Can sentiment indicators signal market reversals?

Arnaud Lagarde[1]

Abstract

As seen in a previous study[2], sentiment indicators, in particular those provided by Hong Kong based firm Amareos[3], can provide valuable market signals. In this study we use machine learning algorithm to test their predictive power of market reversals, and to build and test a viable trading strategy.

As input for the algorithm, we used eight market sentiment indicators (Anger, Anticipation, Disgust, Fear, Gloom, Joy, Optimism and Sentiment) on 20 major equity indices[4] from January 1, 2005 to April 15, 2016. As the target output, we use a classification of the performance of the indices on the following 182 days -approximately six months- split between bottom, top and neutral days.

Our learning algorithm is of the type called random forest. Through calibration on a training set composed of 64% of the data, we obtain a final set of decision trees, or forest. We then examine the out of sample accuracy of this forest on the remaining 36% of the data.

As the accuracy on the test set is relatively high – a result that cannot be explained just by luck - we simulate a trading strategy based on the forest output. The resulting trading strategy produces strong performance, certainly much better than a simple buy and hold, even when adjusted for risk.

[1] Arnaud Lagarde is CRO at Hong Kong based asset manager Mandarin Capital and advisor to Amareos on Risk Management matters. Email: alagarde@mdncapital.com

[2] In a previous study, we showed how they could provide meaningful tail risk protection – see: http://amareos.com/research/files/limit190216.pdf

[3] http://amareos.com/

[4] The full list comprises: ASX, CAC40, CSI300, DAX30, DJ30, EU50, FTSE100, HK50, IBEX35, IBOV, MSCI50, Nasdaq, Nifty50, Nikkei, RU50, Russell2000, SG30, SMI20, SP500 and TSX250

1 Introduction

Market sentiment has become a prominent subject of market research. A quick search on the social science research network (SSRN) for market sentiment turns more than 26,000 papers and counting.

This study has the advantage of using a unique sentiment dataset as its source data.

In the previous paper, we produced meaningful results in tail risk protection using aggregate measures: sentiment and optimism. For this study, we will use a richer dataset.

2 Learning process

2.1 The data and their processing

2.1.1 The input: sentiment indicators

Distributed by Amareos, together with their data partners, the sentiment indicators are derived from the analysis of more than two million finance articles daily from over 50,000 global news sources ranging from mainstream medias, to blogs, forums and other social media platforms such as Twitter.

The two core aggregate measures are sentiment and optimism.

Sentiment measures the positivity and negativity of references about the specific asset. The higher the measure, the better the view of the asset.

Optimism measures the future expectation of the specific asset based on the amount of optimistic and pessimistic references about the asset. The higher the measure, the more positive the optimism is on the asset.

For this study, we will add metrics on the intensity of six primary emotions towards the relevant asset [5]. Those primary emotions are:

Anger, Anticipation, Disgust, Fear, Gloom, Joy

2.1.2 The target output: Performance classification

For the classification process, we chose rather arbitrarily to focus on a 182 days timeframe, which roughly approximates to six months. Obviously, bull and bear markets, which are clearly of interest in a

[5] For the whole dataset and other details, please go to www.amareos.com.

study examining market tops and bottoms, are notoriously longer.[6] However, six months is a popular investment horizon for many asset managers and having a shorter time window has the additional advantages of helping us capture shorter-term market dynamics as well as reduce the amount of calculation needed.

In order to calibrate the algorithm it is necessary to summarize the future 182 days return for a given asset for every day into a single target number. To do this we define a relative distance from the current price to the minimum and maximum prices over the following 182 days. We use the logarithmic performance from the current day to those two extremes. The higher the performance from today to the highest point, the better the expected returns. Similarly, the smallest the drawdown from today to the lowest point, the less risky the expected returns.

For an asset A, on a date t, the formula to obtain our target number is of the form:

$$Target(A,t) = ln\left[\frac{Max\left(A_{\tau\in\{t\leq\tau\leq t+182\}}\right)}{A_t}\right] + ln\left[\frac{Min\left(A_{\tau\in\{t\leq\tau\leq t+182\}}\right)}{A_t}\right]$$

Except in the situation where the value of the asset is at the market top (when the first term becomes zero), the first term will necessarily be positive as at all other points in time the value of the asset at the start of the period will necessarily be smaller than its maximum.

By contrast, the second term will be negative for all values except at the market bottom when the second term also becomes zero.

Intuitively, the way to think about this target value is to consider the best possible performance over the six-month ahead period penalized by the worst possible performance over the same period.

Having calculated this target values, we then segment them into three classes: target values above 0.0953 will be qualified as bottoms, below -0.223 as tops, the remainder will be considered neutral.

Like the choice of timeframe, these classification limits are somewhat arbitrary.

By convention a bear market is defined as a 20% drawdown. So given we are using log returns and that ln(1-20%)=-0.223, it seemed natural to use this limit to define a top. Conversely, based on S&P500 market moves[7], we calculated the average move during a bull market month to be approximately half the amplitude of the average move during a bear market.

Hence, the use of ln(1+10%)=0.0953 as a limit for the definition of a bottom.

---

[6] 25 months for Bear markets and 54 months for Bull markets on the S&P according to JP Morgan, page 15 of their guide to the markets: https://am.jpmorgan.com/us/en/asset-management/gim/adv/insights/guide-to-the-markets/viewer

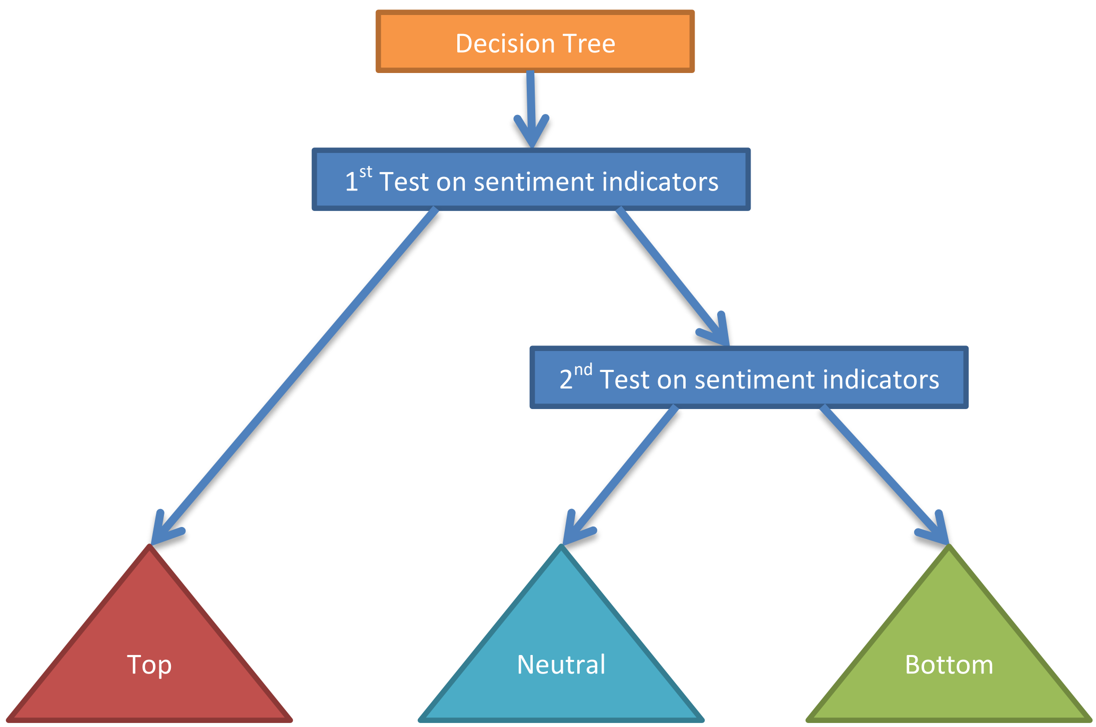
[7] Page 15: : https://am.jpmorgan.com/us/en/asset-management/gim/adv/insights/guide-to-the-markets/viewer

Using this classification split[8], the number of tops and bottoms are frequent enough for our algorithm to learn from the training dataset, but not overly so thereby avoiding problems with switching regimes too often.

2.2 The ensemble learning method: A few words on Random Forest

As we split the performance data into three classes, our dataset is well suited for classification tree models. Using the Amareos sentiment indicators as data input, the model will generate as outputs one of the three classes from our classification: Bottom, Neutral and Top.

An overly simplified version of such a tree is shown in the diagram below:



At each node (blue rectangles), a test is applied to the sentiment indicator inputs and depending upon the outcome of the test, a certain path is chosen; one leading either to another node or to a classification (triangles). Obviously, a classical decision tree is often more complex than the one above, with more nodes, and more paths but it serves to illustrate the basic premise underlying such models.

---

[8] The split over the whole dataset is roughly 28% Bottoms, 61% Neutrals and 10% Tops.

Moreover, because the world is definitely not so simple, we will apply a classifier based on a number of decision trees to improve further the classification rate. The specific classifier we deploy is called Random Forest[9] as it is constituted by a set of trees.

This set of trees or "forest" is simultaneously applied to the data. The output of the forest will be the mode, or most common output of all trees. For example, if a random forest is constituted of 3 trees, and the output of the trees are respectively {Sell, Neutral, Neutral}, the random forest output will be Neutral. In our case 120 trees form our forest, which provides a good compromise between the need to have a high number of trees and the time of training[10].

The set of trees is obtained through a relatively complex training procedure using part of the dataset. This procedure includes a randomized node optimisation, hence the name[11].

While trees are well known classifier, they have often tendency to overfit to the training set; one of the characteristic of Random Forest is to correct for this[12], which makes it appropriate for the particular task at hand[13].

---

[9] https://en.wikipedia.org/wiki/Random_forest

[10] A recent paper suggests the optimal number of trees in a random Forest is between 64 and 128. https://www.researchgate.net/publication/230766603_How_Many_Trees_in_a_Random_Forest

[11] We won't provide more details into the Random Forest technique, as for the interested reader, good explanatory texts from better authors are easily available For example: http://www-bcf.usc.edu/~gareth/ISL/ISLR%20First%20Printing.pdf

[12] http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#remarks

[13] More appropriate, in our view, than some more recent or more performing algorithm like xgboost

2.3 Implementation

To implement our Random Forest, we use the statistical language R[14], and more particularly the recent randomForest [15] package. The code used is available on demand and is reproducible as a seed[16] specified in the code has been used for the random number generation.

For the learning process, we first partition the data between two sets: training and testing. The R function createDataPartition[17] usually allows doing so in a simultaneously random and balanced manner.

One of the issues from our dataset comes from the fact that most of data, target values and sentiment indicators alike, are sticky from one day to the next and the reversals are relatively clustered[18],i.e. if a day is classified as a top, it is likely that the days just before and after, will also be classified as a top. To deal with this issue, we decided to remove some indices completely from the training set. While the accuracy is diminished as a consequence, the risk of overfitting is reduced.

First, we remove the two indices whose performances, target values and the subsequent reversals were the least correlated to the whole dataset[19]. Those indices are CSI300, and Nikkei. Second we removed from the training set the S&P 500 and the CAC40, to provide us with better geographic diversity in the test dataset.

We still used the createDataPartition function on the remaining indices to obtain the training data, removing 20% of the remaining data.

After this process, the training set is composed of a random choice of 80% of the data on the following indices: ASX, DAX30, DJ30, EU50, FTSE100, HK50, IBEX35, IBOV, MSCI50, Nasdaq, NIFTY50, RU50, RUSSELL2000, SG30, SMI20 and TSX250.

The remaining 20%, and the data of CSI300, Nikkei, CAC40 and S&P500 will constitute our testing dataset.

---

[14] https://www.r-project.org/

[15] https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

[16] http://www.inside-r.org/r-doc/base/set.seed

[17] http://www.inside-r.org/node/87010

[18] For more on the issue, http://www.tandfonline.com/doi/abs/10.1080/00949655.2012.741599

[19] See Annex for correlations tables

Originally the dataset was composed of 82,340 daily sentiment observations. Of those, only 54,524 had usable market data, due to the lack of trading over weekends and national holidays (while sentiment data is available every day). After removing the four indices as mentioned and a randomly selected 20% of the remaining data, we were left with a training dataset of 35,012 days or 64% of exploitable data.

We then use the training dataset for our random forest training process.

The function randomForest[20] in R from the homonymous package is used for the training. As explained before, the number of trees is fixed to 120. The function output is a list including all details of the training session, the final model and the calibrated model giving the best fit to the input data.

The predict function can then be applied to this final model on the test data inputs and compare the output (expected class) on the market classification Bottom/Neutral/Top.

[20] https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

3 The results

3.1 Accuracy of classification

The table below gives a quick outlook of the results for the training dataset; the lines represent the output of the Random Forest, and the column the target, i.e. the classification based on the actual real 6 month expectations. Not surprisingly, for the training data, the fit is perfect.

Training Data

		Target			
		Bottom	Neutral	Top	Accuracy of the output
Output	Bottom	10,372			100%
	Neutral		20,968		100%
	Top			3,672	100%
% of Target classified correctly		100%	100%	100%	

While these results merely tells us the training works, results table on data out of the training set are more relevant to check the value of the algorithm; those results are presented below in the same tabular format.

Testing Data

		Target			
		Bottom	Neutral	Top	Accuracy of the output
Output	Bottom	2,725	688	214	75.1%
	Neutral	3,104	10,797	1,054	72.2%
	Top	8	63	859	92.4%
% of Target classified correctly		46.7%	93.5%	40.4%	

The general accuracy rate of the output is 73.7%[21], which is relatively high. Arguably the accuracy rates for Tops and Bottoms are more important than for Neutral days and the number of false positives on tops/bottoms calls should be our biggest concern. 75%[22] of the calls for a bottom were correct and 92.36%[23] of the calls for a top were correct.

While the accuracy of the output is is impressive , with a low number of false positives, the algorithm does not pick up all the actual market tops or bottoms (detection rate). In fact, it detects only 46.7%[24] of the bottom signals and 40.4% of the tops. As bottoms and tops don't normally happen in a day, it may

[21] (2725+10797+859)/19512 data points
[22] 2725 out of 3627 data points, the first line in the table
[23] 859 out of 930 data points, the third line in the table
[24] First column of the table

not be an issue from a trading point of view. Through the strategy implementation, we will see this point later.

That said, we have to say that the results of the classification procedure are nearly too good to be true. This might reflect the impact of data clustering; but this can't affect the four indices removed from the training set, and they will provide a very useful cross check:

CSI300, CAC40, NIKKEI and S&P500

		Target			
		Bottom	Neutral	Top	Accuracy of the output
Output	Bottom	445	537	213	37.2%
	Neutral	2,792	5,746	955	60.5%
	Top	7	23	42	58.3%
% of Target classified correctly		13.7%	91.1%	3.5%	

The algorithm had trouble to pick up the right signals for bottoms and tops and made very little of those calls; only around 13.7% of bottom days and 3.5% of top days were detected[25]. This may be an issue for a trading strategy implementation.

At the same time, 37.2% of the bottom (445 out of 1195) and 58.3% of the top calls (42 out of 72) were correct. Nearly 18% of the bottoms call happened during tops (213 out of 1195) and 9.7% of top calls happened during bottoms (7 out of 72).

This level of imprecision is not surprising. As we have argued before sentiment can't and won't explain all market moves. Sharp moves are by definition extremes, and often the fruit of exogenous surprising events.

For example, Volkswagen (VOW) represents 3% of the DAX. This algorithm cannot be expected to predict events like the move of 45% up in VOW's stock following a short squeeze provoked by a Porsche announcement on a Sunday[26]. Similarly, the algorithm will be unable to protect from the 20% price slump after the announcement of the investigation into VOW for rigged emissions test. [27]

[25] First and third column of the table

[26] http://www.reuters.com/article/us-volkswagen-idUSTRE49R3I920081028

[27] http://www.reuters.com/article/us-usa-volkswagen-idUSKCN0RL0II20150921 Sentiment data can, however, be extremely useful in analyzing the fallout from such events; See: https://amareos.com/blog/2015/10/28/the-market-sentimentalist-german-equities-die-daumen-drucken/

Despite those misclassifications, the Random Forest had the correct classification 6233 times out of 10760, an above average 57.92%, on those out of training set four indices.

As for each day, the probability of choosing the right classification by chance should be a probability of only 33.33%. If modeled via the binomial law, the probability of choosing the right classification 6233 times out of 10760, by chance, is below 0.0001%.

3.2 Importance of Indicators

There are two ways of looking at the importance of an indicator in a random forest algorithm. One is based on the Gini Impurity indices[28], the other one on the accuracy of the output. While we won't go into detail on those[29], for the point of this paper, it is only important to know that the higher the number below, the higher the importance of the considered variables:

	MeanDecreaseAccuracy	MeanDecreaseGini
Sentiment	156.2	2316.6
Optimism	170.7	2342.5
joy	151.6	2452.4
gloom	155.6	2382.9
disgust	159.5	2434.9
anger	193.4	2397.5
fear	171.3	2381.9
anticipation	142.5	2288.7

As the above table shows, Anger turns out to be the most important individual emotion based on accuracy of output whereas Joy is the most important according to the Gini Impurity ranking. These results accord well with popular notions as to the predominant emotions during market reversals, be it a peak or trough.

That said, it should be noted that the values for each emotion are relatively close for each indicator. We have been unable, so far, through visualization technique based on dominant sentiment, to identify at which stage of a price cycle a given asset is located. This led us to believe that it is a very complex process necessitating our current approach.

[28] https://en.wikipedia.org/wiki/Decision_tree_learning

[29] If interested, please read http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#varimp

3.3 Trading strategy

3.3.1 Implementation

As seen above the algoritm has a relatively low rate of false positives for Tops and Bottoms, which is a very appealing characteristic for developing a trading strategy. However, the lower rates of detection may be an issue. How would that affect the performance of a trading strategy based on it?

We implement an extremely simple trading strategy based on the output of the random forest classifications with the following decision rules:

Bottom signal => Build position or stay long

Top signal => Remain flat or cut position

Neutral signal => Stay the course

For once we omitted transaction costs from the return calculation. Anyone planning a real life implementation of the strategy should be mindful of this. However, because trading is limited – once or twice a year based on the 20 indices over the past decade – impact would definitely be minimal.

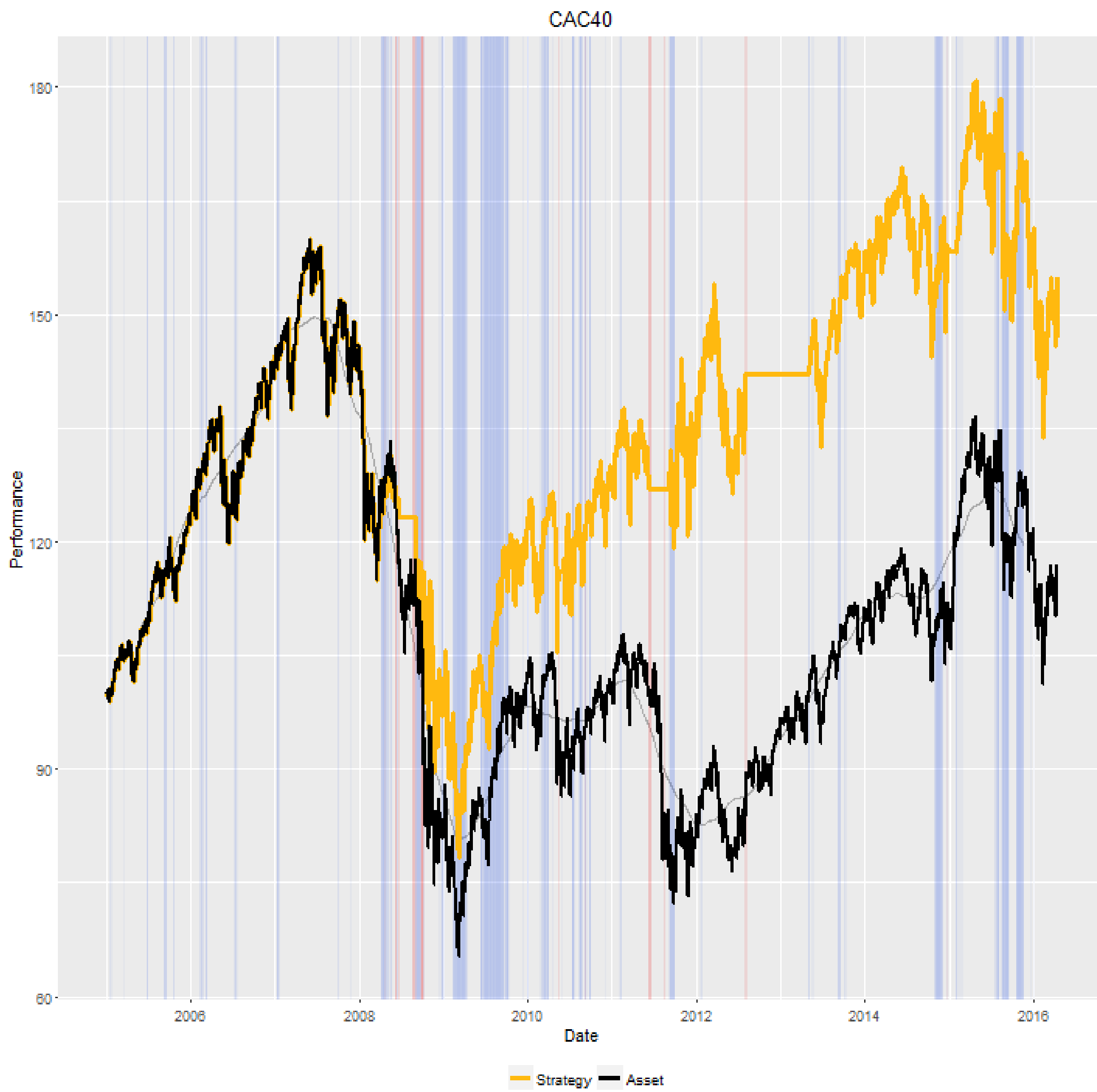
As before, we will focus on the four indices excluded from the training dataset, which will allow us to check the algorithm's out-of-sample performance.

We will also test the strategy on the last six months worth of data for the all the indices. The sentiments for those last six months were available to us but as the future returns of the indices are not known, we couldn't calculate a target value. That's why those dates were not part of the training set and provide us another useful way to cross-check out-of-sample performance.

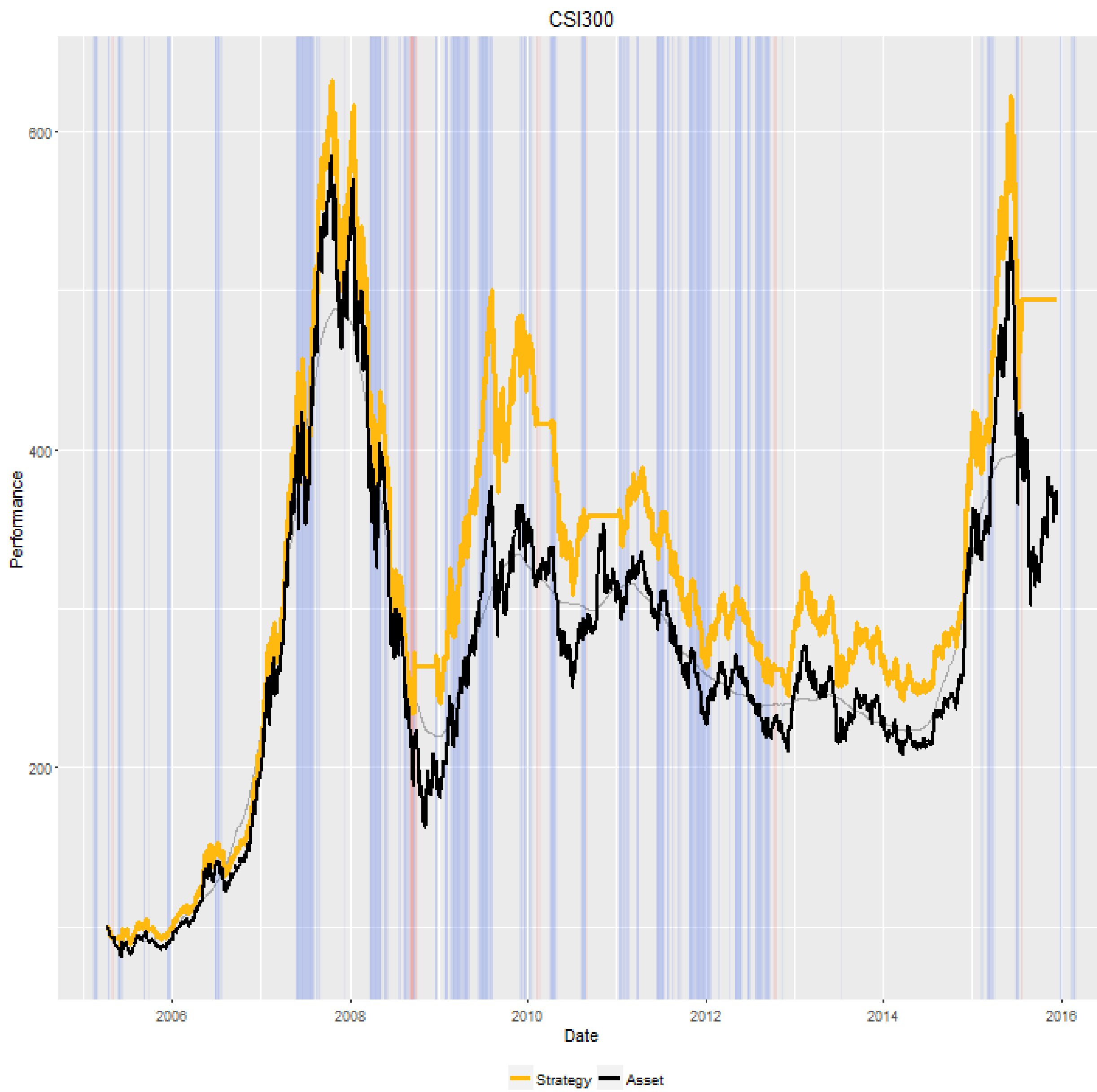
3.3.2 Performances on CAC40, CSI300, Nikkei and S&P500

The four charts below plot the performance of the strategy for the four excluded indices (CAC40, CSI300, Nikkei and S&P500). The vertical red or blue lines denote when the algorithm identified a market top or a bottom respectively. The yellow line is the cumulative return of the strategy; the black line is the return of the underlying index and the light grey, a 200 day moving average of the index. All strategies start at 100, fully invested.

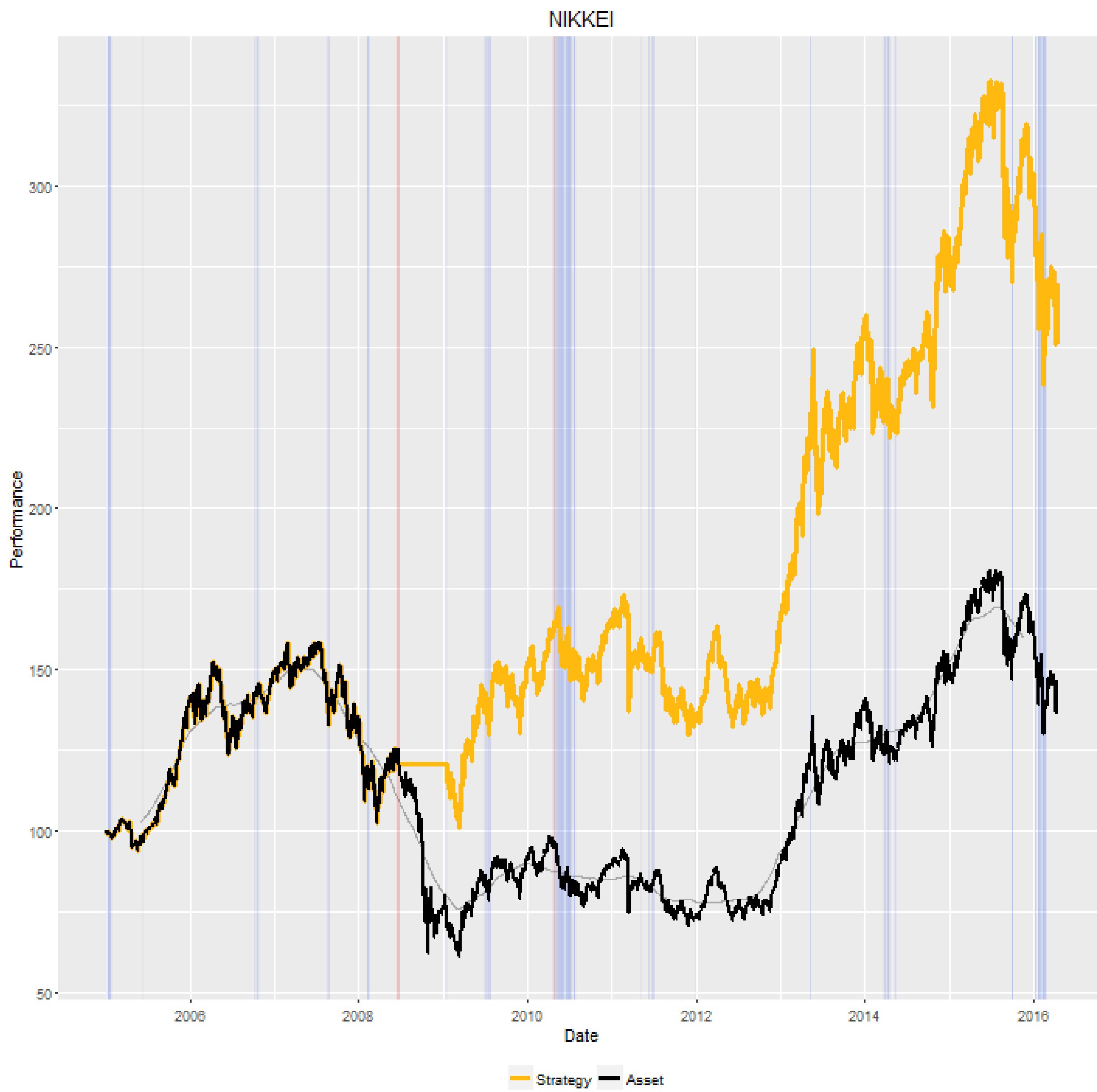
As we can see from the charts, a momentum strategy based on moving average would induce much trading than our sentiment based strategy. While an advantage for our strategy, this is due to the small number of actual reversal, but as well to the fact that our algorithm is missing a fair amount.



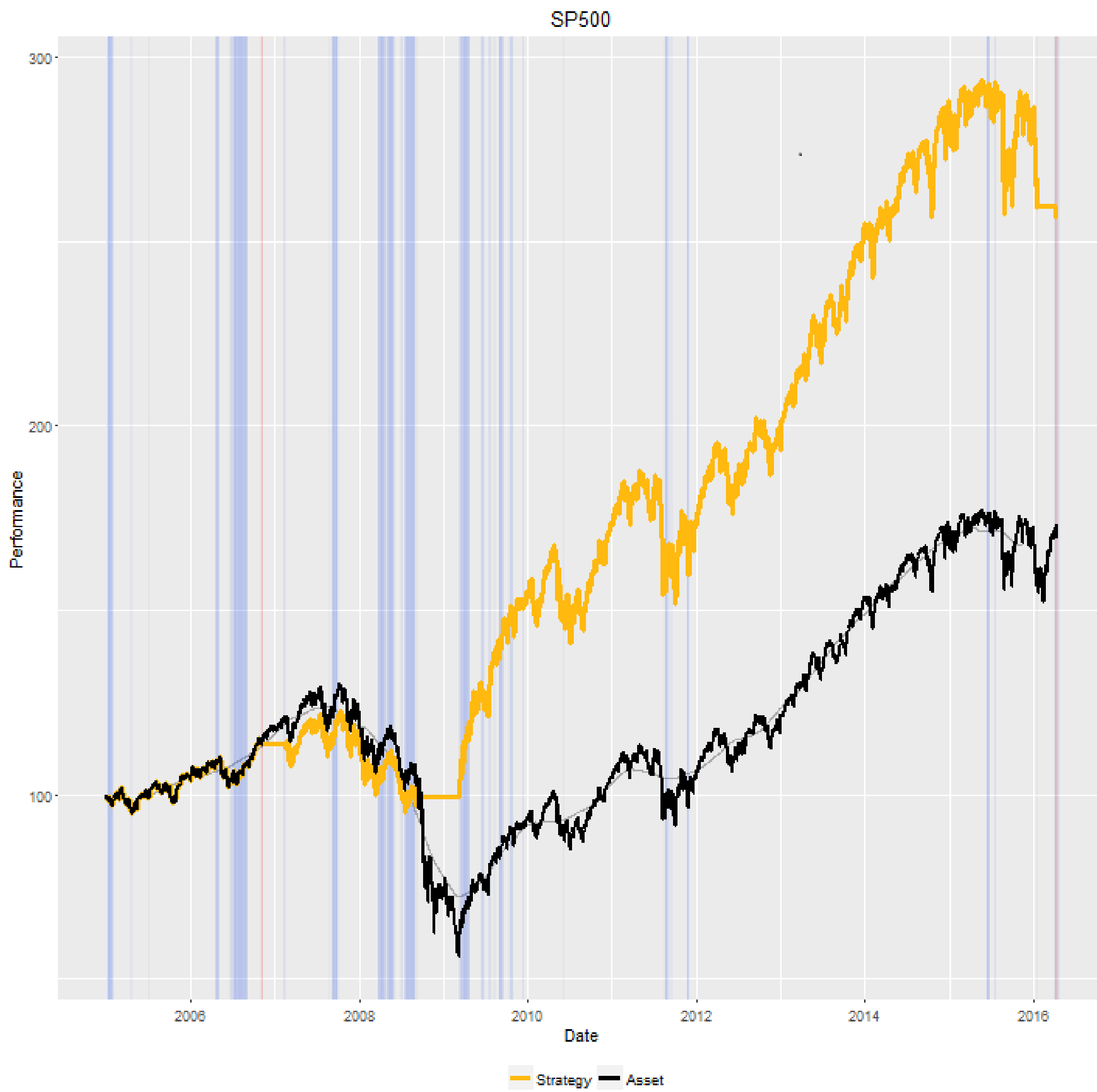
On the CAC40, the signal was messy, notably for the troubled period of 2008-2009. But the timing on the 2011 drawdown was good.



On the CSI300, there were notably more signals and the strategy was really volatile, in line with the underlying. The strategy only avoided a small part of the drawdown of 2008. Nevertheless, the strategy avoided a great deal of the 2015 slump, which accounted for its full outperformance.



The strategy is more stable on the Nikkei, with the bulk of the outperformance attributable to avoiding the majority of the 2008 drawdown.



On S&P 500, it is interesting to see the algorithm sending a top signal early at the end of 2006. The timing of the signals in 2008-2009 couldn't be better though, as most of the drawdown is avoided and the strategy got reinvested at the actual bottom of the market.

The strategy would have underperformed the index in 2016 so far, as it recently emitted a top signal.

We will see in a few months, if this signal was correct. In the meantime, it seems too early.

On all those indices, the maximum drawdowns of the strategy are significantly smaller than the ones of the underlying index. Consequently, on a risk-adjusted basis, the strategy performs better than a simple buy and hold of the index.

	Maximum Drawdowns		
	Strategy	Index	Difference
CAC40	-51.0%	-59.2%	8.1%
CSI300	-63.0%	-72.3%	9.3%
Nikkei	-36.2%	-61.4%	25.1%
S&P500	-22.4%	-56.8%	34.4%

3.3.3 Performance on the last six months

The algorithm couldn't be taught on the last six months period of data for each index, as the maxima/minima and by consequence the target values are yet unknown. Still we have sentiment indicators for the period so our algorithm can produce an output on which we can build a strategy.

As some indices moves, especially the large ones, are correlated, part of our previous results may have come from this correlation. This is not the case here, so the results are particularly interesting, and we provide all the charts for this period in annex.

For seven indices, the algorithm produced only neutral signals during the six month period. Those indices have lost on average 9.5%.

For the thirteen remaining indices, the performances of the strategy and the underlying are compiled in the following table:

	Strategy	Index	Difference
IBEX35	-15.0%	-20.8%	5.8%
EU50	-11.1%	-15.8%	4.7%
SP500	-10.1%	-0.6%	-9.5%
Nasdaq	-8.7%	-3.0%	-5.7%
SMI20	-2.9%	-14.6%	11.7%
MSCI50	1.7%	-5.9%	7.6%
Russel2000	3.3%	-7.6%	10.9%
DJ30	3.6%	1.5%	2.1%
DAX30	6.6%	-10.3%	16.9%
CSI300	7.0%	-5.1%	12.1%
HK50	10.5%	-15.7%	26.2%
IBOV	13.2%	3.4%	9.8%
RU50	22.8%	7.3%	15.5%
Average	**1.6%**	**-6.7%**	**8.3%**
Including Indices with only neutral signal			
Total Average	**-1.5%**	**-7.6%**	**6.2%**

On average the strategy overperformed the indices by more than 8%. If we include the indices for which the algorithm only produce neutral signal (as we would likely be invested in those), the overperformance is still 6.2% on average.

Only for the S&P500 and the Nasdaq have the signal been counterproductive, as the strategy is currently non invested while the indices have rebounded from their low. It will be interesting to see the performance of those two in a few months.

For all the other ones, the signals have proved valuable.

While this period is obviously too short to draw definitive conclusions, the results are nonetheless promising.

4 Conclusion

For this study, we rigorously deployed a proper machine learning process with the Random Forest fitted to the training test reducing our risks of data mining.

The Random Forest based on Amareos sentiment indicators has been able to produce valuable signals when applied to a testing test.

However complete, this study only scratched the surface of what could be done. The classifications of the assets returns can be refined; more classes could be imagined, depending on different time frames.

Also a one size fit all approach may not be the right for indices with quite distinctive characteristics; it would be interesting to apply this method on a large group of stocks from the same index or market.

Combining the sentiment indicators with other common factors is another avenue to be pursued. For example, we could study the complementary of the sentiment indicators with the Fama/French framework extended by Carhart.

While the strategy described in this paper and based on Amareos indicators is only theoretical, it is promising enough to think about a real life implementation.

Acknowledgements

We would like to thank Alexandre Avanian, El Mehdi Benhmade, Bertrand Chen, Yves Guo, the whole team at Mandarin Capital, in particular Nitin Dialdas and Samuel Faveur, and the whole team at Amareos, Federico Bagiotti, Philippe El-Asmar, Jerome Favresse, Jacques Labbaci and Ryan Shea.

Annex

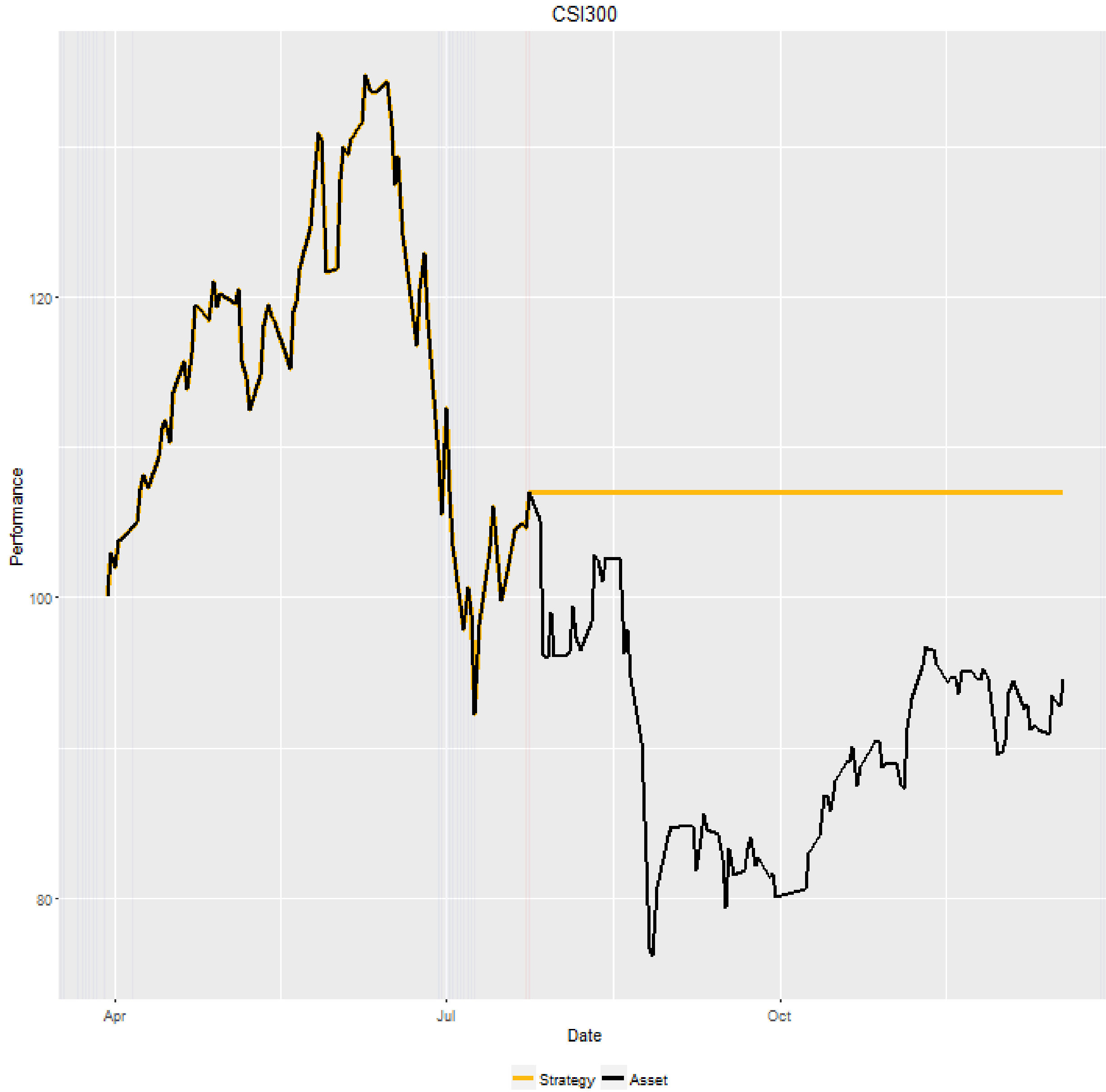
Correlations between indices reversal

	ASX	CAC40	CSI300	DAX30	DJ30	EU50	FTSE100	HK50	IBEX35	IBOV	MSCI50	Nasdaq	NIFTY50	NIKKEI	RU50	RUSSEL2000	SG30	SMI20	SP500	TSX250	Maximum
ASX		0.73	0.41	0.64	0.69	0.68	0.74	0.61	0.56	0.45	0.51	0.77	0.50	0.49	0.36	0.67	0.67	0.69	0.71	0.74	0.77
CAC40	0.73		0.42	0.80	0.61	0.92	0.63	0.61	0.59	0.46	0.54	0.62	0.42	0.52	0.37	0.70	0.58	0.81	0.63	0.64	0.92
CSI300	0.41	0.42		0.45	0.23	0.42	0.31	0.58	0.44	0.44	0.51	0.36	0.46	0.27	0.36	0.34	0.55	0.35	0.27	0.34	0.58
DAX30	0.64	0.80	0.45		0.54	0.81	0.56	0.60	0.56	0.46	0.57	0.59	0.41	0.47	0.38	0.63	0.58	0.68	0.58	0.58	0.81
DJ30	0.69	0.61	0.23	0.54		0.62	0.82	0.46	0.48	0.36	0.45	0.74	0.37	0.41	0.31	0.66	0.53	0.60	0.91	0.77	0.91
EU50	0.68	0.92	0.42	0.81	0.62		0.60	0.63	0.64	0.46	0.55	0.60	0.40	0.51	0.38	0.69	0.59	0.79	0.62	0.62	0.92
FTSE100	0.74	0.63	0.31	0.56	0.82	0.60		0.50	0.47	0.44	0.50	0.75	0.45	0.42	0.36	0.63	0.58	0.59	0.82	0.81	0.82
HK50	0.61	0.61	0.58	0.60	0.46	0.63	0.50		0.58	0.54	0.75	0.55	0.59	0.35	0.50	0.57	0.66	0.55	0.50	0.53	0.75
IBEX35	0.56	0.59	0.44	0.56	0.48	0.64	0.47	0.58		0.48	0.52	0.50	0.41	0.40	0.42	0.50	0.55	0.56	0.49	0.49	0.64
IBOV	0.45	0.46	0.44	0.46	0.36	0.46	0.44	0.54	0.48		0.71	0.41	0.58	0.33	0.64	0.42	0.52	0.44	0.40	0.47	0.71
MSCI50	0.51	0.54	0.51	0.57	0.45	0.55	0.50	0.75	0.52	0.71		0.50	0.65	0.43	0.62	0.54	0.59	0.50	0.48	0.52	0.75
Nasdaq	0.77	0.62	0.36	0.59	0.74	0.60	0.75	0.55	0.50	0.41	0.50		0.43	0.44	0.39	0.72	0.60	0.59	0.81	0.73	0.81
NIFTY50	0.50	0.42	0.46	0.41	0.37	0.40	0.45	0.59	0.41	0.58	0.65	0.43		0.38	0.50	0.33	0.49	0.42	0.39	0.50	0.65
NIKKEI	0.49	0.52	0.27	0.47	0.41	0.51	0.42	0.35	0.40	0.33	0.43	0.44	0.38		0.31	0.43	0.40	0.52	0.41	0.44	0.52
RU50	0.36	0.37	0.36	0.38	0.31	0.38	0.36	0.50	0.42	0.64	0.62	0.39	0.50	0.31		0.44	0.43	0.35	0.35	0.38	0.64
RUSSEL2000	0.67	0.70	0.34	0.63	0.66	0.69	0.63	0.57	0.50	0.42	0.54	0.72	0.33	0.43	0.44		0.54	0.66	0.71	0.63	0.72
SG30	0.67	0.58	0.55	0.58	0.53	0.59	0.58	0.66	0.55	0.52	0.59	0.60	0.49	0.40	0.43	0.54		0.53	0.55	0.61	0.67
SMI20	0.69	0.81	0.35	0.68	0.60	0.79	0.59	0.55	0.56	0.44	0.50	0.59	0.42	0.52	0.35	0.66	0.53		0.61	0.60	0.81
SP500	0.71	0.63	0.27	0.58	0.91	0.62	0.82	0.50	0.49	0.40	0.48	0.81	0.39	0.41	0.35	0.71	0.55	0.61		0.78	0.91
TSX250	0.74	0.64	0.34	0.58	0.77	0.62	0.81	0.53	0.49	0.47	0.52	0.73	0.50	0.44	0.38	0.63	0.61	0.60	0.78		0.81

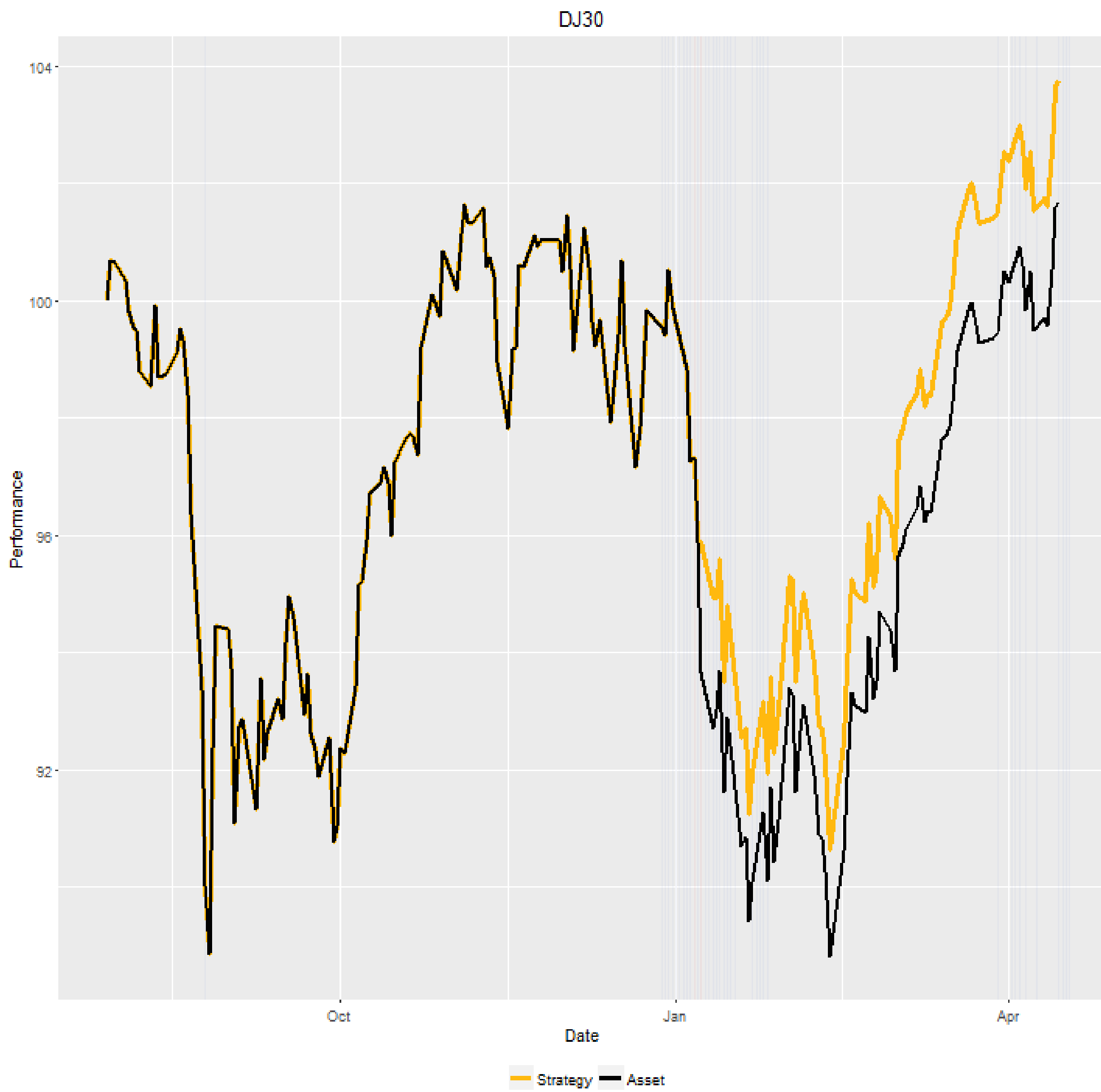
Correlations between Target Values

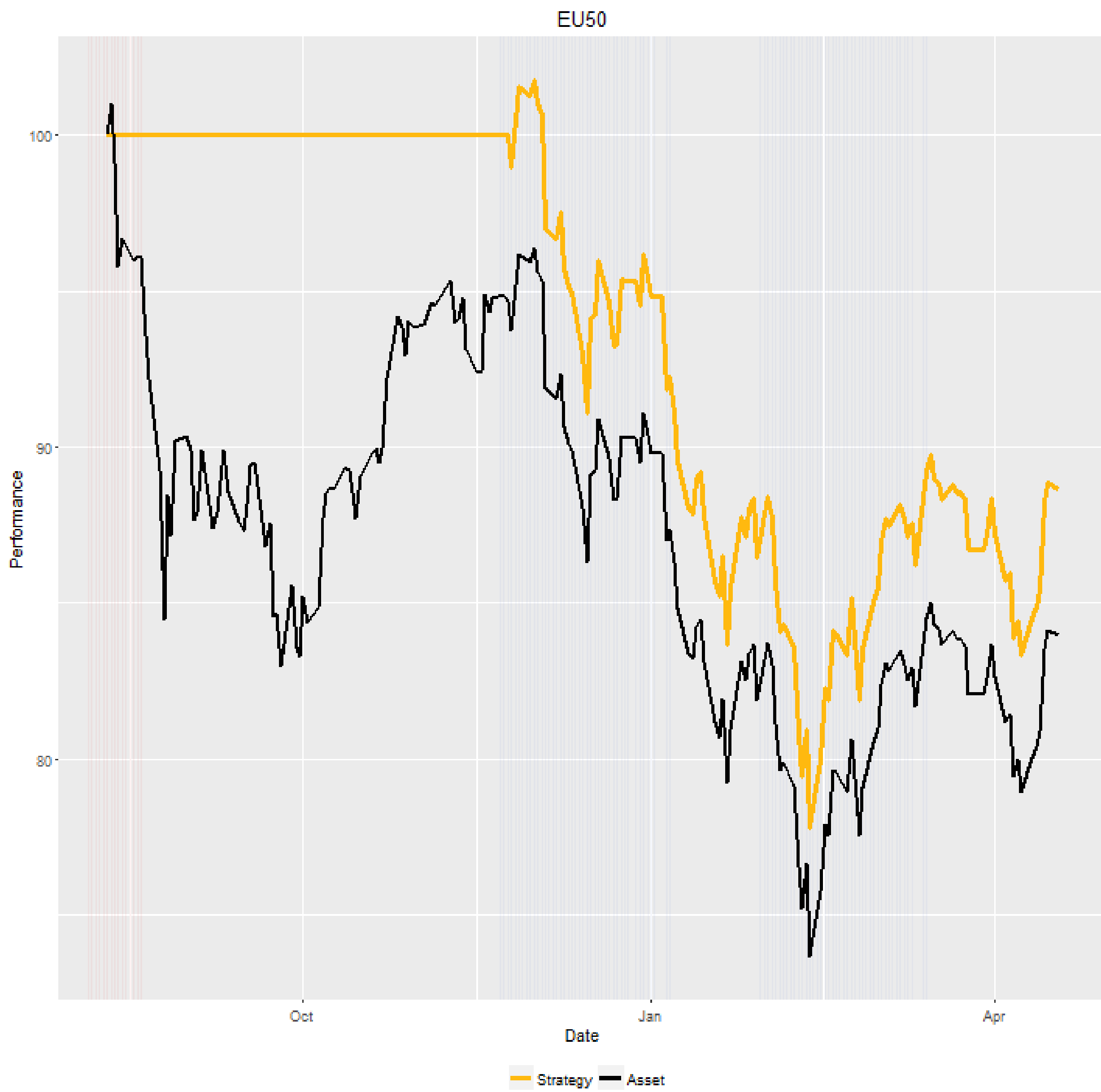
	ASX	CAC40	CSI300	DAX30	DJ30	EU50	FTSE100	HK50	IBEX35	IBOV	MSCI50	Nasdaq	NIFTY50	NIKKEI	RU50	RUSSEL2000	SG30	SMI20	SP500	TSX250	Maximum
ASX		0.89	0.61	0.88	0.86	0.89	0.91	0.84	0.82	0.78	0.85	0.84	0.73	0.79	0.77	0.85	0.91	0.83	0.87	0.87	0.91
CAC40	0.89		0.50	0.96	0.85	0.98	0.92	0.75	0.85	0.68	0.76	0.83	0.63	0.78	0.68	0.88	0.81	0.89	0.87	0.82	0.98
CSI300	0.61	0.50		0.54	0.47	0.51	0.45	0.68	0.50	0.58	0.60	0.45	0.59	0.44	0.43	0.36	0.65	0.35	0.44	0.47	0.68
DAX30	0.88	0.96	0.54		0.86	0.96	0.89	0.78	0.79	0.70	0.79	0.85	0.65	0.77	0.68	0.88	0.83	0.84	0.87	0.83	0.96
DJ30	0.86	0.85	0.47	0.86		0.86	0.89	0.75	0.72	0.66	0.77	0.91	0.59	0.73	0.67	0.92	0.82	0.79	0.98	0.82	0.98
EU50	0.89	0.98	0.51	0.96	0.86		0.90	0.78	0.89	0.70	0.78	0.83	0.66	0.76	0.68	0.87	0.80	0.88	0.88	0.83	0.98
FTSE100	0.91	0.92	0.45	0.89	0.89	0.90		0.76	0.79	0.74	0.82	0.86	0.65	0.78	0.75	0.89	0.86	0.87	0.91	0.86	0.92
HK50	0.84	0.75	0.68	0.78	0.75	0.78	0.76		0.75	0.83	0.92	0.78	0.83	0.63	0.78	0.71	0.87	0.62	0.76	0.81	0.92
IBEX35	0.82	0.85	0.50	0.79	0.72	0.89	0.79	0.75		0.69	0.73	0.70	0.65	0.64	0.64	0.72	0.73	0.77	0.73	0.76	0.89
IBOV	0.78	0.68	0.58	0.70	0.66	0.70	0.74	0.83	0.69		0.93	0.70	0.78	0.59	0.85	0.63	0.82	0.61	0.68	0.84	0.93
MSCI50	0.85	0.76	0.60	0.79	0.77	0.78	0.82	0.92	0.73	0.93		0.81	0.84	0.66	0.90	0.75	0.90	0.65	0.80	0.91	0.93
Nasdaq	0.84	0.83	0.45	0.85	0.91	0.83	0.86	0.78	0.70	0.70	0.81		0.67	0.77	0.70	0.93	0.84	0.76	0.95	0.87	0.95
NIFTY50	0.73	0.63	0.59	0.65	0.59	0.66	0.65	0.83	0.65	0.78	0.84	0.67		0.57	0.71	0.57	0.79	0.56	0.62	0.72	0.84
NIKKEI	0.79	0.78	0.44	0.77	0.73	0.76	0.78	0.63	0.64	0.59	0.66	0.77	0.57		0.59	0.77	0.73	0.79	0.77	0.71	0.79
RU50	0.77	0.68	0.43	0.68	0.67	0.68	0.75	0.78	0.64	0.85	0.90	0.70	0.71	0.59		0.69	0.78	0.58	0.70	0.85	0.90
RUSSEL2000	0.85	0.88	0.36	0.88	0.92	0.87	0.89	0.71	0.72	0.63	0.75	0.93	0.57	0.77	0.69		0.80	0.82	0.95	0.86	0.95
SG30	0.91	0.81	0.65	0.83	0.82	0.80	0.86	0.87	0.73	0.82	0.90	0.84	0.79	0.73	0.78	0.80		0.73	0.84	0.88	0.91
SMI20	0.83	0.89	0.35	0.84	0.79	0.88	0.87	0.62	0.77	0.61	0.65	0.76	0.56	0.79	0.58	0.82	0.73		0.82	0.72	0.89
SP500	0.87	0.87	0.44	0.87	0.98	0.88	0.91	0.76	0.73	0.68	0.80	0.95	0.62	0.77	0.70	0.95	0.84	0.82		0.87	0.98
TSX250	0.87	0.82	0.47	0.83	0.82	0.83	0.86	0.81	0.76	0.84	0.91	0.87	0.72	0.71	0.85	0.86	0.88	0.72	0.87		0.91

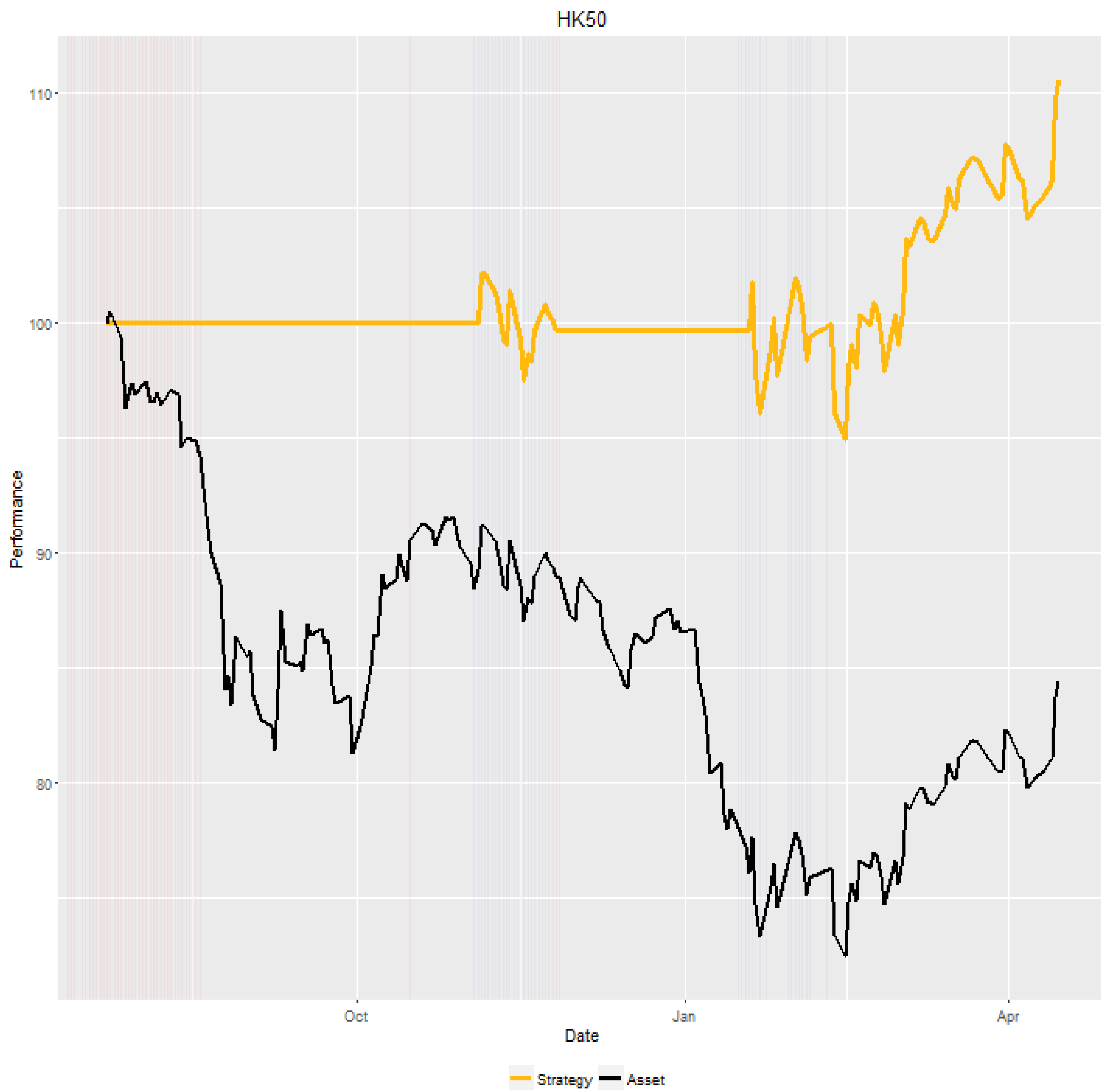
Performances on last six month periods

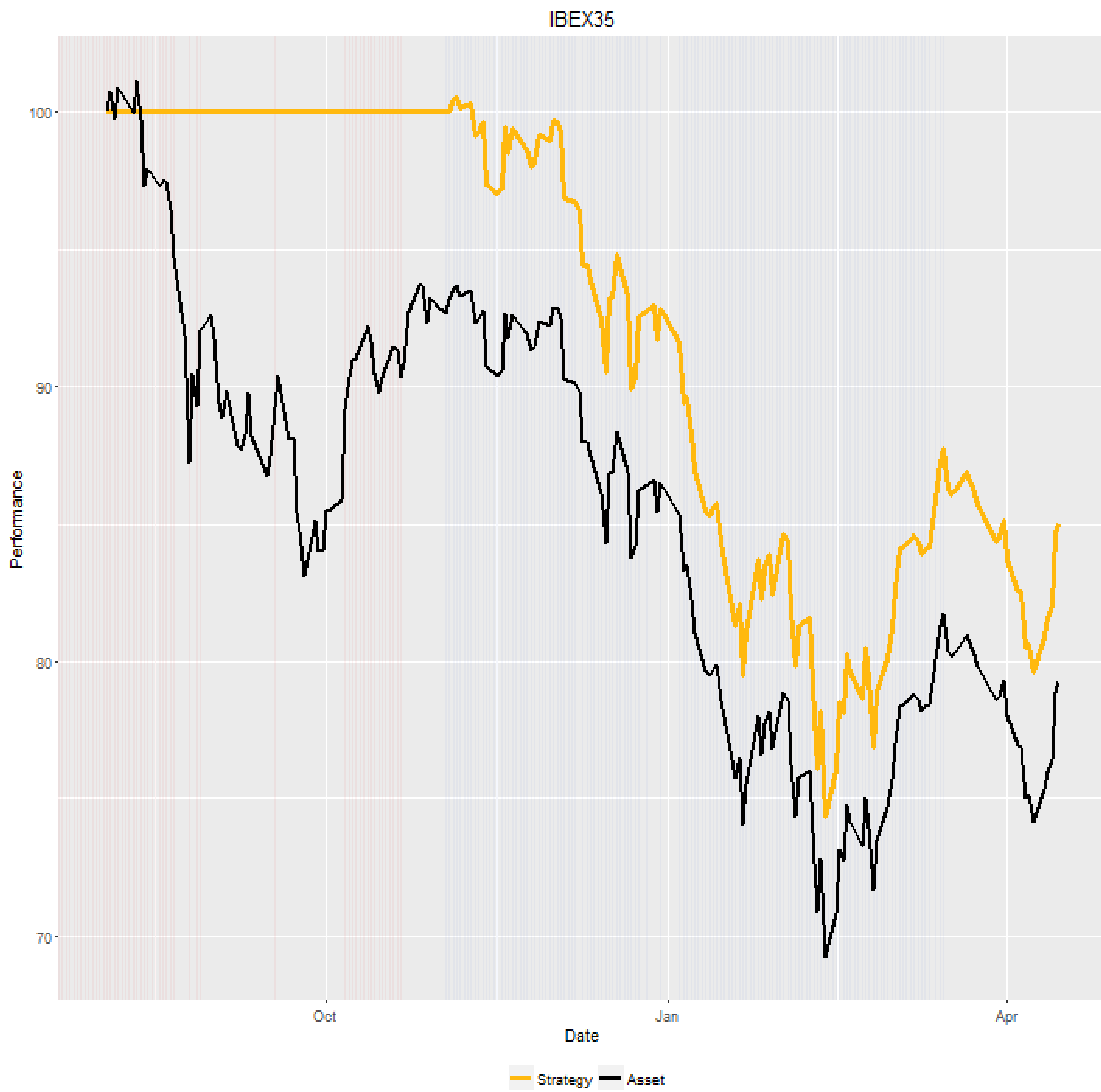


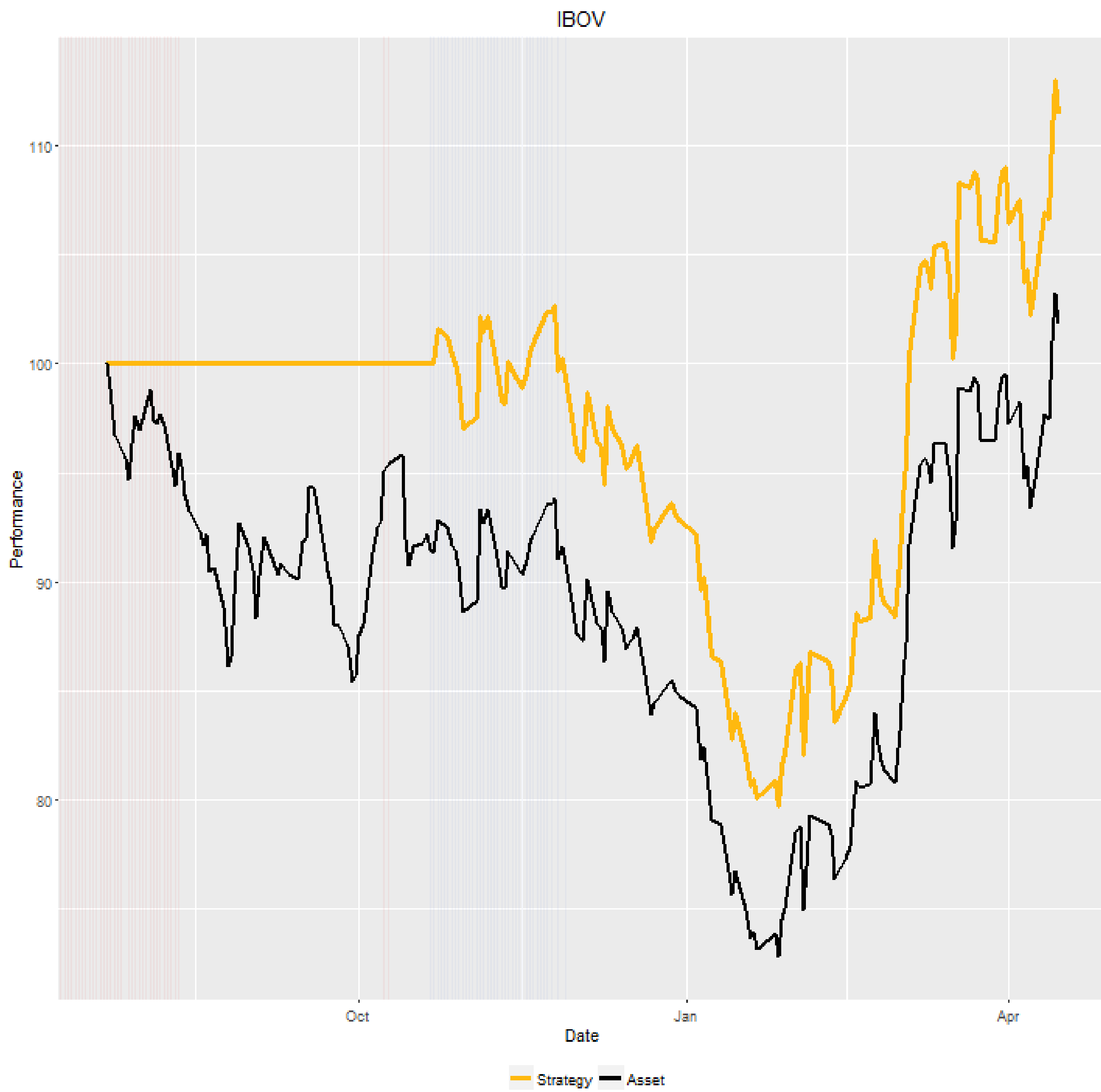

DAX30
Performance
120
110
100
90
80
Oct
Jan
Apr
Date
Strategy
Asset


DJ30
104
100
96
92
Performance
Oct
Jan
Apr
Date
Strategy
Asset

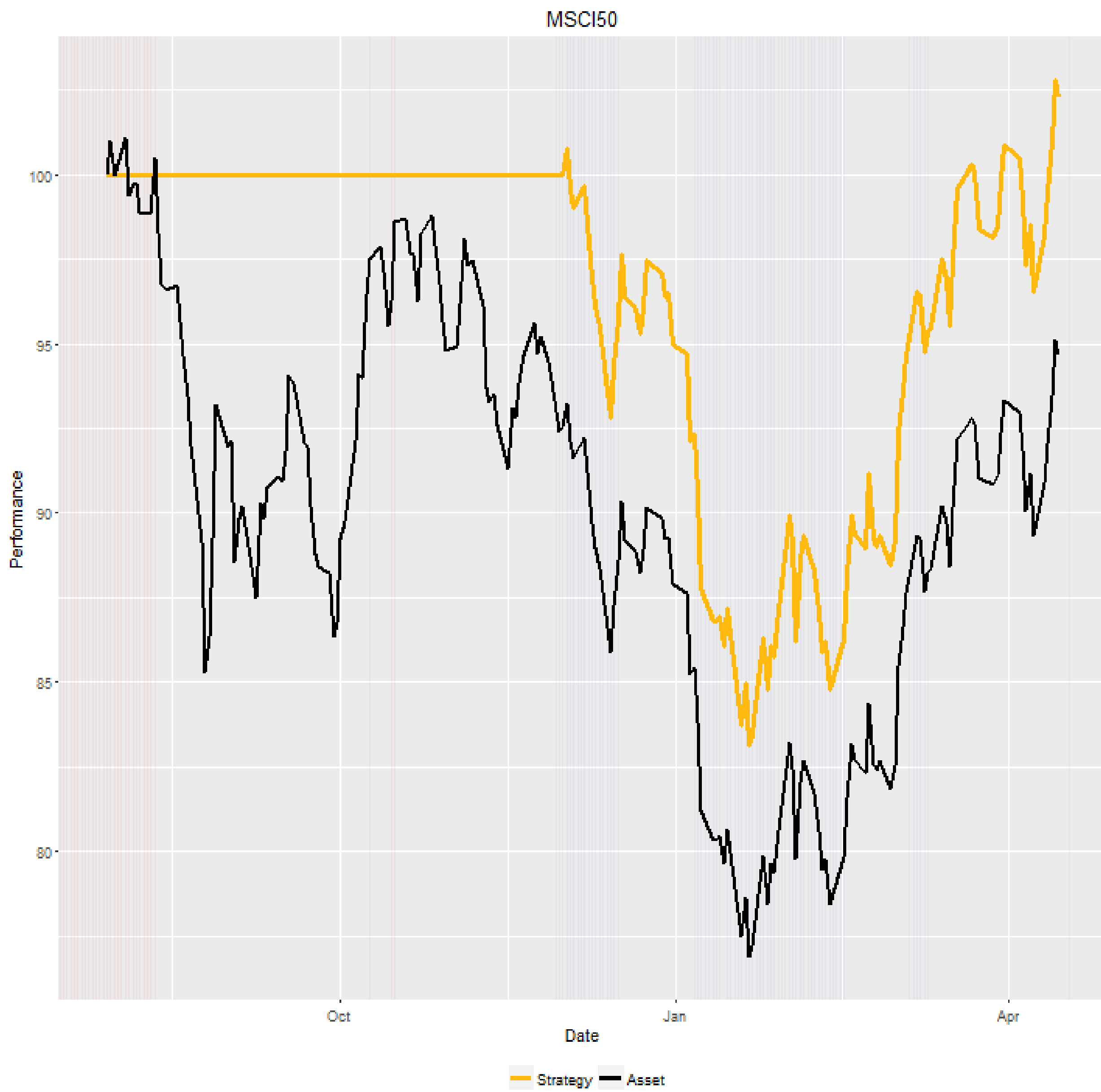

EU50
100
90
80
Performance
Oct
Jan
Apr
Date
Strategy
Asset


HK50
110
100
90
80
Performance
Oct
Jan
Apr
Date
Strategy
Asset

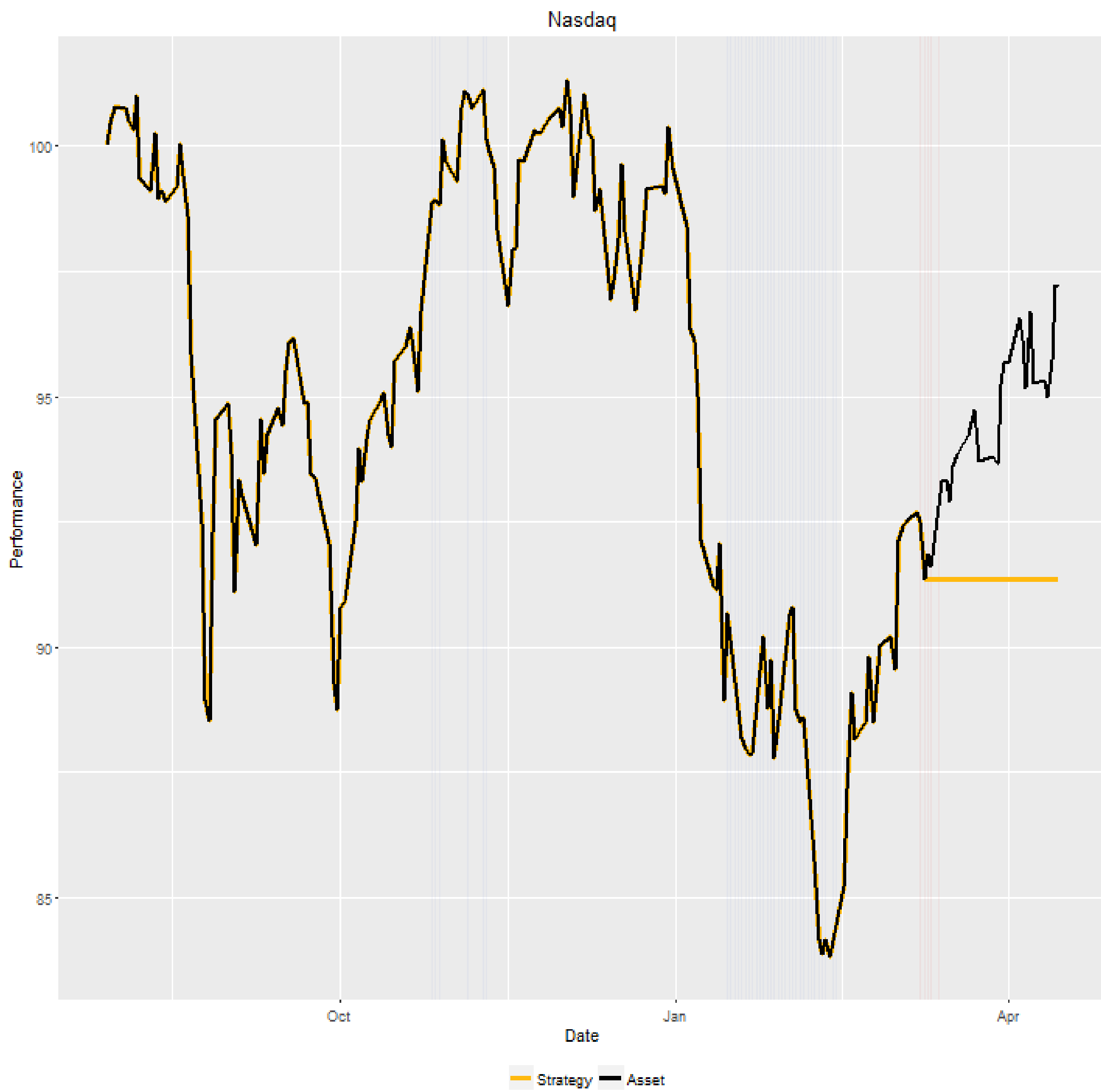

IBEX35
100
90
80
70
Performance
Oct
Jan
Apr
Date
Strategy
Asset



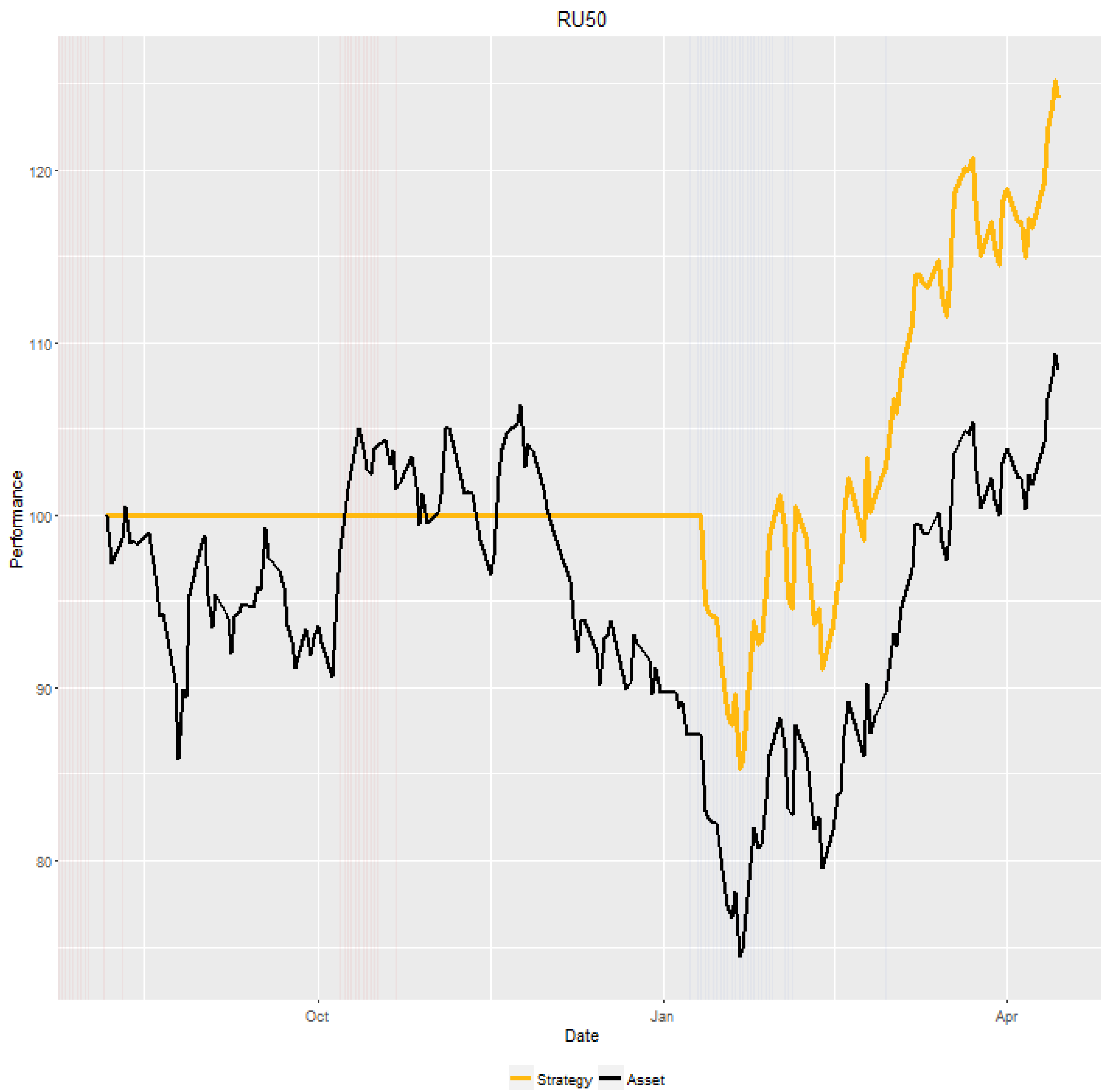
IBOV
110
100
90
80
Performance
Oct
Jan
Apr
Date
Strategy
Asset

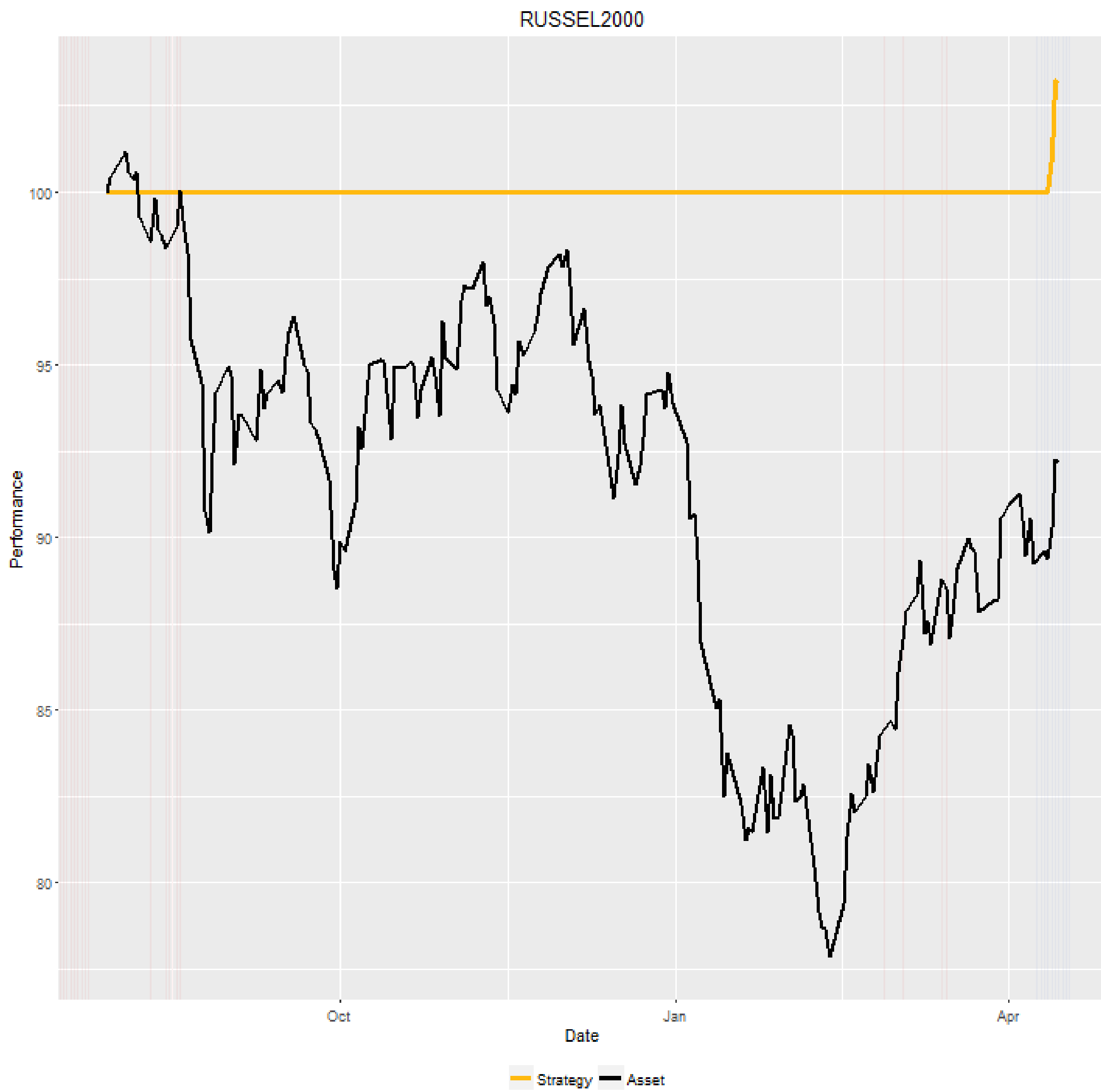


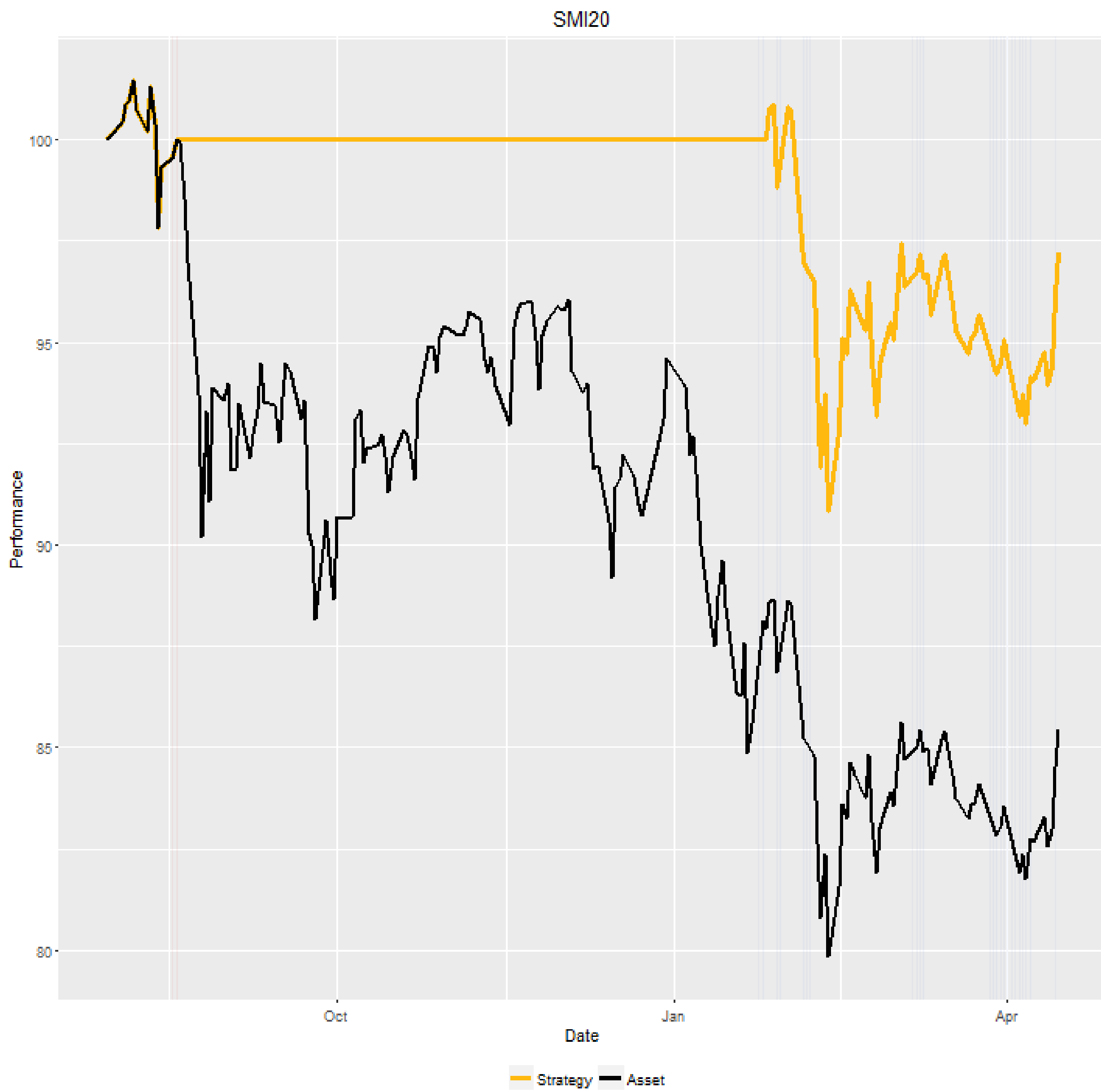
MSCI50
Performance
100
95
90
85
80
Oct
Jan
Apr
Date
Strategy
Asset



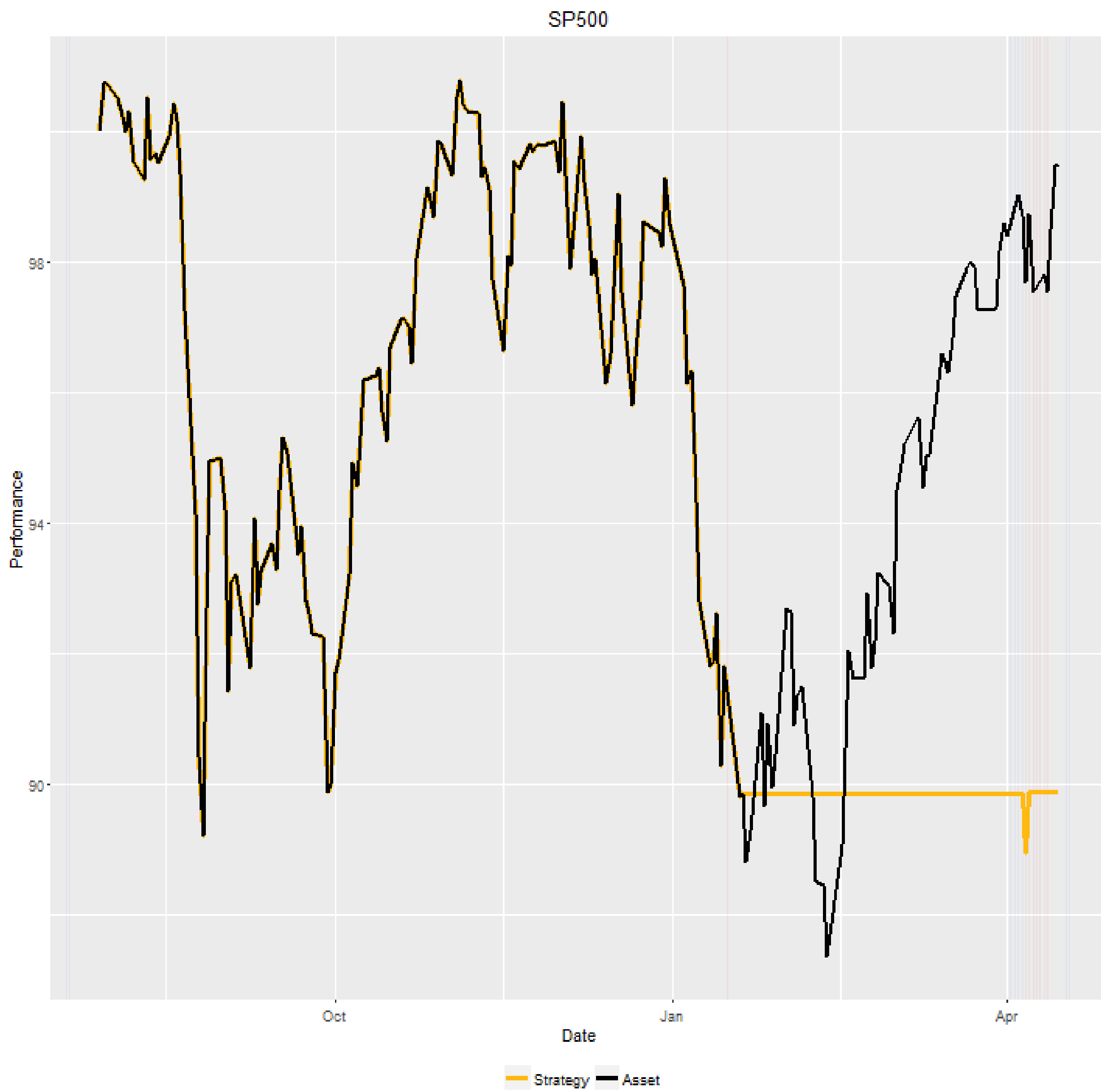
Nasdaq
100
95
90
85
Performance
Oct
Jan
Apr
Date
Strategy
Asset


RU50
120
110
100
90
80
Performance
Oct
Jan
Apr
Date
Strategy
Asset


RUSSEL2000
Performance
100
95
90
85
80
Oct
Jan
Apr
Date
Strategy
Asset



SMI20
100
95
90
85
80
Performance
Oct
Jan
Apr
Date
Strategy
Asset


SP500
Performance
98
94
90
Oct
Jan
Apr
Date
Strategy
Asset