha=.2)
plt.fill_between(portfolio2.index, 0, portfolio2, label='3std', alpha=.2)
plt.fill_between(portfolio3.index, 0, portfolio3, label='outliers', alpha=.2)
plt.legend(loc='upper left')
plt.show()
print(performance.T)
print(returns)
```

## References


1. Three Bollinger Bands Strategies That You Need to Know| https://admiralmarkets.com/education/articles/forex-strategy/three-bollinger-bands-strategies-that-you-need-to-know
2. A Performance Evaluation of Black Swan Investments https://gupea.ub.gu.se/bitstream/2077/33388/1/gupea_2077_33388_1.pdf
3. Estrada, J. Vargas, M. 2012. Black Swans, Beta, Risk, and Return. Journal of Applied Finance; Vol. 22 Issue 2, p77-89
4. Burnie, D. De Ridder, A. 2010. Far Tail or Extreme Day Returns, Mutual Fund Cash Flows and Investment Behavior. Applied Financial Economics, Vol. 20, Issue. 16; p1241–1256
5. Jarque-Bera test and its competitors for testing normality: A power comparison https://www.econstor.eu/bitstream/10419/49919/1/668828234.pdf
6. 4 Ways To Predict Market Performance https://www.investopedia.com/articles/07/mean_reversion_martingale.asp#ixzz4yj9oY5b9
7. Anatomy of Technical Trading: Black Swan Or Mean Reversion https://seekingalpha.com/instablog/167044-bo-peng/46996-anatomy-of-technical-trading-black-swan-or-mean-reversion
8. The Bollinger Bands Trading Strategy Guide https://www.tradingwithrayner.com/bollinger-bands-trading-strategy/
9. Sharp and Calmar Ratio http://ivanidris.net/wordpress/index.php/2014/03/01/sharpe-and-calmar-ratio
10. Ranking stocks with the Calmar and Sortino ratios https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781785282287/7/ch07lvl1sec94/ranking-stocks-with-the-calmar-and-sortino-ratios
11. Win/Loss Ratio Definition https://www.investopedia.com/terms/w/win-loss-ratio.asp
12. Isolation Forest Algorithm https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html
13. Using Bayesian Optimization to reduce the time spent on hyperparameter tuning https://medium.com/vantageai/bringing-back-the-time-spent-on-hyperparameter-tuning-with-bayesian-optimisation-2e21a3198afb
14. Grey Swan Definition from Financial Times Lexicon http://markets.ft.com/research/Lexicon/Term?term=grey-swan
15. Risk Practices: Black swans turn grey - The transformation of risk https://www.pwc.co.uk/services/risk-assurance/insights/risk-practices-black-swans-turn-grey-the-transformation-of-the-risk-landscape.html